

# **A Mathematical Theory of the Unknown**

**Journey Beyond the Frontiers of Human Understanding**

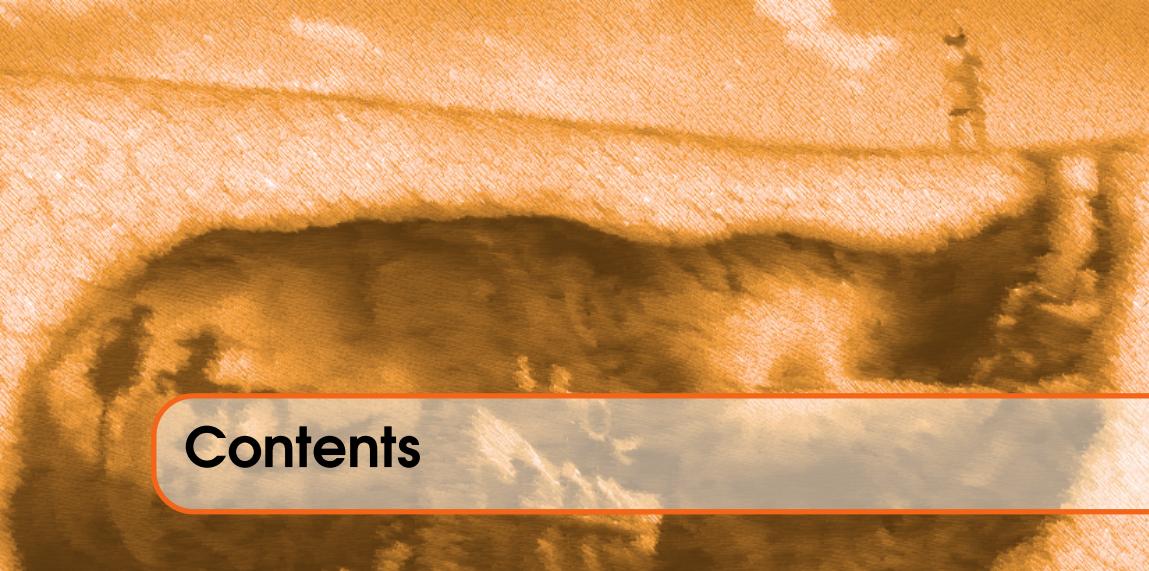
**R. A. García Leiva**

**(This book is 90% complete)**

Copyright © 2025 R. A. García Leiva  
PUBLISHED BY THE AUTHOR



To my wife Justi, my son Daniel,  
and my two daughters Teresa and Lucía.



# Contents

<b>1</b>	<b>Introduction</b>	<b>25</b>
1.1	Scientific Knowledge	26
1.2	Entities	27
1.3	Representations	28
1.4	Descriptions	31
1.5	Miscoding	33
1.6	Inaccuracy	37
1.7	Surfeit	39
1.8	Nescience	41
1.9	Perfect Knowledge	43
1.10	Unknown Unknowns	44

<b>2</b>	<b>Fundamental Elements</b>	<b>51</b>
2.1	Entities	52
2.2	Representation Oracle	54
2.3	Representations	63
2.4	Joint Representations	71
2.5	Descriptions	76
2.6	Descriptions for Joint Representations	81
2.7	Conditional Descriptions	83
2.8	Research Areas	86
2.9	References	90
<b>3</b>	<b>Miscoding</b>	<b>91</b>
3.1	Miscoding	92
3.2	Joint Miscoding	95
3.3	Decreasing Miscoding	97
3.4	Targetless Representations	99
3.5	Miscoding of Areas	101
<b>4</b>	<b>Inaccuracy</b>	<b>105</b>
4.1	Inaccuracy	106
4.2	Conditional Inaccuracy	109
4.3	Decreasing Inaccuracy	111
4.4	Inaccuracy-Miscoding Rate of Change	113
4.5	Inaccuracy of Areas	116
<b>5</b>	<b>Surfeit</b>	<b>119</b>
5.1	Surfeit	120
5.2	Redundancy	123
5.3	Conditional Surfeit	124
5.4	Decreasing Surfeit	126
5.5	Surfeit-inaccuracy rate of Change	127
5.6	Surfeit of Areas	130

<b>6</b>	<b>Nescience</b>	<b>131</b>
6.1	<b>Nescience</b>	132
6.2	<b>Minimizing Nescience</b>	138
6.2.1	Global Criterion .....	139
6.2.2	Weighting Method .....	141
6.3	<b>Perfect Knowledge</b>	143
6.4	<b>Current Best Hypothesis</b>	146
6.5	<b>Unknown Unknown</b>	147
6.6	<b>Science vs. Pseudoscience</b>	151
6.7	<b>Nescience of Areas</b>	152
<b>7</b>	<b>Interesting Questions</b>	<b>155</b>
7.1	<b>Integrative Depth</b>	156
7.2	<b>Applicability</b>	161
7.3	<b>Maturity</b>	163
7.4	<b>Interestingness</b>	165
7.5	<b>Interesting Questions</b>	167
7.6	<b>New Topics</b>	170
7.6.1	Candidate Topic Combinations .....	170
7.6.2	Multi-Objective Characterization of New Topics .....	171
7.6.3	Decision Makers and Practical Selection .....	172
7.6.4	Intradisciplinary and Interdisciplinary Combinations .....	172

## II

## Part 2: Applications

<b>8</b>	<b>Machine Learning</b>	<b>177</b>
8.1	<b>Nescience Python Library</b>	178
8.2	<b>A Note About Compression</b>	178
8.3	<b>Miscoding</b>	181
8.4	<b>Inaccuracy</b>	188
8.5	<b>Surfeit</b>	193
8.6	<b>Nescience</b>	196
8.7	<b>Auto Classification</b>	201
8.8	<b>Auto Regression</b>	202
8.9	<b>Time Series</b>	202

<b>8.10</b>	<b>Anomaly Detection</b>	<b>205</b>
<b>8.11</b>	<b>Decision Trees</b>	<b>208</b>
8.11.1	Algorithm Description . . . . .	209
8.11.2	Algorithm Evaluation . . . . .	212
<b>8.12</b>	<b>Algebraic Model Selection</b>	<b>218</b>
<b>9</b>	<b>Analysis of Science</b> . . . . .	<b>223</b>
<b>9.1</b>	<b>Describing Current Knowledge</b>	<b>224</b>
<b>9.2</b>	<b>Measuring Knowledge</b>	<b>227</b>
9.2.1	Surfeit . . . . .	229
9.2.2	Inaccuracy . . . . .	233
9.2.3	Nescience . . . . .	236
9.2.4	Conclusion . . . . .	239
<b>9.3</b>	<b>Measuring Research Areas</b>	<b>240</b>
<b>9.4</b>	<b>The Evolution of Knowledge</b>	<b>245</b>
<b>9.5</b>	<b>The Demarcation Problem</b>	<b>248</b>
<b>10</b>	<b>The Discovery of the Unknown</b> . . . . .	<b>257</b>
<b>10.1</b>	<b>Relevance</b>	<b>258</b>
<b>10.2</b>	<b>Applicability</b>	<b>261</b>
<b>10.3</b>	<b>Maturity</b>	<b>262</b>
<b>10.4</b>	<b>Interestingness</b>	<b>263</b>
<b>10.5</b>	<b>Interesting Research Questions</b>	<b>266</b>
<b>10.6</b>	<b>New Research Topics</b>	<b>269</b>
<b>10.7</b>	<b>The Unknown Unknowns Project</b>	<b>271</b>

### III

### Part 3: Prerequisites

<b>A</b>	<b>Discrete Mathematics</b> . . . . .	<b>283</b>
A.1	Sets, Relations and Functions	284
A.2	Strings and Languages	288
A.3	Counting Methods	291
A.4	Matrices	293
A.5	Graphs	295

<b>B</b>	<b>Discrete Probability</b>	<b>301</b>
B.1	Interpretations of Probability	302
B.2	Foundations of Probability Theory	304
B.3	Conditional Probability	309
B.4	Random Variables	314
B.4.1	Multivariate Distributions	319
B.4.2	Marginal Probability Mass Function	322
B.4.3	Conditional Probability Mass Function	325
<b>B.5</b>	<b>Characterizing Distributions</b>	<b>327</b>
B.5.1	Measures of Central Tendency	328
B.5.2	Measures of Dispersion	330
B.5.3	Measures of Statistical Relationship	333
<b>B.6</b>	<b>Common Distributions</b>	<b>336</b>
B.6.1	Uniform Distribution	337
B.6.2	Bernoulli Distribution	337
B.6.3	Binomial Distribution	338
B.6.4	Discrete Normal Distribution	339
<b>B.7</b>	<b>Large Random Samples</b>	<b>342</b>
B.7.1	Random Sample	342
B.7.2	Law of Large Numbers	345
B.7.3	Central Limit Theorem	350
<b>C</b>	<b>Computability</b>	<b>353</b>
C.1	Turing Machines	355
C.2	Universal Turing Machines	360
C.3	Non-Computable Problems	361
C.4	Computable Functions and Sets	363
C.5	Oracle Turing Machine	365
<b>D</b>	<b>Coding</b>	<b>371</b>
D.1	Coding	372
D.2	Kraft Inequality	377
D.3	Optimal Codes	380
D.4	Entropy	383
D.5	Huffman Algorithm	389
D.6	Discretization of Continuous Variables	390

<b>E</b>	<b>Complexity</b>	<b>395</b>
E.1	Strings Complexity	396
E.2	Properties of Complexity	400
E.3	Joint Kolmogorov Complexity	402
E.4	Conditional Kolmogorov complexity	403
E.5	Information Distance	406
E.6	Incompressibility and Randomness	412
<b>F</b>	<b>Learning</b>	<b>415</b>
F.1	Statistical Inference	416
F.1.1	Maximum Likelihood Estimator	419
F.1.2	Bayesian Inference	423
F.2	Machine Learning	426
F.2.1	Parametric vs. Non-parametric Models	429
F.2.2	Generative vs. Discriminative Models	430
F.3	Model Accuracy	431
F.3.1	Bias-Variance Decomposition	432
F.3.2	Regression Metrics	432
F.3.3	Classification Metrics	433
F.4	Minimum Message Length	434
F.5	Minimum Message Length	436
F.5.1	Comparison between MDL and MML	438
F.6	Multiobjective Optimization	439
F.6.1	Range of the Solutions	442
F.6.2	Trade-offs	444
F.6.3	Optimization Methods	445
<b>G</b>	<b>Philosophy of Science</b>	<b>451</b>
G.1	What is Science	452
G.2	What is an Entity	455
G.3	Observation in Science	458
G.4	Scientific Representation	460
G.5	Scientific Discovery	464
G.6	Scientific Explaination	466
G.7	Scientific Justification	470
G.8	The Limits of Science	473

<b>H</b>	<b>About Quotes and Photos</b>	<b>479</b>
H.1	Chapter Quotes	480
H.2	Chapter Pictures	483
<b>I</b>	<b>Notation</b>	<b>493</b>
	<b>Bibliography</b>	<b>497</b>
	Books	497
	Articles	499
	<b>Index</b>	<b>503</b>





## Preface

*Perfection is achieved not when there is nothing more to add,  
but when there is nothing left to take away.*  
Antoine de Saint-Exupéry

We live in an era that places a high value on knowledge and celebrates scientific advancement. The development of vaccines in record time, the detection of gravitational waves, and the use of artificial intelligence to model protein structures are just a few examples of the extraordinary progress science has made in recent years. However, this celebration of progress should not obscure the structural limitations within the scientific enterprise itself. Despite its accomplishments, science is constrained by the methodological and conceptual frameworks it employs. For example, the "publish or perish" culture in academia often results in a flood of low-quality publications, which can dilute the impact of meaningful scientific work. Another major constraint lies in the allocation of research funding, which often fails to prioritize investigations with the highest potential societal return.

This book is driven by a central aspiration: to overcome the methodological constraints of science in order to challenge established knowledge, explore the unknown, and progress towards the limits of scientific understanding. Our aim is to investigate not only well-known unsolved problems but

also those that remain outside our current awareness, the known unknowns and the unknown unknowns. But this pursuit is not merely an academic curiosity; it is a practical necessity. Rather than developing a purely theoretical framework, our goal is to create an approach that can be applied in real-world scientific contexts.

Our assumption is that to explore and ultimately reduce what we do not yet know, we need a structured framework for understanding ignorance itself. To this end, we propose a rigorous mathematical theory, named the Theory of Nescience, grounded in the principles of computability, randomness, and artificial intelligence. The theory builds upon a striking insight: perfect knowledge implies randomness. Although this may initially appear counterintuitive, given that science is traditionally associated with structure, order, and explanation, it becomes clearer when we consider the process of theory refinement. As scientific understanding improves, our descriptions of phenomena become more concise. Eventually, a theory becomes so concise that it cannot be simplified any further. Its representation becomes incompressible, which aligns with the mathematical definition of a random string.

Randomness sets the ultimate limit to our knowledge, since a random description cannot be refined any further and, assuming the theory is accurate, our understanding must be perfect. Far from being a handicap, the constraints that randomness imposes on knowledge pave the way for new possibilities in science and technology. By comprehending these limitations, we can address the most challenging open problems and discover completely new research topics.

From this theoretical foundation emerges a new mathematical framework for understanding nescience, or human ignorance, and the processes by which knowledge is acquired, organized, and refined. Some of the contributions of this book are:

- A mathematical theory to quantify our lack of scientific knowledge, based on computability and randomness, and on the assumption that measuring how much we do not know is easier than measuring how much we do know.
- A new collection of metrics to measure how much we do not know, whether about individual research topics or broader research areas, and how new research contributions either reduce or fail to reduce that ignorance.
- A practical framework to address some of the most relevant challenges in science, such as defining what constitutes perfect knowledge, identifying the limits of science, and discovering new, previously unknown, research topics.

- A software library that combines the Theory of Nescience with principles of artificial intelligence to automatically derive new knowledge and make predictions based on data.

The Theory of Nescience challenges us to rethink the aims of scientific inquiry. Instead of striving to maximize what we know, it encourages us to minimize what we do not know. By shifting our focus from accumulation to reduction, we open the door to more meaningful measures of scientific progress. The chapters that follow lay the theoretical groundwork for this new approach, one that begins with ignorance and ends in knowledge.

## **Research Agenda**

In this section, we present a focused list of research questions aimed at addressing important gaps in our current understanding of science and the scientific method. Our goal is to stimulate thoughtful investigation by drawing from a variety of academic perspectives. By pursuing these questions, we hope to push the boundaries of human knowledge and inspire innovative solutions that can benefit both scientific inquiry and society more broadly.

*Can we provide a quantitative characterization of our ignorance regarding a research topic?* Developing such a metric would give us a way to measure not just what we know, but how much we are still missing. This would be especially useful for evaluating scientific contributions, as we could measure how much each new idea or publication actually improves our understanding. By combining this with relevance metrics, we would gain a powerful tool for assessing the true value of scientific work.

*Can we compare the extent of our ignorance across disparate scientific fields?* If this is possible, we could evaluate and prioritize research based on how much there is left to understand, regardless of the field. This approach could help distribute scientific attention and funding more effectively, highlighting areas of science that are in greater need of exploration while still valuing fundamental research.

*Can we establish a systematic procedure to enhance our knowledge?* While the scientific method has long served as the cornerstone of research, it varies significantly across disciplines and often lacks clear, measurable criteria for success. Developing a unified and precise approach could improve the consistency and efficiency of scientific discovery. If successful, it would allow us to evaluate and refine how knowledge is generated, leading to more reliable and accelerated progress across all areas of inquiry.

*Can we devise a method for discovering new, previously unknown, and intriguing research entities and problems?* Many major discoveries arise from questions we had not thought to ask. If we could build a systematic way

to identify these unknown unknowns, we could unlock entirely new areas of research and significantly accelerate scientific progress.

*What constitutes perfect knowledge?* Understanding what it means to know something completely is essential to defining the limits of science. If we could precisely define and recognize perfect knowledge, we would know when a field is complete and when it is time to shift our focus elsewhere. Clarifying whether such completeness is always possible would also help set realistic goals for scientific inquiry.

*What effort is required to fully comprehend an unfamiliar subject?* Measuring the cost of understanding could fundamentally change how we plan and fund research. For example, knowing the minimal effort needed to improve our grasp of cancer treatment by a small percentage would help us allocate resources more wisely. Even an estimate would be useful for setting priorities and managing expectations.

*Do some research topics inherently possess a higher degree of complexity than others?* This question addresses a long-standing debate about intellectual differences across disciplines. If we could objectively measure topic complexity, we might dispel myths of superiority and develop more targeted educational strategies. It would also help explain why some areas advance faster than others.

*Are there research topics beyond the scope of human comprehension?* Human cognition has limits, and it is possible that some problems are fundamentally beyond our capacity to solve. Acknowledging these limits could help us better delegate scientific exploration to machines where appropriate, especially as artificial intelligence becomes more capable of generating novel hypotheses or insights.

*How can we differentiate between science and pseudo-science?* Many fields present themselves as scientific but lack rigorous foundations. A mathematically grounded method for identifying what qualifies as science would have major implications for public policy, education, and research funding. It would also help protect scientific integrity by distinguishing valid inquiry from unsupported speculation.

Each chapter in this book engages with these questions from both theoretical and practical standpoints. Some offer detailed responses, while others lay the groundwork for future research. This openness is by design. The Theory of Nescience is not a closed doctrine but a platform for further exploration, an invitation to investigate the limits of science and to imagine new pathways of understanding. In this light, the book is not merely a treatise but a research agenda, a provocation to conventional thinking, and a call to those who are driven to explore the boundaries of knowledge itself.

---

## Origins of the Theory of Nescience

It was in 1991, when I was eighteen years old, that I first encountered the statement, "*Computers are useless, they can only give you answers.*" This quote, attributed to Pablo Picasso, struck me as profoundly true. Of course, Picasso was referring to the early calculators of his time, not modern computers. But the underlying idea remains relevant: machines are designed to process predefined tasks, not to generate new and meaningful questions on their own. That realization stayed with me for years. However, it wasn't until 2014, more than two decades later, that I began to explore the implications of Picasso's observation from a practical and computational perspective.

The core ideas behind the Theory of Nescience came together during one particularly restless night. Concepts such as nescience, relevance, and the unknown unknown suddenly aligned. In hindsight, my long-standing interests in information theory and Kolmogorov complexity had likely prepared me for this moment. These disciplines proved essential in articulating the mathematical foundation for a theory that had first emerged as a series of intuitive insights.

Over the following weeks, I conducted a series of computer experiments to test these concepts. The results were promising, but the theory needed time, several years, to develop into a rigorous mathematical framework. Initially, my goal was quite focused: to devise a method for identifying interesting and underexplored scientific questions. But as I examined how ignorance, or nescience, changes over time within scientific disciplines, my scope expanded. I began to wonder whether it was possible to define perfect knowledge and to determine whether such a state could be formally described.

This line of inquiry led to an unexpected and striking insight: perfect knowledge could be expressed in terms of randomness. Specifically, when a theory's description cannot be compressed further, when it becomes algorithmically random, it may indicate that the theory is as complete as possible. This realization broadened my original focus into a general framework for studying the structure of ignorance, the development of understanding, and the mechanisms that drive scientific progress.

Encouraging early feedback from colleagues and researchers helped me refine the theory and explore new applications, particularly in data science and machine learning. Its potential to shed light on long-standing questions in the philosophy of science was especially motivating. For instance, it offered new ways to compare our understanding across fields like mathematics and sociology, and to address the problem of how to distinguish between science and pseudoscience.

Continued mathematical development eventually led to a highly surpris-

ing discovery: in certain types of topics additional research can actually increase our ignorance. In these cases, there may exist a critical threshold beyond which further investigation does not improve understanding, but instead introduces greater uncertainty.

This deepened my belief that grappling with ignorance is as fundamental as pursuing knowledge. The Theory of Nescience represents my response to Picasso's challenge: not merely to construct systems that answer questions but to build frameworks capable of identifying which questions are most worth asking.

These developments and insights have shaped what has become a comprehensive and mature theory. One that I hope will not only reframe how we understand knowledge, but also inspire readers to question more deeply, think more broadly, and search more boldly. The chapters that follow present this journey in both its theoretical depth and practical implications.

## About the Book

The Theory of Nescience draws on concepts from multiple academic disciplines, including computability, complexity theory, artificial intelligence, and the philosophy of science. Despite the breadth of its foundations, this book is designed to be self-contained. Readers are expected to have only a basic understanding of first-year calculus and some experience with programming. The content is crafted to serve a wide technical audience, including mathematicians, computer scientists, engineers, and other scientifically inclined readers. The mathematical level is suitable for graduate students and advanced undergraduates.

The book is organized into three main parts: Foundations, Applications, and a collection of appendices on Mathematical Prerequisites. Readers who already have a background in the mathematics covered in the Mathematical Prerequisites may wish to begin directly with the Foundations. However, we recommend at least a brief review of the notation and key concepts introduced in those appendices. Once readers are familiar with the core ideas presented in the Foundations, they can proceed to the Applications. A detailed understanding of the underlying formalism is not essential; a general grasp of the main concepts and results is sufficient to engage with the practical examples and insights explored in that part of the book.

- *Chapter 1 Introduction* provides a gentle entry point to the theory of nescience, presenting a brief overview of its main concepts and results. While it avoids the use of advanced mathematics, the ideas are introduced in a semi-formal manner. Although it is not recommended as a substitute for the full theoretical exposition, readers who find the

mathematics challenging may choose to read only this chapter before proceeding directly to the Applications.

*PART I Foundations* presents a comprehensive account of the theory of nescience, including formal definitions of its core concepts and proofs of key theoretical results. This section forms the core of the book. Readers with prior knowledge in computability, complexity, information theory, probability, and artificial intelligence may choose to begin here directly.

- *Chapter 2 Fundamental Elements* includes the initial step toward quantifying our lack of knowledge, which involves the precise identification of the research entities under examination, determining how to represent them as strings of symbols, and identifying suitable models to explain them. The chapter introduces these fundamental components of the theory of nescience: entities, representations, and descriptions. It examines their properties and the relationships among them. The chapter also discusses how various representations and descriptions can be combined, how background knowledge influences research, and explores the link between perfect knowledge and randomness. It concludes by proposing a novel concept of a research area.
- *Chapter 3 Miscoding* explores the challenge of representing both abstract and concrete research entities as strings of symbols for research purposes. It formally introduces the concept of miscoding and examines its theoretical properties. Miscoding serves as a measure of the error introduced by inaccurate or inappropriate encodings of the entities being studied. The chapter also discusses strategies for minimizing the errors introduced by poor representations, thereby improving the accuracy and utility of symbolic encodings in research.
- *Chapter 4 Inaccuracy* presents a new interpretation of the classical concept of error, specifically how accurately a description reflects an entity. The concept is generalized to apply to a wide range of topics, including abstract ones. The chapter also introduces a conditional variant of inaccuracy and examines its theoretical properties. It investigates techniques to reduce inaccuracy and explores how miscoding (errors resulting from poor representations) influences the degree of inaccuracy observed.
- *Chapter 5 Surfeit* investigates the redundancy present in a description, specifically how many unnecessary elements it contains. Surfeit serves as an indicator of how well we currently understand research topics and areas, since our lack of knowledge about the entity is typically reflected in the length of our prevailing description. The chapter also introduces a conditional variant of surfeit and derives a practical approximation

that can be applied in real-world scenarios. Finally, it examines the relationship between inaccuracy and surfeit.

- *Chapter 6 Nescience* serves as the core of the book, presenting the fundamental elements of the theory. It defines science as a nonlinear multiobjective optimization problem in which the conflicting metrics of miscoding, inaccuracy, and surfeit are minimized simultaneously. A new metric, nescience, is introduced as a function of these three components. The chapter also explores key properties of this metric, including its evolution over time, the concept of perfect knowledge (zero nescience), and methods for identifying our current best model. In addition, it introduces a definition for the frontier of human knowledge and offers a characterization of what lies beyond that boundary.
- *Chapter 7 Interesting Questions* presents a methodology for identifying new research ideas, based on combinatorics and computational creativity, and focusing on how to address challenging open problems. It introduces two new metrics to measure a topic’s relevance and its applicability to existing problems, and examines the properties of these metrics. The chapter also outlines a systematic approach to uncovering what lies hidden in the unknown unknown, illustrating how previously unrecognized research directions may be revealed.

*PART II Applications* presents a collection of practical uses of the concept of nescience in areas such as machine learning, the philosophy of science, and the discovery of new research topics. The included examples have been selected to illustrate the broad applicability of the theory, spanning from abstract research questions to more tangible problems grounded in datasets.

- *Chapter 8 Machine Learning* explores how the theory of nescience can be applied to entities represented by collections of measured samples. We introduce `mnplib`, a software library that can be used to analyze datasets, select relevant features, identify optimal model hyperparameters, and compute the errors of trained models. In addition, the chapter presents novel machine learning algorithms, including an innovative method for deriving optimal decision trees and automating the construction of machine learning models.
- *Chapter 9 Analysis of Science* investigates how well we understand current research topics by applying the proposed metrics. Our aim is to assess the degree of understanding across different areas of science. To this end, we compare research topics within the same academic discipline, as well as across multiple disciplines. The chapter also addresses major open questions in the philosophy of science, including the demarcation problem (how to distinguish science from pseudoscience)

and the nature of scientific progress.

- *Chapter 10 The Discovery of the Unknown* demonstrates the practical application of the theory of nescience for identifying promising research questions. Specifically, we show how nescience can be leveraged to generate new research ideas aimed at solving the most difficult open problems. We also propose a methodology for identifying new research topics—that is, for uncovering what lies hidden in the unknown unknown. Multiple examples of such research questions and novel research areas are provided.

*Appendix Mathematical Prerequisites* introduces the mathematical foundations necessary to quantify our lack of knowledge about a research topic and to assess the randomness of a string. Its primary aim is to establish consistent notation, formally define key concepts, and present important theoretical results. While no prior background is required to follow the material, readers are encouraged to consult the standard references provided at the end of each chapter for deeper understanding.

- *Appendix A Discrete Mathematics* offers a summary of the fundamentals of discrete mathematics needed to understand the more advanced topics discussed in the book. This chapter serves as a quick review of these concepts without providing formal definitions or proofs. Topics covered include sets, relations, strings, graphs, and counting methods. A section on linear algebra (matrices and vectors) is also included.
- *Appendix B Discrete Probability* introduces the foundational concepts of probability related to discrete events. Topics covered include conditional probability, random variables, distribution characterization, common distributions, and large random samples. This chapter aims to equip readers with the necessary background in probability to understand the more advanced statistical learning discussed later in the book.
- *Appendix C Computability* presents a formal definition of the concept of algorithm. It introduces the idea of a universal Turing machine and shows that certain well-defined mathematical problems cannot be solved by computers. The chapter also examines the essential tool of an oracle Turing machine.
- *Appendix D Coding* explores how codes function and how they enable us to compress texts by eliminating redundant patterns without losing essential information. It shows that there is a limit to how much a text can be compressed using this technique, and that this limit is determined by the entropy of the source. The important relation between optimal codes and discrete probabilities is also studied.

- *Appendix E Complexity* introduces an absolute metric known as Kolmogorov complexity, which measures the amount of information contained in a string by calculating the length of the shortest computer program that can produce it. The chapter studies the properties of this metric in detail and explores the relationship between string complexity and randomness.
- *Appendix F Learning* offers a concise overview of the field of statistical learning, presenting key results from statistical inference and machine learning. It also examines the relationship between codes and probabilities, focusing on practical approaches that apply the concept of minimum string length. Additionally, the chapter introduces the concepts and notation associated with nonlinear multiobjective optimization problems.
- *Appendix G Philosophy of Science* provides a brief introduction to the study of science from a philosophical perspective. It reviews key concepts such as scientific representations, models, and theories, and identifies the essential components that any formal theory of science should include. The chapter also offers an overview of our current understanding of how the scientific method works.

## Acknowledgements

I would like to express my gratitude to everyone who has contributed their comments and ideas to the development of the theory of nescience. In particular, I am grateful to Antonio Fernández, Vincenzo Mancuso, and Paolo Casari, who believed in and supported this project from its very beginning when it was merely a far-fetched idea. Others who have provided contributions and valuable feedback include Héctor Cordobés, Luis F. Chiroque, Agustín Santos, Marco Ajmone, Pablo Rojo, Manuel Cebrián, Andrés Ortega, Emilio Amaya, Mattis Choummanivong, Alexander Lynch, Andrés Carrillo, Simon Bihoreau and Ricardo Hernando. Early versions of the `mnp1ib` library described in Chapter 8 have been partially funded by the IMDEA Networks Institute, the European Union’s Horizon 2020 research and innovation programme under grant agreement No 732667 RECAP, and Nokia Spain through the project NetPredict.

The ideas presented in this book build on a long history of thought. I draw inspiration from philosophers such as Plato, Occam, Popper, Feyerabend, and Wittgenstein; from polymaths like Llull, Leibniz, and Newton; and from the mathematical foundations established by Turing, Church, Post, Shannon, Solomonoff, Chaitin, Kolmogorov, and many others. My contribution lies in connecting and reinterpreting some of these insights within a unified

---

framework. The References section at the end of each chapter highlights the sources that have most directly influenced my thinking.

Throughout the text, I use the passive voice ("it is defined") when referring to concepts whose origins I recognize, and the active voice ("we define") when I am unaware of any prior formulation.

I would like to express my sincere gratitude to the open source community. This book would not have been possible without the extensive use of open source tools, which provided both the technical foundation and the flexibility required throughout its development. From the writing process, carried out using `TeX`, `LATEX`, and `TikZ`, to the data analysis and machine learning experiments—powered by Python and libraries such as `scikit-learn`, `statsmodels`, `pandas`, `NumPy`, and `Matplotlib`, the contributions of countless developers and researchers who share their work openly have been indispensable. Their dedication to building and maintaining high-quality, freely available software is a testament to the collaborative spirit of scientific progress.

I would also like to acknowledge the role of ChatGPT in the preparation of this book. As an AI assistant, ChatGPT provided valuable support throughout the review process, helping to refine ideas, clarify language, structure arguments, and polish technical content<sup>1</sup>. While all decisions and final content were ultimately my own, the ability to engage in thoughtful dialogue, receive constructive feedback, and explore alternative phrasings and formulations greatly enhanced the clarity and coherence of the work. I am grateful for this tool, which has proven to be both efficient and intellectually stimulating.

Finally, I am deeply grateful to my parents for providing me with opportunities they themselves never had, and to my wife and three children, who give my life its deepest meaning.

---

<sup>1</sup>The following prompts have been used in the review process:

- Review the English and clarity of the following text.
- Rewrite the following paragraph in a better English.
- Review the following section and find logical inconsistencies and mathematical errors.
- Draw a figure in `Tikz`.





# 1. Introduction

*If presented with a choice between indifferent alternatives,  
then one ought to select the simplest one.*  
Occam's razor principle

We find ourselves in an age where knowledge has become one of our most valued resources, driving both economic progress and societal well-being. Science continually provides us with deeper insights and practical solutions, shaping our daily lives through innovations ranging from advanced medical therapies to transformative digital technologies. However, the path of scientific exploration is neither straightforward nor free from significant barriers. Despite remarkable progress, our current scientific methodologies remain bounded by inherent limitations that restrict our ability to fully understand and address complex issues.

These limitations manifest in various ways, including fragmented research efforts, rigid disciplinary boundaries, and an often narrow approach to funding allocation, which tends to overlook bold, high-risk projects with transformative potential. Moreover, the conventional metrics of scientific success frequently incentivize incremental progress rather than genuinely groundbreaking discoveries, imitating the pace and scope of innovation.

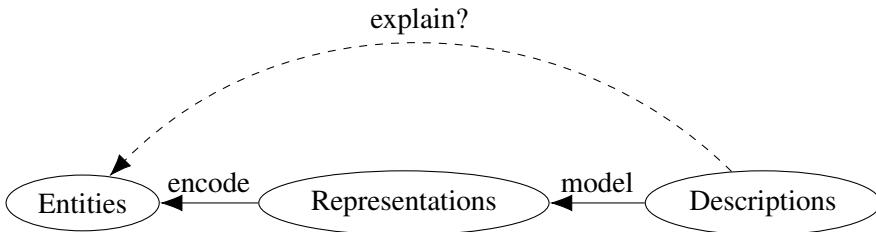


Figure 1.1: The Problem of Understanding

The *theory of nescience* emerges from the critical recognition of these challenges and the urgent need to overcome them. It advocates a renewed approach to exploring the unknown, driven by curiosity and the willingness to question accepted knowledge rigorously. By systematically addressing the limitations inherent in contemporary scientific practice, our objective is to provide robust frameworks that significantly enhance our capability to solve real-world problems and catalyze meaningful advancements across diverse fields.

## 1.1 Scientific Knowledge

The pursuit of knowledge begins with identifying the *entities* we seek to understand, entities that are often extraordinarily diverse. Mathematicians focus on abstract concepts, biologists on living organisms, and engineers on machines. Our quest for understanding is fundamentally driven by our need to predict outcomes, solve problems, and navigate the complexities of the world around us. For instance, we know that applying sufficient heat to wood will ignite a fire, or that a fever may indicate a viral infection. Practical problem-solving, therefore, hinges crucially on our ability to recognize patterns and create simplified models of these entities. These models empower us to interact beneficially with the world.

Yet, while ideally we strive to build precise *descriptions* capable of fully reconstructing the entities we study, in reality, such perfection often eludes us, particularly with abstract entities. Consequently, we rely on *representations*, texts or datasets capturing essential details of these entities. Physicists might represent an entity through experimental results; computer scientists through measured data; sociologists through observed facts. Figure 1.1 illustrates this critical interplay: although we aim to directly describe entities, practically, our descriptions are based on constructed representations.

Understanding the effectiveness of our descriptions involves examining potential errors deeply. Firstly, encoding methods can introduce errors,

which we call *misCoding*. Secondly, our descriptions themselves may fail to precisely recreate the encoded representations, resulting in *inaccuracy*. Finally, due to human cognitive limitations, descriptions should not become unnecessarily complex, introducing *surfeit*.

Collectively, misCoding, inaccuracy, and surfeit hinder our quest for clear understanding and reliable predictions. By amalgamating these three error types into a single measure, known as *nescience*, we quantify our lack of knowledge, providing an essential tool for systematically addressing and reducing our ignorance.

## 1.2 Entities

At the core of our theory of nescience is the recognition that science is, at its essence, a quest to understand the world around us. Throughout history, scientists have examined an extraordinary variety of things (planets, particles, languages, ecosystems, human societies, and much more) in an effort to uncover patterns, formulate explanations, and make predictions. These things we seek to understand, which we refer to as *entities*, form the basis of all scientific activity. An entity might be a tangible object, such as a chemical compound or a cell, or something more abstract, like a mathematical function or a cultural practice.

The scope of what science might investigate is vast and continually evolving. New technologies, shifting societal needs, and fresh philosophical insights regularly bring new entities into view. What unifies this effort is a foundational belief: that some of these entities can be understood through science. That is, we hold that at least part of the unknown is ultimately knowable (see Section G.2). This belief fuels the scientific drive to bring clarity to what was once obscure, to shed light on complexity, and to transform speculation into knowledge. Understanding how we fall short of this ideal, how ignorance persists and why, is the starting point for the theory of nescience.

One important conceptual difficulty we face is that the set of all entities under consideration, denoted  $\mathcal{E}$ , cannot be rigorously defined in mathematical terms—other than requiring it to be non-empty. This may seem like a minor point, but it carries deep implications.

In mathematics, the idea of a "set of everything" is fraught with contradictions. If we tried to construct a set that included absolutely all things (physical objects, abstract ideas, ...) we would quickly encounter logical problems. This is why we deliberately avoid the notion of a universal set in the theory of nescience. A key reason for this caution lies in Cantor's theorem, which we discuss further in Section 2.1. Cantor's result shows that

for any set, the collection of all its subsets is strictly larger in size than the set itself. As a consequence, it becomes impossible to form a set that includes everything without running into contradictions. Similarly, we steer clear of problematic constructions that lead to paradoxes, such as Russell's paradox, another example covered in Section 2.1, which illustrates how self-referential sets can collapse logical consistency.

These mathematical constraints are not mere formalities; they serve a vital purpose. By enforcing clear boundaries around the sets of entities we analyze, we ensure that the theory we are building remains coherent, consistent, and applicable. It allows us to focus our attention on meaningful, well-defined domains where real progress in understanding can be made.

In practice, we will work with well-defined sets, each associated with a specific domain of inquiry and its unique goals. These sets provide a practical framework for applying our theory to real-world contexts. For example, in mathematics, such a set might include different classes of abstract structures such as groups, functions, or topological spaces. In biology, it could encompass the vast diversity of living organisms, from microscopic bacteria to complex multicellular animals. In the realm of social sciences, the entities might include human behaviors, social systems, or economic models. And in computer science, we may focus on algorithms, data structures, and executable programs.

By tailoring our analysis to these different sets, we are able to apply a unified theoretical framework to a wide variety of disciplines. This adaptability is one of the strengths of our approach: it allows us to measure and reduce human ignorance, or nescience, in fields with very different kinds of entities. Our goal is not only to improve our theoretical understanding of these domains but also to contribute practical tools that can support deeper insights, better decision-making, and more effective problem-solving across the sciences and beyond.

### 1.3 Representations

In many instances, entities cannot be directly scrutinized through scientific analysis, particularly if they are abstract. As a result, we are compelled to rely on representations, i.e., symbolic encodings that stand in for the entities we aim to understand. We designate the collection of strings that encode the entities of  $\mathcal{E}$  as  $\mathcal{R}_{\mathcal{E}}$ . These strings, referred to as *representations*, may differ depending on the application of the theory of nescience. In certain scenarios, entities may inherently be string-based (e.g., computer programs), while others might be abstract objects that require encoding into string format (e.g., human needs). It is not uncommon for a single entity

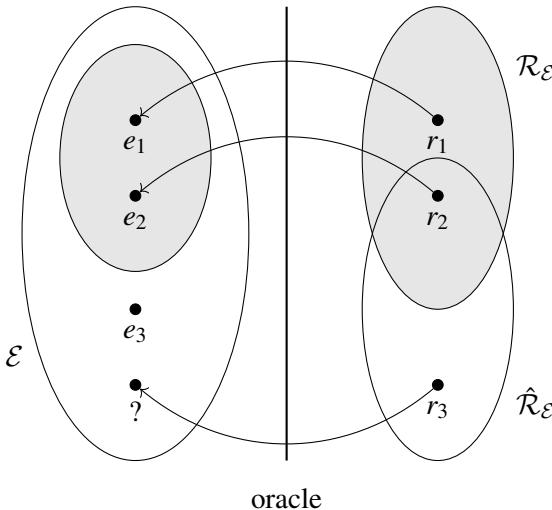


Figure 1.2: Entities and Representations.

$e \in \mathcal{E}$  to have multiple valid representations within  $\mathcal{R}_{\mathcal{E}}$ , for example, a text-based description, a diagram saved as a computer file, or a collection of empirical measurements. Each representation emphasizes different aspects of the entity and may be better suited to particular investigative goals or scientific approaches. Transforming abstract entities into symbol strings in a manner that faithfully captures their complexities and subtleties remains a formidable and unresolved challenge. Consequently, the exact composition of  $\mathcal{R}_{\mathcal{E}}$  often eludes us.

In a world where understanding depends on symbolic surrogates, the idea of constructing an encoding function  $f : \mathcal{E} \rightarrow \mathcal{R}_{\mathcal{E}}$  becomes not only attractive but essential. The function  $f$  represents an idealized encoding process: it assigns to each entity  $e \in \mathcal{E}$  one of its symbolic representations  $r \in \mathcal{R}_{\mathcal{E}}$ . In essence, this function models the act of transforming something we wish to understand (like a physical object, a biological system, or an abstract concept) into a format that can be studied, manipulated, or stored using symbols, typically as strings. In practice, such a function would allow us to systematically move from the domain of real-world or conceptual entities to their formal encodings, which are necessary for analysis in science, computation, and communication. However, defining such a function is no trivial task, precisely because  $\mathcal{E}$  itself is not well defined. The boundaries of what should or should not be included in  $\mathcal{E}$  are inherently blurred. For example, we still lack a precise and universally accepted definition of what constitutes a human need.

To grapple with this indeterminacy, one may resort to theoretical con-

structs such as the *oracle Turing machine* (see Appendix C). While a standard *Turing machine* serves as a mathematical model of computation, the oracle variant introduces a conceptual leap: it simulates a computer with access to an external source of information. The oracle Turing machine can be imagined as a theoretical computer connected not to the actual internet of today, but to an idealized version, an omniscient information source containing perfect knowledge about everything that exists or could exist. This imaginary machine is allowed to submit string-based queries to this vast external database. For instance, it might ask whether a given string  $r$  encodes any entity in  $\mathcal{E}$ . Unfortunately, formulating the question “does  $r$  represent  $e$ ?” would require expressing  $e$  itself as a string of symbols. And since we do not know how to encode  $e$  in advance, we cannot construct such a query. This irony captures the very dilemma we aim to address, the tension between what can be queried, what can be known, and the inherent limitations of representation in scientific inquiry.

From a practical perspective, we typically approximate the set  $\mathcal{R}_{\mathcal{E}}$  with another set  $\hat{\mathcal{R}}_{\mathcal{E}}$  of strings, which we consider to be adequate representations of the entities of  $\mathcal{E}$ . In scientific practice, these representations have traditionally taken the form of illustrations or images (biology), collections of factual data (sociology), or experimental results (physics). With recent significant advancements in the capability of computers to gather and store data, a novel and powerful method has emerged: encoding entities using vast datasets. It’s essential to note that in the encoding process, our objective is not to find the shortest possible representation of the entities but to seek out high-quality representations.

It’s crucial to acknowledge that in numerous practical scenarios, the chosen representations of abstract entities may not fully encapsulate all nuances of the original objects. This means we are grappling with simplified abstractions of reality, which could potentially curtail our capacity to make sweeping assertions about nature (see Chapter 3).

■ **Example 1.1** If we’re studying animals (the set  $\mathcal{E}$ ), we could use a binary encoding of their DNA (the set  $\mathcal{R}_{\mathcal{E}}$ ) as representations. While our current technology doesn’t allow us to bring a creature to life solely from its DNA, theoretically, this could one day be feasible. However, DNA alone doesn’t fully replicate the original animal, as it doesn’t include life experiences. For instance, how would we represent a cat that only has three legs due to an accident? That detail is not recorded in its DNA. If our goal is to study the traits of certain species, working with the DNA of a representative sample of individuals within each species would be adequate. However, if we’re studying specific individuals within a species, we would also need a way to encode each animal’s history or the details not encapsulated by the DNA. ■

Working with strings as representations (the set  $\mathcal{R}_{\mathcal{E}}$ ) inevitably results in certain entities lacking any corresponding encoding (see the gray areas in Figure 1.2; specifically, entity  $e_3$  has no representation). This limitation becomes particularly evident when the set of entities is uncountable. For example, if  $\mathcal{E}$  is the set of real numbers, many elements cannot be represented because representations are restricted to finite binary strings. Real numbers that require infinite precision (such as most irrational numbers) cannot be fully encoded. This mismatch reveals a deeper asymmetry: in many domains of knowledge, the space of conceivable problems or entities far exceeds the space of valid, encodable representations. Intuitively, this suggests that in such domains, the quantity of problems may exceed the number of solutions.

Using approximations of representations (the set  $\hat{\mathcal{R}}_{\mathcal{E}}$ ) can also result in some representations encoding the wrong entities, as illustrated by representation  $r_3$  in Figure 1.2. This occurs because our knowledge about the entities in  $\mathcal{E}$  is often incomplete or imprecise, which can lead us to construct representations that appear valid but fail to correspond to the intended entity.

Another issue with incomplete knowledge is the possibility of having entities whose existence is unknown to us. For instance, representation  $r_1$  in Figure 1.2 is not part of the set  $\hat{\mathcal{R}}_{\mathcal{E}}$  and is therefore overlooked by researchers despite being the representation of a knowable entity  $e_1$ . One of the objectives of this book is to provide a procedure to uncover new, previously unknown, research entities from the set  $\mathcal{R}_{\mathcal{E}}$  (refer to Chapter 10).

## 1.4 Descriptions

Upon identifying the set  $\mathcal{R}$  of potential representations, we are faced with the deeper motivation that drives much of scientific inquiry: the desire to bring order to the complexity of the world. To do this, we must devise appropriate methods to describe these representations, an endeavor that underlies our attempt to articulate, through theories and models, how the world operates. Our limited cognitive capacities as humans compel us to work with simplified, yet insightful, models of nature. These abstractions help us interpret phenomena and forecast the consequences of our actions. Descriptions also change over time, as our understanding of the entities studied improves.

■ **Example 1.2** To anchor these ideas in the concrete, consider the evolving effort to describe the macroscopic behavior of the physical universe (the entity  $e$ ). The sequence of proposed descriptions includes Aristotelian physics, Cartesian mechanics, Newtonian laws, Einstein's relativity, and, potentially, string theory. Each successive model attempts to refine our grasp of reality. Among these, Einstein's theory currently stands as the most complete, since

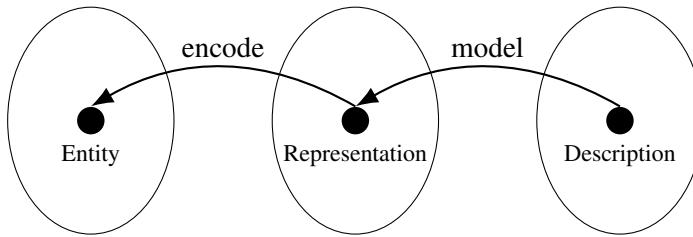


Figure 1.3: Entities, representations, and descriptions.

alternatives such as string theory still await experimental corroboration. ■

Defining a valid description for an entity is not merely a technical challenge; it embodies a fundamental limitation on knowledge. The Berry paradox serves as a compelling reminder of these philosophical intricacies. A phrase like "the smallest positive integer not definable in fewer than twelve words" becomes paradoxical by defining itself in fewer words than allowed. To avoid such pitfalls, the theory of nescience imposes stricter demands: a valid description must be a finite symbol string that allows us to reconstruct effectively and completely a possible representation of the original entity. By "effectively," we mean that this reconstruction can be performed by a computer without human intervention.

From Newton's formulation of classical mechanics to today's explorations, the scientific journey has been shaped by the pursuit of mathematical models. The theory of nescience follows this lineage but extends it by demanding that models be computable, i.e. executable by computers. This requirement that descriptions be computable allows us to avoid many of the paradoxes traditionally associated with the concept of description. In this light, science becomes not only a quest for understanding but also a computation-driven endeavor to approximate reality.

Descriptions are divided into two components: a Turing machine  $TM$  (a computer program) that encapsulates all the regularities found in the entity's representation (the compressible part), and an input string  $a$  that contains a literal description of the remaining elements (the non-compressible part). This dualistic nature of descriptions parallels traditional distinctions in science, such as theories and assumptions, models and initial conditions, problems and specific problem instances, species and individuals, and so on. For instance, a description might consist of a system-modeling set of differential equations (the compressible part), accompanied by a compilation of initial conditions (the non-compressible part). The precise interpretation of the pair  $\langle TM, a \rangle$  relies on the specific characteristics of the entity set to which the theory is applied.

Figure 1.3 illustrates the relationship between entities, representations, and descriptions. The set of all potential descriptions is denoted by  $\mathcal{D}$ . However, not all strings qualify as valid descriptions: each must be grounded in a Turing machine, ensuring that it is computationally meaningful. Furthermore, not every valid description corresponds to a legitimate representation. Since representations  $r$  can be described in multiple ways, the overarching scientific goal emerges as the search for the shortest possible description  $d$  that faithfully reconstructs the observed data.

However, a fundamental obstacle lies in the incomputability of this task. As discussed in Appendix E, there exists no general procedure to determine the shortest program that outputs a given string. This impossibility extends to representations, rendering the pursuit of optimal scientific models as a challenge beyond computer capabilities. As a result, science must rely on heuristic strategies to approximate ideal solutions. Collectively, these heuristics define what is called the *scientific method*.

The theory of nescience is driven by the desire to understand, and ultimately quantify, the various errors that arise in the process of scientific discovery. Figure 1.3 provides the conceptual framework for this endeavor. In Sections 1.5, 1.6, and 1.7, we introduce metrics designed to capture distinct sources of error. These components are then synthesized in Section 1.8 into a single, unified measure: nescience. Although inherently uncomputable, this measure formally expresses the extent to which a given research entity remains poorly understood. It highlights both the boundaries of current knowledge and the domains most deserving of scientific exploration.

## 1.5 Miscoding

As we've observed, in many scientific disciplines, the effort to understand the world often begins with an elusive challenge: the entities we wish to study, denoted as the set  $\mathcal{E}$ , do not always lend themselves to clear or complete representations. Some of these entities are too abstract, others too complex, and many remain partially known. This mismatch between the reality we wish to grasp and the means we have to represent it is not merely a technical limitation; it reflects the very heart of scientific inquiry.

The scientist's journey, then, often starts in uncertainty. We make do with approximations, crafting descriptions that we hope capture enough truth to be useful. Yet, we are aware that these representations carry errors. Our goal becomes not just to encode entities, but to quantify the error introduced by using these inaccurate representations.

We propose to measure the *miscoding* of an inaccurate representation  $r'$  by assessing how difficult it is to transform this flawed representation

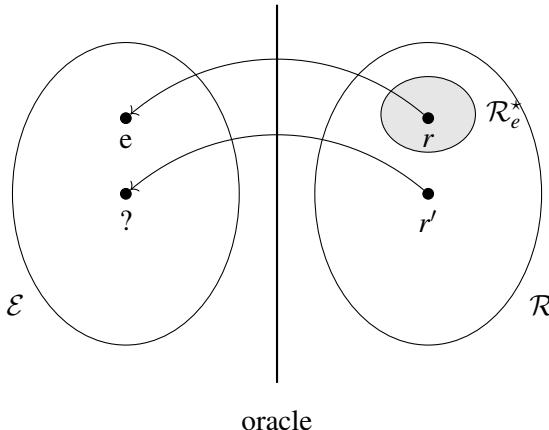


Figure 1.4: Miscoding of representations.

into a correct one. In more technical terms, this difficulty is expressed as the length of the shortest computer program that, when given  $r'$  as input, is able to produce the accurate representation  $r$ . Figure 1.4 provides a visual depiction of this process. The set  $\mathcal{R}_e^*$  consists of all the strings that are considered accurate representations of the entity  $e$ . If  $r'$  falls outside this set, it means our understanding is flawed. Miscoding represents the cost of bridging that gap. Importantly, this cost is not about computational time or speed. It is about *descriptive complexity*, how much must be specified, or programmed, to repair the mistakes. If  $r'$  contains inaccuracies, the required program must identify these deviations and correct them. If instead  $r'$  lacks key information altogether, the missing content must be embedded within the program itself. In this way, miscoding becomes a reflection of our ignorance: the larger the program needed, the more knowledge must be added to arrive at the truth. Miscoding measures how much we still need to learn before our representations truly encode the entities we seek to understand.

However, the previous method of measuring miscoding does not fully capture our intuitive understanding of what it means for a representation to be flawed. There is a deeper issue that emerges when a representation includes more than what is necessary. In other words, it may not only be inaccurate by omission but also through addition. Consider a situation where the representation  $r'$  contains extra information that is irrelevant or unrelated to the entity  $e$  we are trying to understand. This surplus information is not just noise; it actively distorts the description by forcing any model based on  $r'$  to account for elements that have no bearing on the actual entity. The result is a bloated and misleading representation. The description becomes longer not because the entity itself is more complex, but because

our flawed representation includes unnecessary baggage. This problem is not just theoretical; in real-world scientific practice, we frequently encounter such scenarios. Imagine an experiment where multiple variables are being recorded, but only a few of them actually influence the outcome. If we do not yet know which variables are relevant, our current models might treat all recorded features as potentially significant. This lack of understanding can lead us to build explanations and predictors around elements that are, in reality, unrelated to the entity or phenomenon of interest.

To account for this kind of misrepresentation, we must expand our definition of miscoding. We need to ask not only how much effort is required to fix an inaccurate representation, but also how much effort it would take to reconstruct that flawed version from the correct one  $r$ . The higher this effort, the worse the representation, as it suggests that the inaccurate description deviates significantly from what is accurate. This leads us to introduce a second measure, as the length of the shortest computer program that can output the incorrect description  $r'$  given the accurate one  $r$ . Only by considering both directions, how difficult it is to go from  $r'$  to  $r$  and how difficult it is to go from  $r$  to  $r'$ , can we fully assess the degree of miscoding. We therefore define miscoding as the maximum of these two program lengths. This revised definition acknowledges that misinformation can come in multiple forms. It captures both the missing and the misleading, recognizing that a poor representation might not only fail to say what is necessary, but might also say unnecessary things. In this way, miscoding becomes a more complete reflection of the divergence between what we currently believe and what truly is the case.

Nevertheless, this latest definition still poses practical challenges. In many cases, the same entity can be described in multiple, equally valid ways. This multiplicity creates a dilemma: what should we do when our inaccurate representation  $r'$  is far from one correct representation  $r_1$ , but quite close to another valid one  $r_2$ ? Judging solely by its distance to  $r_1$  could unfairly suggest a high level of miscoding, when in fact  $r_2$  may be a legitimate approximation of  $e$ .

The core issue here is that correctness is not always unique. Scientific and mathematical entities often admit many forms of expression—each highlighting different properties, or suited for different contexts. Penalizing a representation for not being similar to just one of these correct forms would ignore the richness and flexibility of representation.

■ **Example 1.3** Consider  $e$  as the abstract entity corresponding to the mathematical constant  $\pi$ , the ratio of a circle's circumference to its diameter. Let  $r$  be the Wallis product, expressed as  $2(\frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \dots)$ , a well-known infinite product that converges to  $\pi$ . Suppose  $r'$  is the infinite series

$4(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots)$ , which corresponds to the Leibniz series. Although  $r'$  is structurally very different from  $r$ , it also accurately represents the same entity  $e$ . To declare  $r'$  as highly miscoded with respect to  $r$  would be misleading, because  $r'$  is not an error; it's an alternative, equally correct expression.

■

As example 1.3 indicates, defining miscoding poses challenges because the set  $\mathcal{R}_e^*$  of valid representations for the entity  $e$  is generally unknown. Theoretically, we could rely again on the oracle Turing machine to solve this problem. However, as observed, we can't ask the oracle if the string  $r$  is a valid representation of the entity  $e$  we are interested in (the set  $\mathcal{R}_e^*$ ), since that would require us to provide a valid encoding of  $e$  as a string of symbols, which typically can't be done. Perhaps, all we can do is ask this oracle how close a given string  $r$  is to being a valid description of some entity in the entire set of entities  $\mathcal{R}_e^*$ .

With this constraint in mind, we have to define the miscoding of a representation as the smallest possible discrepancy, as judged by the oracle, between our given string and a valid representation of some entity. In this way, we allow the oracle to search through the universe of all correct representations, looking for the one that is most similar to our candidate. This definition enables us to talk about miscoding even in the absence of a known ground truth.

But there is one final complication we must confront. Because this definition of miscoding does not rely on knowing the actual entity being represented, it opens the door to a subtle yet critical problem: we might not be representing what we think we are. In other words, our descriptions might be well-formed and internally consistent, yet point to an entirely different entity than we had intended. This kind of mistaken identity is not just a philosophical curiosity; it has occurred repeatedly in the history of science. Researchers have often believed they were investigating one phenomenon, only to later discover that their results pertained to something entirely different (see Figure 1.4).

■ **Example 1.4** In 1961, the Soviet physicist Nikolai Fedyakin conducted experiments that appeared to reveal a new form of water, later called polywater. This substance was reported to exhibit unusual properties, such as a higher boiling point, a lower freezing point, and increased viscosity. Subsequent investigations showed that these observations were due to contamination by trace impurities, and that polywater did not correspond to a distinct entity. ■

According to the theory of nescience, our role as researchers goes beyond the mere identification of correct representations for the entities we wish to understand. It also involves a deeper and more ambitious task: uncovering

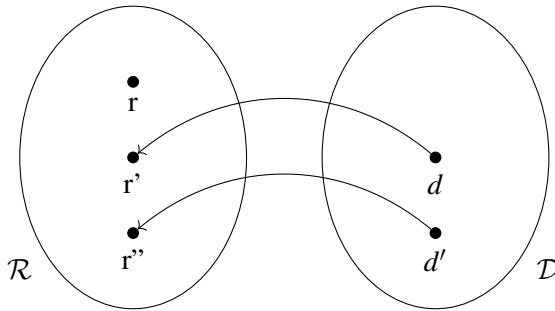


Figure 1.5: Inaccuracy of descriptions.

the principles by which an idealized oracle machine would reconstruct the original entities given their representations. In other words, it is not enough to arrive at accurate representations; we must also strive to understand why these representations are accurate, what makes them effective, and how they reflect the intrinsic structure of the underlying entities, by understanding how this abstract oracle would work. This shift in perspective moves us from a practice of isolated trial and error to a more reflective inquiry into the nature of scientific representation itself. Our goal is not merely to find a representation that works, but to understand more deeply the process by which it encodes reality.

## 1.6 Inaccuracy

In the preceding section, we explored how ignorance may arise when the representation  $r'$  we use to encode an entity  $e$  does not match the actual entity's correct representation  $r$ . That was the problem addressed by miscoding. Now, we turn to another, equally important source of ignorance, what we call *inaccuracy*, which emerges not from selecting the wrong representation, but from failing to describe it adequately.

In an ideal scenario, we would have a description  $d$ , that is, a computer program, capable of fully reconstructing the representation  $r'$ , even if the true representation  $r$  remains unknown. However, this level of precision is rarely attainable in practice. More commonly, we rely on an approximate description  $d'$ , which produces a string  $r''$  that resembles  $r'$ , but is not exactly the same. In such cases, we refer to  $d'$  as an *inaccurate* description of the representation  $r'$  (see Figure 1.5).

If a description is inaccurate for a representation, it is useful to have a quantitative measure of how much we deviate from accurately modeling the representation. A viable way to define this measure is to calculate the effort needed to rectify the output of our inaccurate description. In this

context, the inaccuracy could be determined by the length of the shortest computer program that can generate the correct representation when fed with the incorrect one produced by the description. However, similar to the case with miscoding, to have a holistic understanding of the error associated with the description  $d$ , we must also calculate the difficulty of generating the inaccurate representation given the correct one. It is possible that our description  $d$  models elements unrelated to the representation  $r'$ , and merely ignoring these elements won't solve the problem.

In other words, the inaccuracy measures how difficult it is to convert the output of the description into the intended representation, and the other way around, how difficult it is to convert the representation into the output of our description. The larger this maximum value, the more the description deviates from accurately capturing the intended representation.

We deliberately prefer the term *inaccuracy* over the term *error*. In the terminology of measurement theory or continuous systems, "error" encompasses both precision and accuracy. But in our discrete framework, where descriptions are finite symbol strings, precision loses its relevance. What matters here is how well the structure of the description aligns with the structure of the representation.

In practice, calculating the inaccuracy associated with the description of a representation is a challenging task. As previously mentioned, determining the length of the shortest computer program that can print a string is a non-computable problem. If the original entities are texts, we can estimate inaccuracy using compression algorithms, where string complexity is approximated by the length of the compressed text. If the entities under study are empirical rather than textual, such as experimental observations, their descriptions could be derived from experimental results. Hence, the inaccuracy could be based on the model's error (for instance, by calculating the length of additional information required to thoroughly describe the experiment's results given the model). In this regard, our definition of inaccuracy is a generalization of the concept of error. It can be applied to various types of entities, not only those that can be encoded as datasets.

■ **Example 1.5** Consider Newton's second law of motion,  $F = ma$ . Suppose we construct a dataset by applying known forces to objects of varying masses and measuring their resulting accelerations. If our goal is to study gravitational acceleration, the force and mass terms cancel, isolating acceleration as the variable of interest. Encoding this dataset in full would require a significant number of bits. Yet, recall that our objective with representations is not to minimize string length but to ensure that the encoding captures the richness and structure of the underlying phenomenon.

In this example, we draw on a historical experiment conducted by the

National Bureau of Standards in Washington D.C. between May 1934 and July 1935. The dataset includes 81 measurements of acceleration in centimeters per second squared, for instance, a value like 980.078. Using a uniform 20-bit encoding per measurement, the full dataset requires 1,620 bits. Suppose a model predicts a gravitational acceleration of  $980.000\text{cm/s}^2$  plus noise. If encoding the dataset using this model only requires 453 bits, the model's inaccuracy is estimated as:

$$\frac{453}{1620} = 0.27$$

This tells us how much information must be added to the model to fully account for the empirical data. It quantifies the gap between representation and reality. ■

What this example illustrates is a profound ambiguity: when our models and our data disagree, we cannot determine, in general, whether the failure lies in the experiment (misCoding) or in the model (inaccuracy). This ambiguity is not a flaw of the framework; it is a reflection of the inherent uncertainty we face when attempting to describe the world. It is precisely this uncertainty that the theory of nescience seeks to explore, quantify, and ultimately reduce.

## 1.7 Surfeit

In our pursuit of understanding, we often encounter a paradox: the more complex and elaborate our explanations, the less confident we should be in the depth of our knowledge. Complexity, when unnecessary, signals confusion. When we struggle to explain a concept concisely, it is likely because our grasp of it remains partial. This observation reveals a deeper motivation behind scientific inquiry: the drive to eliminate what is superfluous, to strip our models down to their essential structure. By doing so, we gain clarity not only in explanation but also in prediction and control of phenomena.

Science depends on descriptions, most often in the form of mathematical models, to interpret the past, predict the future, analyze cause and effect, and engineer solutions to practical problems. However, these models, to serve us effectively, must remain within the limits of our cognitive capacities. As scientists and engineers, we are compelled to seek out models that are not only correct but also minimal. This minimality is not an aesthetic preference; it is a cognitive and computational necessity. Even if, in the future, machines take over the work of scientific reasoning and our concern for human comprehensibility fades, the idea of minimizing unnecessary complexity

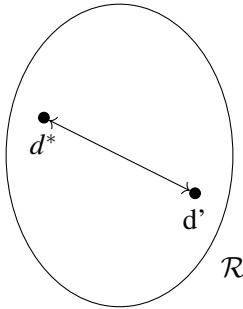


Figure 1.6: Surfeit of a model.

will remain a conceptual cornerstone, albeit perhaps relegated to a more theoretical realm.

The theoretical limit of our knowledge about a representation, denoted as its perfect description  $d^*$ , is given by the shortest possible computer program that can reconstruct that representation. In practice, we rarely achieve this ideal. The excess or surfeit of a description  $d'$  can be calculated by comparing the length of the given description  $d'$  to that of the shortest possible description  $d^*$  for that representation (refer to Figure 1.6). In other words, *surfeit* measures how much longer a description is compared to the most concise description we could ideally achieve. Regrettably, due to our incomplete knowledge, we generally do not know the shortest description of a representation, therefore surfeit is a value that must be estimated in practice.

This definition leads us to a profound insight: perfect knowledge implies randomness. If a description were perfect, it would be incompressible; any pattern or regularity would suggest redundancy that could be eliminated. According to Kolmogorov complexity, incompressible strings are indistinguishable from random sequences (see Section E.6). Thus, within the framework of our theory, a perfect description of a phenomenon must be random.

This conclusion might seem counterintuitive. Traditionally, randomness has been associated with disorder and the absence of meaning. But in our context, a random description is not meaningless, it is simply maximally informative, packing as much content as possible into the smallest possible space. Still, not all random descriptions are perfect. A theory might become increasingly compressed and eventually become random, only for a new representation to emerge, perhaps based on a different encoding, offering a shorter and more insightful description.

Understanding randomness in this way reshapes the boundaries of what

we consider knowable. Rather than seeing randomness as a wall, we see it as a marker of how far we have progressed. It signals the frontier of comprehension, a point where additional refinement is no longer possible unless we reconceptualize the problem. And crucially, this perspective does not just limit us to analyzing what we already know; it provides a strategy for identifying gaps in knowledge and pointing toward new avenues of discovery.

With our definition of surfeit, in which lengthier explanations are deemed inferior, we are not implying that textbooks should always strive for utmost brevity. Contrarily, in certain situations, we anticipate textbooks to be highly redundant. A concise book contains a large amount of information in a highly condensed form, making it challenging for humans to assimilate that information. However, a redundant textbook (such as this one) presents the same amount of information but in a larger space, hence its content is easier to comprehend. Moreover, in fields outside science, redundancy may be desirable. For instance, in law, redundancy aids lawyers in memorizing legal texts, and in music, repetition can contribute to harmony, as exemplified in a canon.

## 1.8 Nescience

*Nescience* is an old-fashioned English word that means "lack of knowledge or awareness." At first glance, it appears to be synonymous with the word "ignorance." However, there is a subtle but important distinction. Ignorance refers to the absence of knowledge when such knowledge exists and could be acquired (for instance, by reading a book), whereas nescience refers to the absence of knowledge when that knowledge does not yet exist, when it is unknown to everyone. The theory of nescience has been developed to quantitatively measure how much we do not know in situations where knowledge is not yet available, aiming to quantify the extent of our collective lack of knowledge.

Intuitively, the extent of what we do not know about a research entity can be assessed through the metrics of miscoding, inaccuracy, and surfeit associated with a given representation and its description. These three metrics capture the main types of errors we might make. Miscoding reflects how well the representation encodes the original entity under investigation; inaccuracy indicates how effectively the description models that representation; and surfeit measures the quality of the description itself in terms of unnecessary descriptive complexity. The best combinations of representations and descriptions are those that exhibit low miscoding, low inaccuracy, and low surfeit.

However, these factors are often in tension with one another. Improving

one dimension can inadvertently worsen another, making the process of refining knowledge inherently complex. For example, increasing the complexity of a description might improve its accuracy, bringing the output closer to the intended representation; but this gain may come at the cost of clarity and simplicity, thereby increasing surfeit. Similarly, modifying a representation to better match the original entity, thus reducing miscoding, can lead to an increase in the inaccuracy of the description; the description may no longer align well with the new representation. In scientific research, such trade-offs are frequently encountered. More accurate models often require additional parameters, sophisticated computations, or richer structures—factors that can obscure understanding, hinder reproducibility, or limit practical applicability. Recognizing these tensions helps researchers make more informed decisions about model selection: balancing the competing goals of simplicity, fidelity, and succinctness. The theory of nescience makes these trade-offs explicit, providing a structured way to analyze and optimize them.

A pair  $(r, d)$ , composed of a description and a representation, is said to be Pareto optimal if there is no other pair  $(r', d')$  that improves at least one of the three components of nescience—miscoding, inaccuracy, or surfeit—without worsening another. In this sense, a Pareto optimal pair represents a balance point where any improvement in one dimension would result in a degradation in another. This notion helps us identify a set of candidate  $(r, d)$  pairs that offer the best possible trade-offs and jointly provide strong explanatory value for an entity  $e$ .

However, in scientific practice, we typically aim to select a single description to serve as the model of a research entity. To make this selection, we must define a utility function that allow us to classify and rank the candidate descriptions. The exact form of this utility function depends on the field in which the theory of nescience is being applied. For instance, in machine learning, where entities are represented as datasets, a reasonable utility function might be the average of the three components of nescience: miscoding, inaccuracy, and surfeit. This provides a simple and effective way to evaluate and choose the most suitable description.

In traditional scientific approaches, it is common to fix a particular representation of the entity under investigation and then focus on selecting or developing a model that minimizes inaccuracy and possibly surfeit. While this methodology is often effective within well-established domains, it risks overlooking better alternatives that arise from reconsidering the representation itself. The theory of nescience emphasizes that miscoding, inaccuracy, and surfeit must be minimized jointly. This holistic approach avoids the danger of settling into a local minimum—where a model seems optimal given a fixed representation but a better explanation might exist elsewhere

in the space of possible (representation, description) pairs. By simultaneously considering and optimizing both the representation and description, the theory enables a more comprehensive and flexible exploration of scientific models.

## 1.9 Perfect Knowledge

In this book, we assume that the final objective of science is to achieve perfect knowledge, understood as the state in which our understanding of an entity is both accurate and efficient. There are two fundamental strategies for decreasing nescience. One is to develop new descriptions; these could stem from the formulation of novel theories, improvements upon existing explanations, or the elimination of unnecessary assumptions. The second is to discover better representations—ways of encoding or framing the entity under study that more faithfully reflect its essential properties. Scientific progress, from this perspective, is inherently iterative: over time, new candidate pairs of descriptions and representations are proposed, each new proposal intended to bring us closer to this ideal state by reducing nescience.

When the nescience of a pair composed of a representation and a description is equal to zero, we say that *perfect knowledge* about an entity has been achieved. This state corresponds to the simultaneous nullification of the three components of nescience (misencoding, inaccuracy, and surfeit), each being exactly zero in the ideal, limiting case. Misencoding being zero means the representation perfectly encodes the intended entity; inaccuracy being zero means the description fully and faithfully reconstructs the representation; and surfeit being zero means the description contains no redundancy, it is as concise as possible. In this ideal scenario, our understanding of the entity is complete, exact, and optimally efficient. In such a situation, perfect knowledge requires that descriptions be random in the algorithmic sense—incompressible and patternless—embodying the optimal efficiency of understanding. These random descriptions mark the ultimate boundary of knowledge, beyond which any modification would only increase nescience.

Nevertheless, reality is more complicated. Our current measurements of misencoding, inaccuracy, and surfeit are merely estimations, approximations that are themselves subject to error and limited by our tools and understanding. Consequently, it is not always evident whether a particular refinement is truly superior. From a practical standpoint, we therefore relax the requirement of constant decline and instead accept the weaker condition that nescience should decrease on average over time. Temporary setbacks or local increases are tolerable, as long as the general trajectory is toward a deeper and more refined understanding of the entity under study.

It is important to recognize that there may not necessarily be a single ultimate theory or uniquely optimal pair. Multiple combinations of representations and descriptions may theoretically yield a nescience of zero. The most suitable combination in any given context often depends on the intended practical application. Different applications may favor different aspects of a representation or description, such as interpretability, computational efficiency, or generalizability, leading to different but equally valid realizations of perfect knowledge.

However, perfect knowledge is not always achievable. On the Pareto frontier, where each point represents a pair  $(r, d)$  of representation and description that achieve the best possible balance between the components of nescience, reducing one component inevitably increases another. In such cases, even if our understanding is not complete, we face the paradox that further research can only increase our ignorance, thereby dooming all efforts in this direction to failure. The only possible solution is to reframe the problem from a different perspective, adopting an alternative pair of representations and descriptions  $(r', d')$  that allow progress in reducing overall nescience.

We can use this property—the reduction of nescience—as a criterion to distinguish between valid scientific disciplines and those that fall outside the scope of science, a challenge known as the *demarcation problem* in the philosophy of science. In scientific fields, successive refinements in descriptions and representations tend to yield a measurable decrease in nescience, signifying genuine progress in understanding. In contrast, non-scientific theories, including pseudosciences, typically fail to exhibit this pattern. Despite the introduction of new descriptions or representations, there is no meaningful reduction in nescience over time. This implies that such disciplines fail to produce cumulative knowledge or deeper insights. In pseudoscientific domains, further research often leads to reinterpretations, embellishments, or rhetorical shifts, rather than the kind of substantive progress that characterizes scientific inquiry. As a result, these areas remain stagnant, unable to generate new insights or approach perfect knowledge.

## 1.10 Unknown Unknowns

We have previously discussed the existence of an unknown unknowns area, comprising problems for which we not only lack solutions and whose very existence escapes our awareness. Within the framework of the theory of nescience, our goal is to develop a systematic procedure for identifying and exploring potential research entities hidden in this region. One approach could involve randomly generating binary strings and querying an oracle to assess whether any of them closely approximates the representation of

a (hopefully unknown) entity. This embodies the idea of discovering new knowledge by pure chance. However, the sheer magnitude of the space of possible strings makes this brute-force strategy computationally impractical. Consequently, we must seek more efficient and guided methods for navigating this uncharted domain.

To discover what lies hidden in the unknown unknowns, we must first delineate the region encompassing everything we already know. This known region comprises two types of topics. The first are the known knowns—these are topics that are well understood, where our descriptions are accurate, concise, and reliable. The second group includes the known unknowns, which are problems we are aware of but for which we still lack complete or satisfactory explanations. The boundary separating this region of known topics from what lies beyond, a boundary we refer to as the knowledge frontier. It represents the outer limit of our current understanding, a conceptual demarcation where the known ends and the unexplored begins. Any entity that exists beyond this frontier—one that has not yet been identified or studied, constitutes a new research entity residing in the domain of the unknown unknowns.

Yet, identifying the exact list of already known topics in practice is far from straightforward. The main challenge lies in determining which topics have already been studied and formally documented through published research. Scientific knowledge is dispersed across countless articles, journals, and disciplines, and there is no single, unified repository that captures the full extent of human understanding. Moreover, variations in terminology and differences in how topics are categorized further complicate the task. Thus, establishing the scope of the known demands careful analysis of bibliographic data, ontological classifications, and, often, expert consensus.

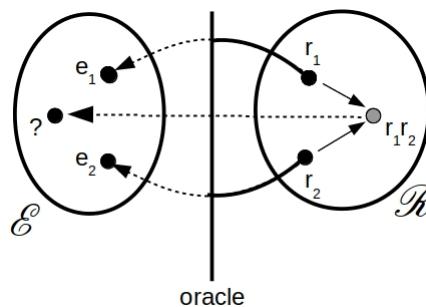


Figure 1.7: Discovering new research entities.

In this book, we explore an alternative strategy for identifying new research directions by combining already known concepts. The basic idea is

straightforward: by taking two distinct representations, each corresponding to a different known entity, we can identify a new entity by joining them into a single representation. To make this approach systematic, we assume that the space of valid representations is closed under such combinations—in other words, that combining any two representations will always yield another one. This assumption enables us to construct joint topics mechanically and to explore their potential to uncover novel insights or previously unexamined questions. In practical terms, this involves computing all possible combinations of known entities and selecting those with the greatest potential to yield new and interesting research directions. However, the precise meaning and significance of any new entity formed in this way remain to be determined through further investigation and reflection.

■ **Example 1.6** We could combine prominent topics from the field of theoretical computer science with those from phenomenology to identify promising new research directions. By merging the concepts of "minimum complexity computer programs" and "self-awareness," we arrive at a potential new research topic: "minimum complexity self-aware computers." This would involve investigating the minimum complexity required for a computer program to exhibit self-awareness. ■

As wisely stated by Saint John of the Cross (1542-1591), "to go where we do not know, we must go by a path we do not know." This insight holds true in scientific discovery: the likelihood that a combination of two already known entities will lead to a new entity located in the unknown unknowns is greater when the entities being combined are themselves poorly understood. In contrast, combining well-understood entities is more likely to produce a result that remains within the bounds of current knowledge—that is, inside the knowledge frontier (see Figure 1.8). This is because the areas surrounding a well-known entity are often already thoroughly explored, leaving little room for the kind of novelty sought—ideas that lie beyond the current boundaries of knowledge and have the potential to open up entirely new lines of inquiry.

Another approach to increasing the chances of reaching the unknown unknowns is to combining topics from two distinct fields of knowledge. The likelihood that such a combination has already been explored is relatively low, primarily because it would require someone with substantial expertise in both areas—a rarity in today's academic landscape, where scientists are increasingly specialized in narrow domains. Interdisciplinary combinations, therefore, offer fertile ground for novel discoveries, as they may produce connections that have never been examined or even imagined within the confines of a single discipline.

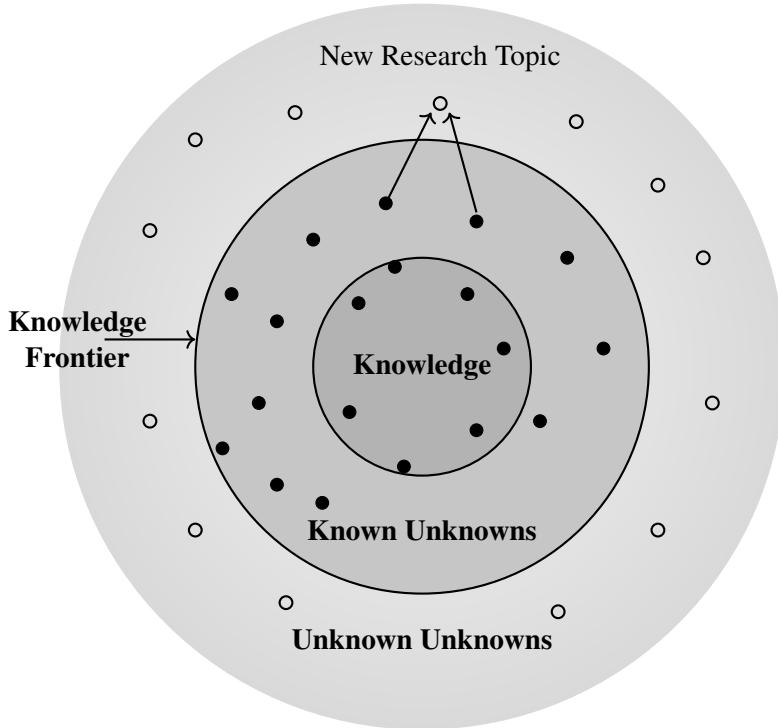


Figure 1.8: The structure of knowledge.

## References

The theory of nescience builds upon several well-established foundations in information theory, computability, algorithmic complexity, and the philosophy of science. The following references provide the theoretical background and conceptual tools necessary to understand and situate the ideas introduced in this chapter.

Sipser's book [Sip12] is a widely respected introduction to formal languages, automata, and computability theory. It lays the groundwork for understanding which descriptions are computationally feasible, an essential aspect of the theory of nescience, as it assumes that knowledge must be computable to be meaningful.

Li and Vitányi's book [LV13] is a comprehensive volume that presents the theory of Kolmogorov complexity, which formalizes the idea of description length using the shortest program capable of generating a given object. The concepts developed by Li and Vitányi are central to the theory of nescience, especially in defining metrics such as inaccuracy, miscoding, and surfeit based on algorithmic information theory.

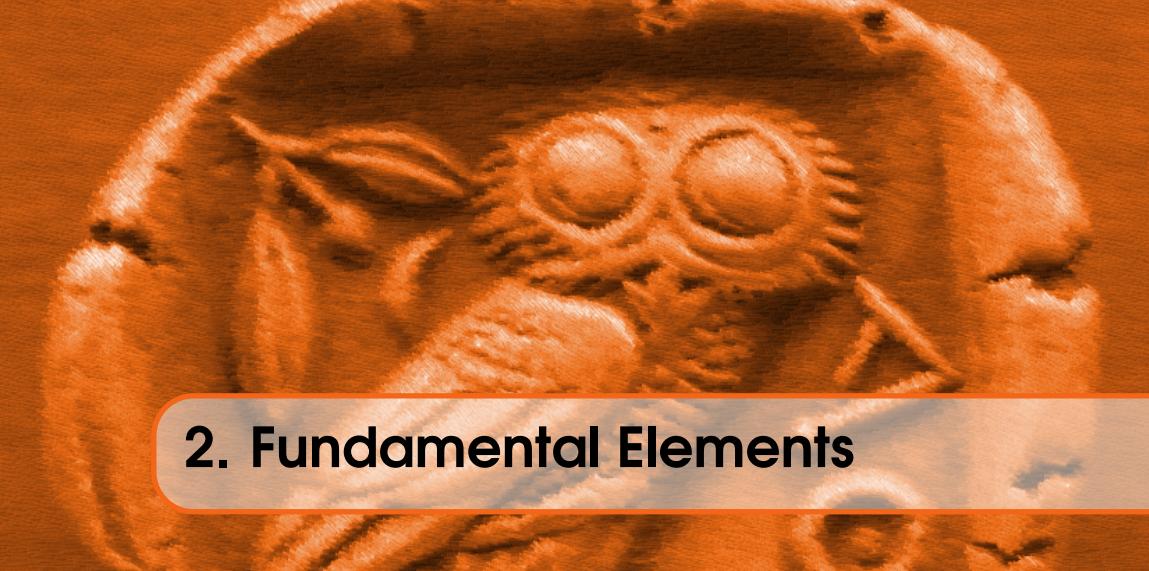
Cover and Thomas's book [CT12] is a foundational text in information theory, providing the mathematical framework for understanding concepts such as entropy, mutual information, and data compression. It serves as a rigorous yet accessible introduction to how information can be quantified, transmitted, and encoded, ideas that relate to the notions of description length in the theory of nescience.

Chalmers's book [Cha13] offers a critical and historical introduction to the philosophy of science. It examines the assumptions, limits, and methodologies of scientific practice. This perspective is invaluable for situating the theory of nescience within the broader discourse about how scientific knowledge is constructed, evaluated, and refined over time.

# Part 1: Foundations

<b>2</b>	<b>Fundamental Elements</b>	<b>51</b>
<b>3</b>	<b>Miscoding</b>	<b>91</b>
<b>4</b>	<b>Inaccuracy</b>	<b>105</b>
<b>5</b>	<b>Surfeit</b>	<b>119</b>
<b>6</b>	<b>Nescience</b>	<b>131</b>
<b>7</b>	<b>Interesting Questions</b>	<b>155</b>





## 2. Fundamental Elements

*We are all agreed that your theory is crazy.  
The question which divides us is whether it is crazy enough.*  
Niels Bohr

The first step in quantifying our lack of knowledge involves the precise identification of the entities under investigation. The elements of this collection depend largely on the specific application of the theory of nescience. Each application requires a distinct set of entities, whether mathematical objects, living organisms, human needs, or otherwise. Fortunately, the process of quantifying what we do not know remains consistent across all such domains.

The next step is to devise a procedure for representing the identified entities as strings of symbols. Accurately encoding a research entity is a complex and unresolved epistemological challenge. The solution proposed within the theory of nescience is based on the concept of an oracle Turing machine. The practical feasibility of this solution depends largely on the level of abstraction of the entities being studied. For example, encoding abstract mathematical objects poses significant difficulties and often requires an approximation. In contrast, encoding computer programs is relatively straightforward, as they are already expressed as strings.

Once a suitable method for encoding the original entities as string-based representations has been established, the final step is to produce a description. This description should be both accurate and concise, reflecting our current understanding of the representation. In the theory of nescience, descriptions are required to be computable, meaning that a computer must be able to fully reconstruct the original representation from its description. However, descriptions refer to representations (that is, to the way entities are encoded) not to the entities themselves. As a result, how well a description captures knowledge about an entity depends largely on the quality of the representation used.

In this chapter, we will formalize these core concepts: entities, representations, and descriptions, among others. We will also explore what constitutes a perfect description, how to compute the combined representation of multiple entities, and how to describe a representation given some prior background knowledge. Additionally, we will examine the concept of a research area and its associated properties.

## 2.1 Entities

Defining the nature of a research entity is a complex and unresolved philosophical problem. We approach this complexity from a fundamentally pragmatic perspective, rather than a philosophical one. Our theory starts from the premise that there exists a non-empty set of *entities* that we seek to understand.

**Notation 2.1.** *We represent the set of research entities under consideration as  $\mathcal{E}$ .*

The elements of  $\mathcal{E}$  are defined relative to a particular area of knowledge in which the theory of nescience is applied, such as mathematics, physics, engineering, or the social sciences, or may be further restricted to a specific subarea within any of these disciplines. From a formal perspective,  $\mathcal{E}$  should be regarded as a fixed but generally inaccessible collection: although its elements are determined by the underlying domain of inquiry, there is in general no effective (computable) procedure for accessing, deciding membership in, or enumerating its elements.

The set  $\mathcal{E}$  contains both *knowable* and *unknowable* entities (see Section G.2 for further discussion of the notion of an entity and Section 2.3 for a formal characterizations of each of these types of entities). Knowable entities are those for which a correct or perfect representation may exist, and they may be either concrete, such as physical particles or biological species, or abstract, such as mathematical structures or algorithms. The set  $\mathcal{E}$

also includes unknowable entities, whether real, such as infinite systems or idealized structures, or fictitious, such as hypothetical physical entities later shown not to exist.

Unknowable fictitious entities are included in  $\mathcal{E}$  because they play a genuine role in scientific research: even when they do not correspond to any real entity, they guide experimentation, shape theories, and contribute to the dynamics of discovery, error, and revision. An entity belongs to  $\mathcal{E}$  insofar as it is the intended target of representation, measurement, explanation, or modeling, independently of whether it ultimately admits a correct or perfect representation.

In the theory of nescience, the possibility of using universal sets is excluded; that is, the existence of a set  $\xi$  containing all possible objects cannot be assumed. The problem with universal sets is that their existence is incompatible with Cantor's theorem (see Example 2.1). Cantor's theorem proves that the power set  $\mathcal{P}(\xi)$ , consisting of all possible subsets of  $\xi$ , has more elements than the original set  $\xi$ . This contradicts the assumption that  $\xi$  includes everything. The set  $\mathcal{E}$  must be explicitly fixed and domain-scoped; that is, it must denote the set of entities under consideration in a given research area, rather than an ambiguous or open-ended collection.

■ **Example 2.1 — Cantor's theorem.** Cantor's theorem proves that for any set  $A$ ,  $d(A) < d(\mathcal{P}(A))$ <sup>1</sup>. Consider  $f : A \rightarrow \mathcal{P}(A)$ , a function that maps each element  $x \in A$  to the set  $\{x\} \in \mathcal{P}(A)$ ; evidently,  $f$  is injective, implying  $d(A) \leq d(\mathcal{P}(A))$ . To substantiate that the inequality is strict, let's assume there exists a surjective function  $g : A \rightarrow \mathcal{P}(A)$  and consider the set  $B = \{x \in A : x \notin g(x)\}$ . As  $g$  is surjective, there must exist a  $y \in A$  such that  $g(y) = B$ . This, however, raises a contradiction,  $y \in B \Leftrightarrow y \notin g(y) = B$ . Consequently, the function  $g$  cannot exist, therefore  $d(A) < d(\mathcal{P}(A))$ . ■

In the theory of nescience, not all conceivable sets are acceptable, as some may give rise to paradoxes. Take, for example, Russell's paradox, which proposes a set  $R$  consisting of all sets that are not members of themselves. The paradox arises when we try to discern whether  $R$  is a member of itself (see the Example 2.2). To avoid such problems, the theory of nescience is based on the Zermelo-Fraenkel set of axioms, along with the Axiom of Choice (or ZFC). The Axiom of Separation (if  $P$  is a property with parameter  $p$ , then for any set  $x$  and parameter  $p$  there exists a set  $y = \{u \in x : P(u)\}$  that includes all those sets  $u \in x$  that have property  $P$ ) allows the use of this notation only to construct sets that are subsets of already existing sets. A more extensive Axiom of Comprehension (if  $P$  is a property, then there exists a set  $y = \{u : P(u)\}$ ) would be required to allow sets like the one

---

<sup>1</sup>Here  $d(\cdot)$  denotes cardinality, see Appendix A.

proposed by Russell's paradox. Russell's paradox arises from the use of an unrestricted comprehension principle. In the axioms of ZFC, and in the theory of nescience, the axiom of comprehension is considered false.

■ **Example 2.2 — Russell's paradox.** Suppose  $R$  is the set of all sets not members of themselves, such that  $R = \{x : x \notin x\}$ . The contradiction arises when querying if  $R$  is a member of itself. If  $R$  is not a member of itself, by its own definition, it should be; conversely, if  $R$  is declared to be a member of itself, its definition dictates it should not be. Symbolically, this can be written as  $R \in R \Leftrightarrow R \notin R$ . ■

In this book we do not address the classic problems of ontology, that is, the classification of entities that exist in the world and can be known. Furthermore, we do not attempt to resolve epistemological questions, such as how scientific knowledge is validated by evidence, or what the nature of that evidence is.

Once a set  $\mathcal{E}$  of entities has been selected, the next step is to uniquely encode them as strings of symbols, which will make them easier to describe. A method for performing this encoding efficiently is described in the following section.

## 2.2 Representation Oracle

The representation of the entities composing the set  $\mathcal{E}$ , which may be abstract or only informally specified, raises significant challenges (see Section G.4 for an overview of proposed solutions in the scientific literature, and their limitations). This book presents a particular approach to this problem by dividing scientific representation into two closely related subproblems: the encoding of entities by means of representations, and the modeling of these representations by means of descriptions. From this perspective, scientific models do not explain entities directly, but rather explain them through their representations (see Figure 1.1 in the Introduction reproduced here for ease of reference). In this section we focus on encoding; the role of descriptions is addressed in Section 2.5.

In Kolmogorov complexity, the representation problem is usually approached by assuming that the set  $\mathcal{E}$  is well defined and countable, and that a total encoding function  $f : \mathcal{E} \rightarrow \mathbb{N}$  exists, analogous to a Gödel numbering. The theory of nescience similarly adopts the idea that entities can be encoded as numbers, or equivalently as finite strings. Our approach, however, departs from this tradition by dropping the requirement that  $\mathcal{E}$  be explicitly definable or enumerable.

Rather than postulating an encoding function with domain  $\mathcal{E}$ , we take

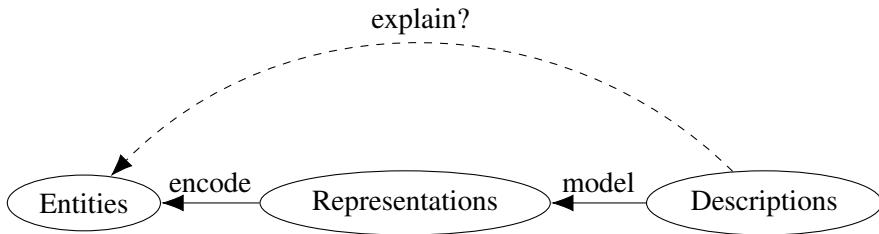


Figure 2.1: The Problem of understanding

as primitive the well-defined set of finite strings and assume the existence of an abstract oracle that internally associates certain strings with entities. More precisely, we introduce an oracle  $\mathcal{O}_{\mathcal{E}}$  that acts as a scientific authority. Given a finite string  $r$ , the oracle determines whether  $r$  contains sufficient information to reconstruct a unique entity without additional input. If so, the oracle internally associates  $r$  with that entity; otherwise, the string is deemed non-representational.

The oracle  $\mathcal{O}_{\mathcal{E}}$  is an idealized construct. For most collections of entities, it cannot be realized in practice, and no explicit decoding procedure associated with it can be provided. The role of  $\mathcal{O}_{\mathcal{E}}$  is purely formal: it serves as a reference device for delimiting what representations of entities are in principle obtainable within the framework. Accordingly,  $\mathcal{O}_{\mathcal{E}}$  is treated as a black box and is defined only through its input-output behavior. In this respect,  $\mathcal{O}_{\mathcal{E}}$  is analogous to the oracle in an oracle Turing machine (see Definition C.5.1).

For the purposes of this book, and without loss of generality, we restrict attention to binary strings as the medium of representation.

**Definition 2.2.1** Given a set of entities  $\mathcal{E}$ , a *representation oracle* is a total decision oracle

$$\mathcal{O}_{\mathcal{E}} : \mathcal{B}^* \rightarrow \{\text{true}, \text{false}\}$$

with the following semantics: there exists an oracle-internal partial *representation function*

$$\delta_{\mathcal{E}} : \mathcal{B}^* \rightharpoonup \mathcal{E}$$

such that, for every  $r \in \mathcal{B}^*$ ,

$$\mathcal{O}_{\mathcal{E}}(r) = \text{true} \iff \delta_{\mathcal{E}}(r) \text{ is defined.}$$

When defined,  $\delta_{\mathcal{E}}(r)$  yields a unique entity of  $\mathcal{E}$  reconstructed from  $r$  without any additional information. The function  $\delta_{\mathcal{E}}$  is not assumed to be computable or explicitly characterizable, and serves only to fix the semantic interpretation of the oracle.

**TODO: Introduce the concept of "decodable" representation as those strings for which the Oracle answer true.**

Within this framework, only elements of  $\mathcal{B}^*$ , that is, finite binary strings, are eligible to serve as representations of entities (see the discussion of the ontology problem in Section G.4). Other forms of representation (such as physical models, diagrams, or verbal descriptions) are admissible only insofar as they can be transcribed into binary strings. No distinction is drawn between scientific representations and other types of representations (see the discussion of representational demarcation in Section G.4). Finally, it is essential to note that  $\delta_{\mathcal{E}}$  is a partial function: not every binary string represents an entity.

In general, the internal oracle need not be injective, surjective, or unique for a given collection of entities, reflecting the plurality and ambiguity inherent in real-world scientific representation.

■ **Example 2.3** For a given set of entities  $\mathcal{E}$ , the existence of a representation internal function  $\delta_{\mathcal{E}}$  does not imply its uniqueness. For instance, a binary negation oracle (which transforms the zeros of a binary string into ones, and the ones into zeros), assigning to each string  $x \in \mathcal{B}^*$  the entity  $\delta_{\mathcal{E}}(\neg x)$  would also qualify as a representation function. ■

Rather than deploying individual oracles for each set  $\mathcal{E}$ , we could have employed a universal oracle machine. This is a machine  $\mathcal{U}_{\mathcal{O}}$  that, given the encoding of an oracle  $\mathcal{O}_{\mathcal{E}}$  and a string  $r$  as input, computes  $\mathcal{U}_{\mathcal{O}}(\langle \mathcal{O}_{\mathcal{E}}, r \rangle) = \mathcal{O}_{\mathcal{E}}(r)$ . Universal machines are particularly applicable to the universal set  $\xi$ , encompassing all entities. However, the theory of nescience is explicitly domain-relative: the choice of  $\mathcal{E}$  determines which entities are under investigation and fixes the admissible representations in which the notion of nescience is defined. We work with entity sets  $\mathcal{E}$  corresponding to different areas of knowledge, each associated with a single oracle  $\mathcal{O}_{\mathcal{E}}$ , the one most suitable to our current knowledge and practical needs.

■ **Example 2.4** Consider the case when the subjects of study are animals. Initially, one might use detailed physical descriptions of animals as encodings. In this scenario, the internal representation function would be a hypothetical machine capable of reconstructing the original animal from its description. As our understanding of biology advances, we might instead adopt an alternative encoding based on the animals' DNA. Both of these encodings serve as valid representations of the entities. ■

As illustrated in Example 2.3 and Example 2.4, the entities in  $\mathcal{E}$  can be encoded in multiple ways (see the problem of style in Section G.4). Different oracles accommodate different encoding schemes. In practical terms, some strings provide more adequate representations of an entity than others (see the standard of accuracy in Section G.4). The optimal representation depends on the type of questions we aim to answer. Representations must not only correspond to entities, but also they must be relevant to them (see the requirement of directionality in Section G.4 and Chapter 3).

We employ an oracle function rather than an oracle relation  $\mathcal{R}_{\mathcal{O}} \subset \mathcal{B}^* \times \mathcal{E}$ , not merely to associate strings with their corresponding entities, but to reconstruct an entity from its representation. The internal oracle's ability to recover original entities underpins our capacity to make hypotheses about entities based on their representations (see surrogate reasoning in Section G.4). According to the theory of nescience, scientific inquiry involves not only learning how to encode entities properly but also understanding the mechanisms by which internal representation functions decode them.

The purpose of encoding entities in the theory of nescience differs fundamentally from that in Shannon's information theory (see Appendix D), as illustrated in Example 2.5.

■ **Example 2.5** Take, for instance, a set  $\mathcal{E}$  consisting of two books: "The Ingenious Gentleman Don Quixote of La Mancha" and "The Tragedy of Romeo and Juliet". We might encode the first book with the string "0" and the second with the string "1". While these strings allow us to uniquely identify each book within the set, they do not qualify as valid encodings within the theory of nescience. In information theory, the goal is to uniquely identify an object based on a reference, assuming mutual agreement between the sender and the receiver about the mapping from references to objects. In contrast, the theory of nescience seeks representations that preserve the richness and detail of the original entities. For example, it would be impossible to hypothesize about Cervantes' influence on Shakespeare using only the strings "0" and "1". ■

One possible response to the limitation discussed in Example 2.5, where entities are encoded using overly simplistic strings that fail to capture their internal structure, would be to require the set  $\mathcal{E}$  to be infinite, as is done in Kolmogorov complexity. However, while requiring  $\mathcal{E}$  to be infinite prevents all entities from being encoded using uniformly short strings, it does not prevent the use of representations that, although longer, remain arbitrary and devoid of semantic content. That is, they do not allow us to make meaningful inferences about the original entities based solely on their representations. To enable such reasoning, the encoding must preserve essential structural

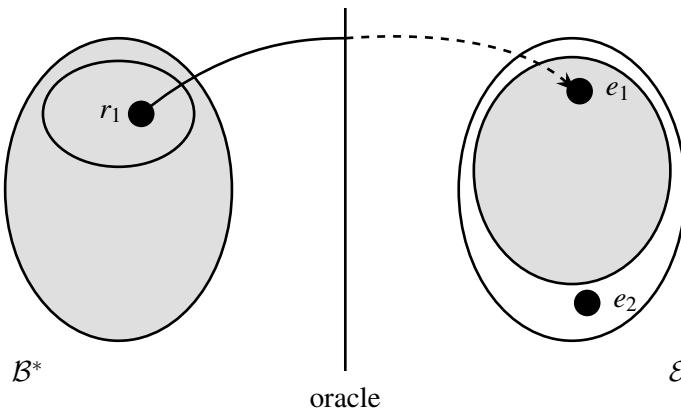


Figure 2.2: Encodings and Entities

and semantic features of the entities, not just their identity.



One of the fundamental challenges in science, and in human intellectual activity more broadly, is the tendency to confuse symbols with the things they represent. The theory of nescience has been carefully designed to avoid this issue by clearly distinguishing between research entities and their representations. However, making this distinction explicit at every step would render the book unnecessarily difficult to read. We have aimed to strike a balance between clarity of exposition and rigor of definition. Occasionally, especially when introducing new ideas, we use the term *topic* to refer broadly to an entity, a representation, or both. Nevertheless, in all mathematical definitions and propositions, this distinction is made unambiguously. In the case of any uncertainty, the formal definitions should be taken as the definitive reference.

### Redundant Representations

Within the theory of nescience, representations are not required to be minimal. Their primary role is to be correct, in the sense that they contain sufficient information for the oracle to reconstruct the corresponding entities. Representations themselves are not intended to support scientific reasoning directly, and therefore need not be concise. The requirement of minimality becomes relevant only at a later stage, when we introduce descriptions, or models (see Section 2.5). Descriptions aim to capture essential aspects of representations in a more compact form and are explicitly designed to support reasoning, explanation, and prediction. In that context, assumptions about background knowledge and shared conventions are not only acceptable but often necessary to achieve brevity and explanatory power.

**Definition 2.2.2** Let  $\mathcal{O}$  be an oracle with representation function  $\delta$ , and let  $r \in \mathcal{B}^*$  a decodable representation. We say that  $r$  is a *non-redundant* representation if there is no decodable representation  $s \in \mathcal{B}^*$  such that

$$l(s) < l(r) \wedge \delta(s) = \delta(r)$$

**Notation 2.2.** We denote by  $\bar{\mathcal{R}}_{\mathcal{O}_E}$  the set of all non-redundant decodable representations.

**Proposition 2.2.1 — Existence of non-redundant descriptions.** Fix an oracle  $\mathcal{O}$  and an entity  $e \in \delta(\mathcal{R}_{\mathcal{O}}^*)$ . Then there exists at least one non-redundant description  $r \in \mathcal{R}_{\mathcal{O}}^{*,0}$  such that  $\delta(r) = e$ .

*Proof.* Let  $A_e := \{ |r| : r \in \mathcal{R}_{\mathcal{O}}^*, \delta(r) = e \} \subseteq \mathbb{N}$ . By assumption  $e \in \delta(\mathcal{R}_{\mathcal{O}}^*)$ , so  $A_e \neq \emptyset$ . Since  $\mathbb{N}$  is well-ordered,  $A_e$  has a minimum  $m$ . Choose  $r \in \mathcal{R}_{\mathcal{O}}^*$  with  $\delta(r) = e$  and  $|r| = m$ . By construction, there is no shorter  $s \in \mathcal{R}_{\mathcal{O}}^*$  with  $\delta(s) = e$ , hence  $r \in \mathcal{R}_{\mathcal{O}}^{*,0}$ . ■

■ **Example 2.6 — Redundant expansions and non-redundancy.** Let  $\mathcal{B} = \{0, 1\}$  and let  $\sigma$  be a style in which appending any number of trailing zeros is redundant:

$$\text{Red}_{\sigma}(r) = \{ r0^k : k \in \mathbb{N} \}.$$

Consider an oracle  $\mathcal{O}$  for which  $\delta(101) = e$  for some entity  $e$ , and assume redundancy closure. Then  $1010, 10100, 101000, \dots$  are all in  $\mathcal{R}_{\mathcal{O}}^*$  and decode to the same entity  $e$ . Among these,  $101$  is non-redundant (there is no shorter representation of  $e$  in this family), whereas  $10100$  is redundant relative to  $\mathcal{O}$  because it is longer than  $101$  and carries no essential additional information about  $e$ . ■

**Proposition 2.2.2 — Redundancy does not affect the core.** Assume redundancy closure (Assumption 2.2.3). For any oracle  $\mathcal{O}$ , the set  $\mathcal{R}_{\mathcal{O}}^{*,0}$  contains, for each entity  $e \in \delta(\mathcal{R}_{\mathcal{O}}^*)$ , at least one shortest representation of  $e$  accepted by  $\mathcal{O}$ . In particular, adding redundant expansions to  $\mathcal{R}_{\mathcal{O}}^*$  does not create new elements of  $\mathcal{R}_{\mathcal{O}}^{*,0}$ .

*Proof.* The first claim is Proposition 2.2.1. For the second, redundant expansions (by definition) are longer strings constructed from an already accepted representation. They therefore cannot be shorter than an existing shortest representation of the same entity, hence they cannot enter  $\mathcal{R}_{\mathcal{O}}^{*,0}$ . ■

**Definition 2.2.3 — Redundancy closure.** For every oracle  $\mathcal{O}$  under style  $\sigma$ , for every  $r \in \mathcal{R}_{\mathcal{O}}^*$ , and every  $r' \in \text{Red}_{\sigma}(r)$ , we have  $r' \in \mathcal{R}_{\mathcal{O}}^*$ .

**Definition 2.2.4** Let  $\mathcal{O}_A$  and  $\mathcal{O}_B$  be two oracles defined over the set of entities  $\mathcal{E}$ . We say are said that the oracles  $\mathcal{O}_A$  and  $\mathcal{O}_B$  are *comparable* if

$$\mathcal{O}_A^\top = \mathcal{O}_B^\top.$$

### Minimal Oracles

Ideally, representation oracles should be minimal in size, meaning that all information needed to reconstruct an entity is carried by the representation string rather than being hard-wired into the oracle. A non-minimal oracle can simplify the representation problem by offloading complexity to its internal representation function, but it also forces researchers to understand not only the representation itself but the oracle's internal reconstruction process, something that, given the abstract nature of oracles, is generally not accessible in practice.

When an oracle contains hard-wired information, it can reconstruct an entity from a representation that leaves part of that information implicit, representations that an oracle without that hard-wired knowledge would be unable to decode. This property provides a natural basis for ordering oracles: we compare the sets of distinct non-redundant representations they accept, since an oracle with more hard-wired information admits a larger set of non-redundant decodable representations.

**Definition 2.2.5** Let  $\mathcal{O}_A$  and  $\mathcal{O}_B$  two comparable oracles. We say that  $\mathcal{O}_A$  *reduces* to  $\mathcal{O}_B$ , denoted by  $\mathcal{O}_A \triangleright \mathcal{O}_B$ , if

$$\bar{\mathcal{R}}_{\mathcal{O}_B} \subsetneq \bar{\mathcal{R}}_{\mathcal{O}_A}.$$

$\mathcal{O}_B$  is a simplification of  $\mathcal{O}_A$ , in the sense that it decodes strictly fewer non-redundant representations while still decoding the same entity subset of entities of  $\mathcal{E}$ .

We now formalize the idea that one oracle may be a *simplification* of another, obtained by removing hard-wired information.

This is consistent with the intuition that hard-wired information increases decodability (and therefore tends to enlarge the set of non-redundant perfect representations by enabling representations with missing information).

■ **Example 2.7** Consider representations of chemical compounds. A non-minimal oracle may incorporate extensive background chemical knowledge (e.g., periodic table, standard valences, and common structures). Under such an oracle, a compound could be represented by a short string such as its common name (e.g., *water*, *ethanol*), from which the oracle reconstructs the

full molecular structure. In this case, most of the information required for reconstruction is implicit in the oracle rather than explicit in the representation. By contrast, a more minimal oracle requires the representation to encode the relevant structure directly, for instance by specifying atoms and bonds. The representation is then longer but makes explicit what is being encoded and reduces reliance on background assumptions embedded in the oracle. ■

■ **Example 2.8 — Hard-wiring and reduction.** Let  $\mathcal{E} = \{e_0, e_1\}$ ,  $\mathcal{B} = \{0, 1\}$ , and fix a style  $\sigma$  in which redundant expansions are obtained by appending trailing zeros, as in Example 2.6. Consider two comparable oracles for  $(\sigma, S)$  with  $S = \{e_0, e_1\}$ :

- Oracle  $\mathcal{O}_B$  decodes  $e_0$  from the representation 0 and decodes  $e_1$  from 1, and nothing shorter. Then  $\mathcal{R}_{\mathcal{O}_B}^{*,0} = \{0, 1\}$ .
- Oracle  $\mathcal{O}_A$  contains one extra hard-wired bit of information allowing it to decode  $e_0$  also from the empty string  $\varepsilon$  (an information-omitted representation). Then  $\mathcal{R}_{\mathcal{O}_A}^{*,0} = \{\varepsilon, 0, 1\}$ .

Here  $\mathcal{R}_{\mathcal{O}_B}^{*,0} \not\subseteq \mathcal{R}_{\mathcal{O}_A}^{*,0}$  and  $\mathcal{R}_{\mathcal{O}_A}^{*,0} \not\subseteq \mathcal{R}_{\mathcal{O}_B}^{*,0}$ , so these two oracles are incomparable under  $\triangleright$ . Now modify  $\mathcal{O}_A$  so that it decodes all representations that  $\mathcal{O}_B$  decodes and, in addition, decodes  $\varepsilon$  as  $e_0$ . Then  $\mathcal{R}_{\mathcal{O}_A}^{*,0} = \{\varepsilon, 0, 1\}$ , and therefore  $\mathcal{O}_A \triangleright \mathcal{O}_B$ . ■

Basic properties of reduction.

**Proposition 2.2.3** Let  $S \subset \mathcal{E}$  a subset of entities, and  $\mathfrak{O}_S$  the set of comparable oracles over  $S$ . The reduction relation  $\triangleright$  on is:

- 1 *irreflexive*:  $\neg(\mathcal{O} \triangleright \mathcal{O})$  for all  $\mathcal{O} \in \mathfrak{O}_S$ ,
- 2 *transitive*: if  $\mathcal{O}_A \triangleright \mathcal{O}_B$  and  $\mathcal{O}_B \triangleright \mathcal{O}_C$ , then  $\mathcal{O}_A \triangleright \mathcal{O}_C$ , for all  $\mathcal{O}_A, \mathcal{O}_B, \mathcal{O}_C \in \mathfrak{O}_S$
- 3 *not total*: there may exist  $\mathcal{O}_A, \mathcal{O}_B \in \mathfrak{O}_S$  such that neither  $\mathcal{O}_A \triangleright \mathcal{O}_B$  nor  $\mathcal{O}_B \triangleright \mathcal{O}_A$  holds.

*Proof.* Irreflexivity follows from the fact that no set is a strict subset of itself. Transitivity follows from transitivity of strict set inclusion. Non-totality is witnessed by any pair of incomparable sets under  $\subsetneq$ , as illustrated in Example 2.8. ■

An oracle is minimal if it cannot be reduced to any other comparable oracle.

**Definition 2.2.6** Let  $\mathfrak{O}$  be a set of comparable oracles. We say that  $\mathcal{O} \in \mathfrak{O}$  is *minimal* if

$$\nexists \mathcal{O}' \in \mathfrak{O} : \mathcal{O} \triangleright \mathcal{O}'.$$

Equivalently,  $\mathcal{O}$  is minimal if there is no  $\mathcal{O}' \in \mathfrak{O}_{\sigma,S}$  with

$$\mathcal{R}_{\mathcal{O}'}^{\star,0} \subsetneq \mathcal{R}_{\mathcal{O}}^{\star,0}.$$

**TODO:** Minimal oracles are those who have no wired information.

■ **Example 2.9 — A minimal oracle.** Let  $\mathcal{E} = \{e_0, e_1, \dots, e_7\}$  and let  $\mathcal{B} = \{0, 1\}$ . Fix a style  $\sigma$  in which redundant expansions may append trailing zeros (as before), and let  $S = \mathcal{E}$ . Consider an oracle  $\mathcal{O}_{\text{bin}}$  whose non-redundant descriptions of each entity are exactly the length-3 binary strings,

$$\mathcal{R}_{\mathcal{O}_{\text{bin}}}^{\star,0} = \{000, 001, 010, 011, 100, 101, 110, 111\},$$

with  $\delta(abc) = e_k$  where  $k$  is the integer represented by  $abc$ . Any oracle that hard-wires additional information (while still decoding the same  $S$  under  $\sigma$ ) can be constructed so that it also decodes some entities from shorter strings (e.g., from a 2-bit prefix, or from the empty string), thereby introducing additional non-redundant descriptions and strictly enlarging  $\mathcal{R}^{\star,0}$ . Within the class  $\mathfrak{O}_{\sigma,S}$  that requires decoding exactly  $S$  under the binary style  $\sigma$ ,  $\mathcal{O}_{\text{bin}}$  is a natural candidate for minimality because it does not decode any entity from fewer than 3 bits. ■

**Proposition 2.2.4 — Characterization of minimality.** Fix  $(\sigma, S)$ . An oracle  $\mathcal{O} \in \mathfrak{O}_{\sigma,S}$  is minimal if and only if  $\mathcal{R}_{\mathcal{O}}^{\star,0}$  is a minimal element (under set inclusion) among the family  $\{\mathcal{R}_{\mathcal{O}'}^{\star,0} : \mathcal{O}' \in \mathfrak{O}_{\sigma,S}\}$ .

*Proof.* This is immediate from Definition 2.2.6 and Definition ??:  $\mathcal{O}$  is minimal precisely when there is no  $\mathcal{O}'$  with a strictly smaller non-redundant core. ■

**TODO:** Minimality is a local property.

**Proposition 2.2.5 — Non-uniqueness and non-existence.** Fix  $(\sigma, S)$ .

- 1 Minimal oracles need not be unique: there may exist distinct minimal  $\mathcal{O}_1, \mathcal{O}_2 \in \mathfrak{O}_{\sigma,S}$  with incomparable sets  $\mathcal{R}_{\mathcal{O}_1}^{\star,0}$  and  $\mathcal{R}_{\mathcal{O}_2}^{\star,0}$ .
- 2 A minimal oracle need not exist: the family  $\{\mathcal{R}_{\mathcal{O}}^{\star,0} : \mathcal{O} \in \mathfrak{O}_{\sigma,S}\}$  may contain infinite strictly descending chains under  $\subsetneq$  with no minimal element.

*Proof.* (1) Incomparability of  $\mathcal{R}^{\star,0}$ -sets yields distinct minimal elements in a partial order. (2) Standard examples of set systems admit strictly descending chains with empty intersection not attained within the system; the same phenomenon can occur here depending on how  $\mathfrak{O}_{\sigma,S}$  is specified. ■

## 2.3 Representations

In Section 2.2 we addressed the scientific representation problem by introducing an abstract oracle  $\mathcal{O}_{\mathcal{E}}$ , which evaluates finite binary strings and indicates whether each string contains sufficient information, according to an idealized standard, to reconstruct a unique entity in  $\mathcal{E}$ . Those strings constitute the set of perfect representations <sup>2</sup>.

**Definition 2.3.1** Let  $\mathcal{E}$  be a collection of entities, and let  $\mathcal{O}_{\mathcal{E}}$  be a representation oracle with oracle-internal reconstruction function  $\delta_{\mathcal{E}} : \mathcal{B}^* \rightarrow \mathcal{E}$ . We define the set of *perfect representations* for  $\mathcal{E}$  with respect to  $\mathcal{O}_{\mathcal{E}}$  as:

$$\mathcal{R}_{\mathcal{O}_{\mathcal{E}}}^* = \left\{ r \in \mathcal{B}^* \mid \delta_{\mathcal{E}}(r) \downarrow \wedge \neg \exists s \sqsubset r : (K(s \mid r \setminus s) > O(1) \wedge \delta_{\mathcal{E}}(r \setminus s) = \delta_{\mathcal{E}}(r)) \right\}.$$

The set  $\mathcal{R}_{\mathcal{O}_{\mathcal{E}}}^*$  is generally unknown, since its definition depends on the behavior of an abstract and typically inaccessible oracle. Intuitively, a representation is considered perfect if it contains all the information required for the oracle to reconstruct the corresponding entity without relying on any additional information. A perfect representation must neither include incorrect information nor omit information essential for reconstruction; otherwise, the oracle would fail to recognize it as valid.

Definition 2.3.1 allows *redundant* information within a perfect representation, but excludes *irrelevant* information. Intuitively, redundancy corresponds to information that is already implicit in the remaining part of the representation: if a substring  $s$  can be reconstructed from the rest of the string  $r \setminus s$  by a program of constant size, that is, if  $K(s \mid r \setminus s) = O(1)$ , then  $s$  introduces no essentially new information about the entity beyond what is already contained in  $r \setminus s$ . Such redundant content does not threaten perfectness and may legitimately remain in the representation. In fact, redundancy is often desirable in scientific and engineering practice, because it can facilitate the construction of usable descriptions or models: for instance, training modern machine learning systems (such as neural networks) typically requires very large datasets, which contain extensive redundancy in the form of repeated patterns, or multiple observations that jointly support the same underlying structure.

By contrast, an *irrelevant* substring is one that is both dispensable for reconstruction of the entity (i.e., removing it does not change the reconstructed entity,  $\delta_{\mathcal{E}}(r \setminus s) = \delta_{\mathcal{E}}(r)$ ) and non-redundant (i.e., it is not determined by the

---

<sup>2</sup>Recall that  $s \sqsubset r$  denotes a substring of  $r$ , and  $r \setminus s$  denotes the string obtained from  $r$  by removing all the occurrences of  $s$  (see Section A.2).

rest of the representation,  $K(s \mid r \setminus s) > O(1)$ ). Such a substring contains independent information that is not related to the entity being represented, and therefore must be removed. Otherwise, any description or model derived from the representation would be forced to account for this extraneous content, increasing complexity without improving our ability to explain or predict properties of the original entity. As a result, irrelevant information can obscure the structure that truly matters and make scientific inference more difficult.

The notion of irrelevance employed here should be understood as irrelevance relative to both the entity and the representation style. A representation is assumed to belong to a particular representational style given by the oracle (for instance, a dataset, an image, an equation, or a text description), and perfectness requires that the representation does not contain independent information that falls outside that style while leaving the reconstructed entity unchanged. For example, consider an entity  $e$  representing the data-generating mechanism behind a controlled experiment, and suppose that a dataset  $D$  of measured variables is already sufficient for the oracle to reconstruct  $e$ . If we concatenate to  $D$  a photograph taken in the laboratory, the resulting string still contains valid information about the experiment, but this additional content is not part of the dataset-style encoding of the entity and can be removed without changing  $\delta_{\mathcal{E}}$ . Since the image is not determined by the remaining dataset, it has high conditional complexity and is classified as irrelevant by the definition.

Throughout this work, we assume that a specific representation oracle  $\mathcal{O}_{\mathcal{E}}$  has been fixed for the collection  $\mathcal{E}$  under study.

**Notation 2.3.** *When the choice of oracle is clear from context, we denote the set of perfect representations simply by  $\mathcal{R}_{\mathcal{E}}^*$ . If the set of entities is also clear from the context, we denote the set of perfect representations by  $\mathcal{R}^*$ .*

**TODO:** review the rest of the section based on the new definition of perfect representation.

An entity may admit more than one perfect representation, even when the oracle is restricted to a particular representational style.

**Definition 2.3.2** Given an entity  $e \in \mathcal{E}$ , we define the set of *perfect representations of  $e$*  as

$$\mathcal{R}_e^* = \{r \in \mathcal{R}_{\mathcal{O}_{\mathcal{E}}}^* \mid \delta_{\mathcal{E}}(r) = e\}.$$

The set  $\mathcal{R}_e^*$  is oracle-relative, generally unknown, and need not be decidable.

■ **Example 2.10** Given a DNA-based encoding scheme (see Example 2.4),

an individual animal may admit multiple valid representations. Owing to the degeneracy of the genetic code, distinct DNA sequences can encode the same proteins through synonymous codons. These alternative sequences differ at the nucleotide level but preserve the information required to reconstruct the same biological entity, as they give rise to identical functional outcomes. ■

**Notation 2.4.** We write  $r_e^* \in \mathcal{R}_e^*$  to indicate that the string  $r_e^*$  is a perfect representation of the entity  $e$ , whenever such a string is known.

The sets of perfect representations associated with different entities must be disjoint.

**Proposition 2.3.1** For any two distinct entities  $e_1, e_2 \in \mathcal{E}$  with  $e_1 \neq e_2$  we have that  $\mathcal{R}_{e_1}^* \cap \mathcal{R}_{e_2}^* = \emptyset$ .

*Proof.* Suppose, for contradiction, that there exists a string  $r \in \mathcal{R}_{e_1}^* \cap \mathcal{R}_{e_2}^*$ . By definition of  $\mathcal{R}_{e_1}^*$ , we have  $\delta_{\mathcal{E}}(r) = e_1$ . Likewise, by definition of  $\mathcal{R}_{e_2}^*$ , we have  $\delta_{\mathcal{E}}(r) = e_2$ . Since  $\delta_{\mathcal{E}}$  is a function on its domain, it assigns a unique value to each  $r \in \mathcal{R}_{\mathcal{O}_{\mathcal{E}}}^*$ , and therefore  $e_1 = e_2$ , contradicting the assumption that  $e_1 \neq e_2$ . Hence no such  $r$  exists and  $\mathcal{R}_{e_1}^* \cap \mathcal{R}_{e_2}^* = \emptyset$ . ■

We distinguish between *knowable* and *unknowable* entities (see the gray areas in Figure 2.2; for example, the entity  $e_2$  is not encoded by any string). Knowable entities are those that can, in principle, be understood through scientific inquiry (see Section G.2), whereas unknowable entities lie beyond the reach of human comprehension, either because they are inherently inaccessible to observation and reasoning or they exceed the cognitive or methodological limits of science.

**Definition 2.3.3** We say that an entity  $e \in \mathcal{E}$  is *knowable* if there exists at least one  $r \in \mathcal{B}^*$  such that  $\mathcal{O}_{\mathcal{E}}(r) = \text{true}$  and  $\delta_{\mathcal{E}}(r) = e$ . An entity  $e \in \mathcal{E}$  is *unknowable* if it is not knowable.

A priori, it is not possible to determine whether an entity  $e \in \mathcal{E}$  is knowable or unknowable. Identifying suitable knowable entities for study is a matter of trial and error. In this book, we say nothing further about unknowable entities, except to note that their unknowability cannot, in general, be established in advance.

Since our framework restricts representations to finite strings, uncountable sets of entities cannot be injectively encoded using finite binary strings. This situation reflects the fact that, in certain domains of knowledge, the space of possible problems or entities may exceed the space of representations or solutions available to us.

■ **Example 2.11** If the collection of entities under study consists of real numbers, then there exist numbers that cannot be encoded using finite binary strings. This is because the set  $\mathbb{R}$  has the cardinality of the continuum, whereas the set  $\mathcal{B}^*$  is countable. ■

### Extended Set of Representations

Since the oracle defines an ideal notion of representation, the set of perfect representations  $\mathcal{R}_e^*$  of a given entity  $e \in \mathcal{E}$  can be regarded as the long-term goal of scientific inquiry about that entity. In practice, however, scientific inquiry does not proceed by direct access to elements of  $\mathcal{R}_e^*$ , but rather as a discovery process in which increasingly informative approximate representations of  $e$  are gradually developed. These approximate representations may omit information required for complete reconstruction, include incorrect or misleading content, or contain symbols that are irrelevant to the representation of  $e$ .

We say that a string represents a particular entity if it contains a non-trivial amount of algorithmic information about it. This is captured by requiring that conditioning on the string yields a reduction in the Kolmogorov complexity of a perfect representation of the entity.

**Definition 2.3.4** Let  $e \in \mathcal{E}$  be an entity. We define the set of *representations* of  $e$  as

$$\mathcal{R}_e = \{r' \in \mathcal{B}^* \mid \exists r^* \in \mathcal{R}_e^* \text{ such that } K(r^* \mid r') \leq K(r^*) - O(1)\}.$$

A representation  $r'$  is associated with an entity whenever the reduction in the conditional Kolmogorov complexity of a perfect representation  $r$  given  $r'$ , relative to the unconditional complexity of  $r$ , cannot be attributed solely to technical artifacts such as encoding conventions or machine-dependent implementation details.

■ **Example 2.12** The astronomical records used in the time of Ptolemy to describe the positions of celestial bodies throughout the year constituted an imperfect encoding of the entity "positions of celestial bodies," due to systematic observational inaccuracies. More accurate encodings were later obtained through the improved observations of Tycho Brahe, and modern astronomy provides even higher-precision representations. ■

We accept as representations strings that may be extremely poor encodings of the entities under study, and may not support even rudimentary inferences. This reflects early stages of scientific inquiry, where representations are often fragmentary, speculative, or weakly informative, yet still guide exploration and motivate further investigation. The theory of nescience ac-

commades the progressive refinement of such representations, from highly imperfect encodings to increasingly accurate ones as knowledge advances (see Chapter 3).

Unlike the case of perfect representations, the sets  $\mathcal{R}_e$  associated with different entities need not be disjoint, since a single imperfect representation may contain minimal but genuine information about more than one entity.

■ **Example 2.13** Consider the early study of infectious agents, before the distinction between viruses and bacteria was clearly established. At that stage, representations such as "a microscopic agent that causes disease and reproduces inside a host" captured minimal but genuine information about both entities. Such a representation reduces the descriptive complexity of perfect representations of viruses as well as of bacteria, without providing enough information to discriminate between them. This string would therefore be accepted as a representation of both entities, and would belong simultaneously to the sets  $\mathcal{R}_{\text{virus}}$  and  $\mathcal{R}_{\text{bacterium}}$ . ■

We are interested in the set composed of all possible representations of all knowable entities in  $\mathcal{E}$ .

**Definition 2.3.5** Let  $\mathcal{E}$  be a collection of entities, and for each  $e \in \mathcal{E}$  let  $\mathcal{R}_e \subseteq \mathcal{B}^*$  be the set of representations of  $e$ . We define the set of *representations* for  $\mathcal{E}$  as

$$\mathcal{R}_{\mathcal{E}} = \bigcup_{e \in \mathcal{E}} \mathcal{R}_e.$$

As in the case of the set of perfect representations  $\mathcal{R}_{\mathcal{E}}^*$ , the set  $\mathcal{R}_{\mathcal{E}}$  is generally unknown in practice. Although its definition is explicit, it depends on the true structure of the entities under investigation and on the existence of perfect representations, both of which are inaccessible in real scientific contexts, and so, it can't be enumerated or computed in practice.

**Notation 2.5.** *When the set of entities is clear from the context, we denote the set of representations simply by  $\mathcal{R}$ .*

Strings that do not belong to  $\mathcal{R}_{\mathcal{E}}$ , that is, strings that do not contain information about any entity in  $\mathcal{E}$ , correspond to targetless representations, which are explicitly allowed within the theory of nescience (see Sections 3.4 and G.4).

## Effective Representations

The set of representations  $\mathcal{R}$  is not accessible in practice, since it depends both on the true structure of the entities under investigation and on the unknown internal representation function. For this reason, scientific work

cannot proceed directly with  $\mathcal{R}$ , but instead relies on a further level of approximation. We therefore introduce the set  $\hat{\mathcal{R}} \subseteq \mathcal{B}^*$ , consisting of candidate strings that constitute the representations currently available within scientific practice. The elements of  $\hat{\mathcal{R}}$  are called *effective representations*: finite descriptions that can be formulated, communicated, and manipulated within existing scientific frameworks, without any guarantee of correctness or completeness. Scientific progress can then be understood as the ongoing evolution of the set of effective representations  $\hat{\mathcal{R}}$ , which progressively approximates  $\mathcal{R}$  and, in an idealized sense, approaches toward  $\mathcal{R}^*$ .

**Definition 2.3.6** Let  $\mathcal{E}$  be a set of entities. For each entity  $e \in \mathcal{E}$ , we denote by

$$\hat{\mathcal{R}}_e \subseteq \mathcal{B}^*$$

the set of *effective representations* of  $e$ , that is, the finite binary strings that are currently taken, within a given scientific context, to represent or approximate the entity  $e$ .

The notion of effective representations allows us to distinguish between known and unknown entities, in a representational or effective sense.

**Definition 2.3.7** An entity  $e \in \mathcal{E}$  is said to be *effectively known* if  $\hat{\mathcal{R}}_e \neq \emptyset$ , and *effectively unknown* otherwise.

In the theory of nescience, the notions of *effective knowledge* and *knowability* refer to two distinct and independent concepts. An entity is said to be *effectively known* if at least one effective representation is available, that is, if  $\hat{\mathcal{R}}_e \neq \emptyset$ , whereas it is said to be *knowable* if a perfect representation exists, i.e., if  $\mathcal{R}_e^* \neq \emptyset$  (see Definition 2.3.3). Effective knowledge therefore concerns what is currently represented and manipulated within scientific practice, while knowability concerns what can, in principle, be fully and correctly represented.

The typical situation in science is one in which an entity is knowable but still effectively unknown. In such cases, a perfect representation exists in principle, yet no effective representation has been discovered or constructed so far. This situation is not exceptional; on the contrary, it characterizes the overwhelming majority of entities in most scientific domains and reflects the open-ended, exploratory nature of scientific inquiry. By contrast, a far more problematic situation arises when an entity is unknowable, yet effectively known. In this case, scientists develop and refine representations of something that, in practice, does not correspond to any real or well-defined entity, as the following example illustrates.

■ **Example 2.14** The luminiferous ether was a hypothetical medium postulated to explain the propagation of light as a wave. The Michelson–Morley experiment provided strong empirical evidence against the existence of such a medium, and Einstein’s theory of special relativity later rendered the ether concept unnecessary by postulating the invariance of the speed of light in vacuum. As a result, the ether was abandoned as a physical entity. ■

The luminiferous ether is an example of effectively known entity that is unknowable, because no perfect representation corresponds to it. In such cases, scientists may develop and refine representations of an entity that ultimately does not exist. Scientific effort is then directed toward something that cannot, even in principle, be reconstructed by the oracle. Unfortunately, there is no practical way to identify such situations in advance using only finite and imperfect representations.

**Definition 2.3.8** Let  $\mathcal{E}$  be a set of entities. We define the set of *effectively known representations* for  $\mathcal{E}$  as

$$\hat{\mathcal{R}}_{\mathcal{E}} = \bigcup_{e \in \mathcal{E}} \hat{\mathcal{R}}_e.$$

The set  $\hat{\mathcal{R}}_{\mathcal{E}}$  characterizes the space of all representations that are possible in principle for the entities in  $\mathcal{E}$ , independently of whether they are currently available or known. By contrast, the set  $\hat{\mathcal{R}}_{\mathcal{E}}$  consists only of those effective representations that have actually been discovered, constructed, or adopted within scientific practice at a given time. Thus,  $\hat{\mathcal{R}}_{\mathcal{E}}$  should be understood as a time-dependent, practically accessible subset of  $\mathcal{R}_{\mathcal{E}}$ , whereas  $\mathcal{R}_{\mathcal{E}}$  itself remains a conceptual and generally inaccessible object.

**Notation 2.6.** When the set of entities is clear from the context, we denote the set of effective representations simply by  $\hat{\mathcal{R}}$ .

We can also define the set of effectively known entities, that is, the set of those entities for which at least one effective representation exists.

**Definition 2.3.9** Given a set of entities  $\mathcal{E}$  and the family of sets of effective representations  $\{\hat{\mathcal{R}}_e\}_{e \in \mathcal{E}}$ , we define the set of *effectively known entities* as

$$\hat{\mathcal{E}} = \{e \in \mathcal{E} \mid \hat{\mathcal{R}}_e \neq \emptyset\}.$$

The set  $\hat{\mathcal{E}}$  corresponds to entities for which representational activity exists or has existed within scientific practice. This includes entities that are currently under active investigation (e.g., artificial intelligence), entities that were investigated in the past but are no longer central to ongoing research (e.g., the aerodynamics of zeppelins), and entities that were once investigated but later rejected as non-existent or false (e.g., the luminiferous ether).

A further consequence of working with approximate, or effective, representations is that some candidate strings may, unbeknownst to us, fail to encode the entity we intend to study; membership in  $\hat{\mathcal{R}}_e$  reflects contextual assignment rather than guaranteed correctness.

■ **Example 2.15** In the late Eighteenth Century, chemist Joseph Priestley believed he was studying a substance called "phlogiston", which was thought to be a fire-like element released during combustion. All of his experiments and representations were constructed around this idea. However, in reality, Priestley was observing the properties of a completely different entity: oxygen. Though his descriptions were coherent and reproducible, they were ultimately anchored to the wrong entity. ■

Effective representations are monotonic over time, in the sense that the set of representations associated with an entity grows as new representations are discovered; representations that are later discarded, rejected, or superseded are not removed from the set.

**Notation 2.7.** We denote by  $\hat{\mathcal{R}}_e^t$  the set of effective representations for an entity  $e \in \mathcal{E}$  available at time  $t$ , and by  $\hat{\mathcal{E}}^t$  the set of effectively known entities at time  $t$ .

Next proposition proves the monotonicity property of effective representations over time.

**Proposition 2.3.2** Let  $\hat{\mathcal{R}}_e^{t_1}$  and  $\hat{\mathcal{R}}_e^{t_2}$  denote the set of effective representations associated with an entity  $e \in \mathcal{E}$  at times  $t_1$  and  $t_2$  respectively. Then

$$\hat{\mathcal{R}}_e^{t_1} \subseteq \hat{\mathcal{R}}_e^{t_2} \quad \text{for all } t_1 < t_2.$$

*Proof.* By assumption, representations discovered at time  $t_1$  are retained at all later times, even if they are discarded or superseded, and additional representations may be introduced. Therefore, any effective representation available at time  $t_1$  is also available at time  $t_2 > t_1$ , which implies  $\hat{\mathcal{R}}_e^{t_1} \subseteq \hat{\mathcal{R}}_e^{t_2}$ . ■

As a direct consequence, the set of effectively known entities is itself monotonic over time.

**Proposition 2.3.3** Let  $\hat{\mathcal{E}}^{t_1}$  and  $\hat{\mathcal{E}}^{t_2}$  denote the set of effectively known entities at times  $t_1$  and  $t_2$  respectively. Then

$$\hat{\mathcal{E}}^{t_1} \subseteq \hat{\mathcal{E}}^{t_2} \quad \text{for all } t_1 < t_2.$$

*Proof.* By definition  $\hat{\mathcal{E}}^t = \{e \in \mathcal{E} \mid \hat{\mathcal{R}}_e^t \neq \emptyset\}$ . If  $e \in \hat{\mathcal{E}}^{t_1}$ , then  $\hat{\mathcal{R}}_e^{t_1} \neq \emptyset$ . By the previous proposition, this implies  $\hat{\mathcal{R}}_e^{t_2} \neq \emptyset$  for all  $t_2 > t_1$ , and therefore  $e \in \hat{\mathcal{E}}^{t_2}$ . ■

The set of effectively known entities  $\hat{\mathcal{E}}$  is thus not fixed by  $\mathcal{E}$ , but by the interaction between entities, representational practices, and available inferential tools. In this sense, scientific progress can be understood as the progressive expansion and refinement of the known subset  $\hat{\mathcal{E}}$ , together with a gradual reduction of representational ambiguity.

Finally, as in the case of representations themselves, the sets of effective representations associated with different entities need not be disjoint. In particular, for two distinct entities  $e_1, e_2 \in \hat{\mathcal{E}}$ , it is possible that  $\hat{\mathcal{R}}_{e_1} \cap \hat{\mathcal{R}}_{e_2} \neq \emptyset$ , reflecting the ambiguity of some effective representations (see Example 2.13).

## 2.4 Joint Representations

Ideally, scientific work would proceed by using a perfect representation of the entity  $e$  under study, that is, an element of the set of perfect representations  $\mathcal{R}_e^*$ . However, since such perfect representations are not accessible in practice, we must instead rely on the approximate representations in  $\mathcal{R}_e$ . Certain representations are of high quality, in the sense that they encode substantial information about  $e$ . Yet,  $\mathcal{R}_e$  also includes low-quality representations, which convey only limited information about  $e$ . Moreover,  $\mathcal{R}_e$  may include representations that contain incorrect or misleading symbols: fragments that do not correspond to the entity and do not contribute to obtaining a perfect representation. Such errors may mislead us during inquiry if we treat them as valid information.

If we want to increase our knowledge about an entity, we must use representations that are as complete and correct as possible, that is, as close as possible to a perfect representation. One way to achieve this is to try different strings until we discover a high-quality representation. However, this method can be extremely time-consuming and impractical. A more efficient approach is to improve a poor representation by adding missing symbols, removing unnecessary symbols, or combining known representations, each of which contains partial information.

### Extended Representations

A natural way to improve a representation consists of adding symbols that encode the information that it is missing from the representation. This operation, which we call *extension*, plays a fundamental role in the gradual accumulation of knowledge.

**Definition 2.4.1** Let  $r, s, t \in \mathcal{B}^*$  be arbitrary binary strings and  $e \in \mathcal{E}$  be an entity, and suppose that the concatenated string  $rt$  is a representation of  $e$ , that is,  $rt \in \mathcal{R}_e$ . We call the concatenation  $rst$  an *extension* of the

representation  $rt$ , and *extension string* to the string  $s$ .

An extension may occur at any position within a representation, and both  $r$  and  $t$  are allowed to be empty. In particular, the usual case of appending symbols at the end of a representation corresponds to  $t = \lambda$ .

The extension of a representation is a representation itself.

**Proposition 2.4.1** Let  $r, s, t \in \mathcal{B}^*$  be arbitrary binary strings such that  $rt \in \mathcal{R}_e$ . Then we have that  $rst \in \mathcal{R}_e$ .

*Proof.* Assume  $rt \in \mathcal{R}_e$ . By definition of  $\mathcal{R}_e$ , there exists a perfect representation  $r^* \in \mathcal{R}_e^*$  such that

$$K(r^* \mid rt) \leq K(r^*) - O(1). \quad (2.1)$$

Let  $g : \mathcal{B}^* \rightarrow \mathcal{B}^*$  be the computable function that, on input the (effectively decodable) concatenation  $rst$ , outputs  $rt$  (i.e., it discards the middle component  $s$  and concatenates the first and third components).<sup>3</sup>

A standard monotonicity property of conditional Kolmogorov complexity states that for any computable function  $g$ ,

$$K(x \mid y) \leq K(x \mid g(y)) + O(1), \quad (2.2)$$

because a program that computes  $x$  from  $g(y)$  can be preceded by a constant-length preprocessor that computes  $g(y)$  from  $y$ .

Applying (2.2) with  $x = r^*$  and  $y = rst$  gives

$$K(r^* \mid rst) \leq K(r^* \mid g(rst)) + O(1) = K(r^* \mid rt) + O(1).$$

Combining this with (2.1) yields

$$K(r^* \mid rst) \leq (K(r^*) - O(1)) + O(1) = K(r^*) - O(1),$$

where the  $O(1)$  term remains a constant independent of the particular strings.

Therefore, there exists  $r^* \in \mathcal{R}_e^*$  such that  $K(r^* \mid rst) \leq K(r^*) - O(1)$ , and hence  $rst \in \mathcal{R}_e$  by definition. ■

The previous proposition formalizes the intended role of extension: adding correct and relevant information refines a representation without altering its referent.

As a direct consequence of the previous proposition we have the following corollary.

---

<sup>3</sup>Formally, we may regard  $rst$  as a standard computable encoding of the triple  $(r, s, t)$  into a single string (e.g., via a self-delimiting pairing/tupling function). Under any such convention, extracting  $(r, s, t)$  and outputting  $rt$  is computable.

**Corollary 2.4.2** Let  $rt \in \mathcal{R}_{\mathcal{E}}$  be a representation of an entity  $e \in \mathcal{E}$ , and let  $rst$  be an extended representation of  $rt$ . Then  $rst \in \mathcal{R}_{\mathcal{E}}$ .

*Proof.* Since  $rt \in \mathcal{R}_e$  we have that  $rst \in \mathcal{R}_e \subset \mathcal{R}_{\mathcal{E}}$ . ■

The extension of a representation does not guarantee uniqueness of reference, as next example shows.

■ **Example 2.16** Consider an entity  $e$  corresponding to a physical system described by classical mechanics, and let  $rt$  be a representation encoding its mass, position, and velocity. Suppose that the extension string  $s$  adds information about relativistic corrections. The extended representation  $rst$  may still represent the original classical system under certain approximations, but it may also correspond, for the oracle, to a different entity: a relativistic physical model. In this case,  $rst$  simultaneously belongs to  $\mathcal{R}_e$  and to  $\mathcal{R}_{e'}$  for some distinct entity  $e'$ . ■

This example illustrates an important subtlety: while extension preserves representationality and is intended to preserve entity identity, it may introduce enough additional structure for the oracle to associate the resulting string with another entity as well. Extended representations, therefore, may increase expressive power at the cost of potential ambiguity in reference.

## Reduced Representations

An alternative strategy for improving a representation consists of simplifying it by removing symbols that are believed to be incorrect or irrelevant.

**Definition 2.4.2** Let  $r, s, t \in \mathcal{B}^*$  be arbitrary binary strings and  $e \in \mathcal{E}$  be an entity, and suppose that the concatenated string  $rst$  is a representation of  $e$ , that is,  $rst \in \mathcal{R}_e$ . We call the string  $rt$ , obtained by removing the substring  $s$ , a *reduced representation* of  $rst$ .

As with extension, this definition is intentionally generic: reduction may remove symbols from any position within a representation, and the removed substring  $s$  may encode information of any kind.

The goal of a reduced representation is to produce a more accurate representation by eliminating symbols that are misleading, superfluous, or incorrectly encoded. However, unlike extension, reduction may eliminate symbols that are essential for the oracle to reconstruct the entity. Reduced representations are not always representations.

■ **Example 2.17** Let  $r$  and  $t$  be the empty strings  $\lambda$  and  $s \in \mathcal{B}^*$  a string such that  $rst \in \mathcal{R}_e$ . Then, the reduced representation  $rt$  is the empty string, and so, it is not a representation, since  $K(r^* \mid \lambda) = K(r^*)$  for all  $r^* \in \mathcal{R}^*$ . ■

Even when a reduced string remains a valid representation, reduction does not guarantee preservation of the represented entity.

■ **Example 2.18** Consider an entity  $e$  corresponding to a specific chemical compound, and suppose that  $rst$  encodes its full molecular structure, including stereochemical information. Let the substring  $s$  encode the spatial configuration of a chiral center. The reduced representation  $rt$ , which omits this information, may still correspond to a valid chemical compound, but one that is chemically distinct: a different stereoisomer. In this case,  $rt$  represents a different entity  $e'$ , even though it was obtained by reducing a representation of  $e$ . ■

The risk of reduction becomes particularly evident when the removed symbols are not known in advance to be redundant. While extension adds potentially useful information that the oracle may ignore, reduction removes information that the oracle may require. As a result, reduction does not preserve representationality and does not preserve entity identity in general.

## Joint Representations

A third, and particularly powerful, operation on representations consists of combining two existing representations into a single one. This operation, which we call *joint representation*, plays a central role in the theory of nescience, as it provides a systematic mechanism for both refining existing knowledge and discovering new entities.

■ **Definition 2.4.3** Let  $r, s \in \mathcal{R}_\top$  be two representations of an entity  $e \in \mathcal{E}$ . We define the *joint representation* of  $r$  and  $s$  as the concatenated string  $rs$ .

Joint representations serve two distinct goals. When  $r$  and  $s$  represent the same entity, their joint representation may yield a more complete or less biased representation of that entity by aggregating complementary information. When  $r$  and  $s$  represent different entities, their joint representation may correspond to a new entity not previously represented.

■ **Example 2.19** Let  $e$  be the entity corresponding to the set of causes of lung cancer. Suppose that  $r$  is a dataset obtained from a population sample consisting exclusively of male subjects, and that  $s$  is a dataset obtained from a population sample consisting exclusively of female subjects. Both  $r$  and  $s$  belong to  $\mathcal{R}_e$ , but each constitutes a biased and incomplete representation of the entity. The joint representation  $rs$  combines both datasets and yields a more informative and less biased representation of  $e$ . ■

With the following propositions we formally study the properties of the concept of joint representations.

**Proposition 2.4.3** Joint representation is associative. That is, for all  $r, s, t \in \mathcal{R}_{\mathcal{E}}$ , we have

$$(rs)t = r(st).$$

*Proof.* Joint representation is defined as string concatenation. Since string concatenation is associative, the equality  $(rs)t = r(st)$  holds for all binary strings, and therefore for all representations. ■

Despite its associativity, joint representation is not commutative. A concrete counterexample arises when  $r$  encodes a theoretical model and  $s$  encodes experimental constraints. The order in which these components are presented may affect how the oracle reconstructs the entity, leading to different outcomes.

Joint representation also satisfies an idempotence property.

**Proposition 2.4.4** For any representation  $r \in \mathcal{R}_{\mathcal{E}}$ , the joint representation  $rr$  represents the same entity as  $r$ .

*Proof.* The concatenated string  $rr$  contains no information that is not already present in  $r$ . While redundancy may increase, the oracle can ignore duplicated symbols during reconstruction. Consequently,  $rr \in \mathcal{R}_{\mathcal{E}}$  and  $\mathcal{O}_{\mathcal{E}}(rr) = \mathcal{O}_{\mathcal{E}}(r)$ . ■

We now turn to closure properties.

**Proposition 2.4.5** If  $r, s \in \mathcal{R}_{\mathcal{E}}$  and  $rs \in \mathcal{R}_{\mathcal{E}}$ , then  $rs$  is a representation.

*Proof.* This follows directly from the definition of  $\mathcal{R}_{\mathcal{E}}$  as the domain of the oracle. Whenever  $rs$  belongs to  $\mathcal{R}_{\mathcal{E}}$ , it is, by definition, a valid representation. ■

Although syntactically well defined, joint representation is not guaranteed to preserve entity identity.

**Proposition 2.4.6** Let  $r, s \in \mathcal{R}_e$  be representations of the same entity  $e$ . Then the joint representation  $rs$  does not necessarily belong to  $\mathcal{R}_e$ .

*Proof.* The joint representation  $rs$  may introduce interactions between symbols in  $r$  and  $s$  that the oracle interprets as defining a different entity. Even if  $r$  and  $s$  individually encode correct and complementary information about  $e$ , their combination may cross a representational threshold that leads the oracle to reconstruct a distinct entity  $e' \neq e$ . Hence,  $rs \in \mathcal{R}_{e'}$ . ■

This phenomenon is not a defect but a defining feature of joint representations.

■ **Example 2.20** Consider a domain in which entities are chemical compounds and representations encode molecular descriptors. Let  $r$  represent a compound with anti-inflammatory properties and let  $s$  represent a structurally related compound with antiviral properties. While both  $r$  and  $s$  individually correspond to known compounds, the joint representation  $rs$  may encode a novel molecular structure. If the oracle associates  $rs$  with a previously uncharacterized compound, then  $rs$  represents a new entity. ■

The concept of joint representation can be extended to any arbitrary, but finite, collection of representations. This allows us to incorporate multiple partial representations into our research or to use them in the process of discovering new entities.

**Definition 2.4.4** Let  $r_1, r_2, \dots, r_n \in \mathcal{R}_{\mathcal{E}}$  be a finite collection of representations. The *joint representation* of  $r_1, r_2, \dots, r_n$  is defined as the concatenated string  $r_1 r_2 \dots r_n$ .

The possibility that joint representations of known entities give rise to representations of new entities is a critical element of the theory of nescience. It provides a formal mechanism for the discovery of previously unknown entities through the systematic combination of existing knowledge, without requiring access to representations that lie outside the current epistemic horizon.

## 2.5 Descriptions

So far, our goal in working with strings from  $\mathcal{B}^*$  has been to construct encodings, or representations, that are as complete and detailed as possible for the entities in  $\mathcal{E}$ , regardless of their length. However, as stated in Chapter 1 of this book, human understanding requires the formulation of concise models of these entities, since human reasoning cannot operate effectively on lengthy representations (see Section 1.4).

■ **Example 2.21** In Example ??, we showed that a good representation of the entity "lung cancer" could be a dataset in which various risk factors are measured. However, smokers do not decide to quit smoking because they have studied and understood this extensive dataset. Rather, they do so because they understand the much simpler derived model: "smoking increases the risk of lung cancer." ■

A description or model<sup>4</sup> is a finite binary string that is mapped to a representation of an entity (see Figure 1.3 in Chapter 1). Importantly, descriptions do not directly model the entities themselves (i.e., the target systems);

<sup>4</sup>In the theory of nescience, the terms "description" and "model" are used interchangeably.

instead, they operate on representations of those entities (encodings in the form of strings) serving as approximations of the original entities through those representations.

In the theory of nescience, we require that descriptions be computable, so that the original representations can be fully and effectively reconstructed from them. This requirement of computability allows us to clearly define the limits of the concept of a "description." For example, paradoxes involving self-reference, such as the Berry paradox (i.e., "the smallest positive integer not definable in less than twelve words," see Section 1.4), can be addressed within the framework of computability.

**Definition 2.5.1 — Model.** Let  $d \in \mathcal{B}^*$  be a binary string of the form  $d = \langle TM, a \rangle$ , where  $TM$  is the encoding of a prefix-free Turing machine and  $a$  is the input string to that machine. If  $TM(a) \downarrow$ , then  $d$  is called a *description*.

Intuitively, a description consists of two parts: a Turing machine that captures and compresses the regularities present in the representation, and a string that contains what remains, that is, the incompressible or random part.

**Definition 2.5.2** We define the *set of descriptions*, denoted by  $\mathcal{D}$ , as:

$$\mathcal{D} = \{d \in \mathcal{B}^* : d = \langle TM, a \rangle \wedge TM(a) \downarrow\}.$$

Let  $r \in \mathcal{R}$  be a representation. We define the set of *descriptions for r*, denoted by  $\mathcal{D}_r$ , as:

$$\mathcal{D}_r = \{d \in \mathcal{D} : TM(a) = r\}.$$

Finally, given an entity  $e \in \mathcal{E}$ , we define the set of *descriptions for e*, denoted by  $\mathcal{D}_e$ , as:

$$\mathcal{D}_e = \{d \in \mathcal{D} : \exists r \in \mathcal{R}_e, TM(a) = r\}.$$

From an ontological point of view, descriptions are string-based objects that satisfy the additional requirement of being computable. In some cases, a description may also qualify as a representation, namely when it is accepted by the oracle as encoding an entity. In this sense, there can exist descriptions that describe other descriptions. However, in practice, it is not advisable to use descriptions as representations of entities, since what we seek in a good representation is the inclusion of as many details as possible about the original entities, rather than a concise encoding. Using descriptions in place of representations would make the task of scientific discovery considerably more difficult.

Since each description corresponds to one, and only one, representation, we can define a function that maps descriptions to representations. Given that descriptions are encoded Turing machines, it is natural to define this mapping using a universal Turing machine. As a result, not only are individual descriptions of representations computable, but the function that maps descriptions to representations is also computable.

**Definition 2.5.3** We call a *description function*, denoted by  $\delta$ , a universal Turing machine  $\delta : \mathcal{D} \rightarrow \mathcal{B}^*$  (which halts on all inputs in  $\mathcal{D}$ ) that maps descriptions to their corresponding representations.

If  $d = \langle TM, a \rangle$  is a description of the representation  $r$ , then we have that  $\delta(d) = \delta(\langle TM, a \rangle) = TM(a) = r$ .

Inspired by the principle of Occam's razor<sup>5</sup>, if two explanations are equivalent, we should prefer the shorter one. Accordingly, the limit of what can be known, or understood, about a representation, that is, its perfect model, is given by the shortest description that allows us to reconstruct that representation.

**Definition 2.5.4** Let  $\mathcal{D}_r$  be the set of descriptions of a representation  $r \in \mathcal{R}$ , and let  $d \in \mathcal{D}_r$  be a description of  $r$ . We say that  $d$  is a *perfect description* of the representation  $r$  if there is no other description  $d' \in \mathcal{D}_r$  such that  $l(d') < l(d)$ .

Recall that what we know about an entity  $e$  depends on the quality of the representation  $r$  used. If the representation  $r$  is incorrect, we cannot achieve perfect knowledge of  $e$ , even if we have found the perfect description  $d$  for  $r$ .

**Notation 2.8.** We denote by  $d_r^*$  the fact that the description  $d$  is a perfect description of the representation  $r$ .

The perfect description of a representation may not be unique; that is, there could be multiple optimal ways to compute  $r$ .

**Definition 2.5.5** Let  $\mathcal{D}_r$  be the set of descriptions of a representation  $r \in \mathcal{R}_E$ . We define the *set of perfect descriptions* for  $r$ , denoted by  $\mathcal{D}_r^*$ , as the subset of  $\mathcal{D}_r$  consisting of all perfect descriptions of  $r$ .

Unfortunately, the set of perfect descriptions of a representation is generally unknown, and as Proposition 2.5.1 shows, there exists no algorithm to compute it. In practice, we must rely on approximations to estimate how far our current best description is from a perfect one, that is, to quantify

<sup>5</sup>The Occam's razor principle refers to the number of assumptions in an explanation, not to the length of the explanation itself.

how much we do not know about a particular representation of an entity (see Chapter 5).

**Proposition 2.5.1** Let  $r \in \mathcal{B}^*$  be a representation and let  $d_r^*$  be a perfect description of  $r$ . Then we have  $l(d_r^*) = K(r)$ .

*Proof.* Apply Definition E.1.2 and note that the Turing machines  $TM$  used in descriptions of the form  $\langle TM, a \rangle$  are required to be prefix-free. ■

The actual length of a description  $l(d)$  for a representation  $r$  depends on the specific encoding of Turing machines used. This encoding method is determined by the chosen description function  $\delta$ . Fortunately, if we replace our description function with a different one, the length of perfect descriptions remains essentially unchanged, up to an additive constant that does not depend on the representation itself.

**Corollary 2.5.2** Let  $r \in \mathcal{R}$  be a representation, and let  $\delta$  and  $\dot{\delta}$  be two different description functions. Let  $d_r^*$  be a perfect description of  $r$  under  $\delta$ , and  $\dot{d}_r^*$  a perfect description under  $\dot{\delta}$ . Then  $|l(d_r^*) - l(\dot{d}_r^*)| \leq c$ , where  $c$  is a constant that does not depend on  $r$ .

*Proof.* Apply Proposition 2.5.1 and Theorem E.1.1. ■

In practice, within the theory of nescience, we are often not interested in computing the exact numerical value of the nescience associated with an entity for a given description–representation pair. Rather, our goal is to determine a weaker but sufficient notion: the relative ordering of different pairs of descriptions and representations according to their nescience. From this perspective, the specific details of the universal Turing machine used to compute description lengths are not essential<sup>6</sup>. Accordingly, for the remainder of this book we assume that the description function  $\delta$  is fixed to a reference universal Turing machine. Alternatively, the reader may interpret all theorems in this book involving the length of the shortest models as holding up to an additive constant that does not depend on the topics themselves.

A remarkable consequence of Proposition 2.5.1 is that perfect descriptions must be incompressible; that is, *perfect knowledge implies randomness* (see Section E.6).

<sup>6</sup>Do not confuse the internal workings of the universal Turing machine that maps descriptions to representations, which are not of interest here, with the internal workings of the universal oracle Turing machine that maps representations to entities, which are of interest, as understanding this mechanism is crucial to understanding how things work.

**Corollary 2.5.3** Let  $d_r^*$  be a perfect description of a representation  $r$ . Then  $d_r^*$  is incompressible, in the sense that

$$K(d_r^*) \geq l(d_r^*) - c,$$

for some constant  $c$  independent of  $r$ .

*Proof.* By Proposition 2.5.1, the length of a perfect description satisfies

$$l(d_r^*) = K(r),$$

up to an additive constant fixed by the choice of universal Turing machine. Suppose that  $d_r^*$  were compressible, that is, that there existed a description of  $d_r^*$  of length strictly smaller than  $l(d_r^*) - c$ , for arbitrarily large  $c$ . Then this shorter description could be used to construct a description of  $r$  shorter than  $d_r^*$ , contradicting the minimality of  $d_r^*$  as a perfect description. Hence,  $d_r^*$  must be incompressible. ■

The converse does not generally hold: a description can be random without being the shortest possible one (incompressibility does not imply minimality). That is, we may have a description  $d$  of a representation  $r$  such that  $l(d) = K(d)$ , yet  $l(d_r^*) < l(d)$ .

■ **Example 2.22** Consider a deep neural network with an input layer of one thousand nodes, ten hidden layers of fifty thousand nodes each, and an output layer of one thousand nodes. Suppose the network is trained to output a fixed string of one thousand 1's for any given input. The Kolmogorov complexity of a string encoding of the trained neural network is much greater than that of the output string itself, which consists of one thousand identical bits. ■

The concept of a perfect description can be generalized from individual representations to entire entities. This generalization allows us to study the nature and properties of the entities themselves.

**Definition 2.5.6** Let  $\mathcal{D}_e$  be the set of descriptions of an entity  $e \in \mathcal{E}$ . We define the *set of perfect descriptions* of the entity  $e$ , denoted by  $\mathcal{D}_e^*$ , as the subset of  $\mathcal{D}_e$  consisting of perfect descriptions. The elements of  $\mathcal{D}_e^*$  are denoted by  $d_e^*$ .

If  $d_e^* \in \mathcal{D}_e^*$  there must exist a representation  $r \in \mathcal{R}_e^*$  such that  $d_e^* \in \mathcal{D}_r^*$ .

Technically speaking, we could have descriptions that are longer than the representations they describe, that is, descriptions that do not compress the representations. However, they are epistemically uninformative, since they fail to achieve any compression.

**Definition 2.5.7** Let  $r \in \mathcal{B}^*$  be a representation, and  $d \in \mathcal{D}_r$  one of its descriptions. If  $l(d) \geq l(r)$ , we say that  $d$  is a *pleonastic description* of the representation  $r$ .

■ **Example 2.23** Consider the set of all possible finite graphs. Since graphs are abstract mathematical objects, we must represent them as strings, for instance, using a binary encoding of their adjacency matrices (see Section A.5 for an introduction to graphs). The description  $d = \langle TM, r \rangle$ , where  $r$  is the representation of a graph and  $TM$  is a Turing machine that simply halts, belongs to  $\mathcal{D}_r$  because  $TM(r) = r$ . However, this description is of limited interest, as it is likely not the shortest possible description of  $r$ . ■

It may happen that there is no shorter possible description of a representation than the representation itself. This occurs when the representation is an incompressible string. As discussed in Section E.6, the overwhelming majority of strings are incompressible. Conducting research on incompressible representations is unproductive, as it is not possible to find shorter models for such representations.

An interesting case arises when all the descriptions in  $\mathcal{D}_e$  are pleonastic, that is, there exist no models shorter than the representation for any of the possible representations of the entity. This situation would occur if all representations of the entity  $e$  are random strings. In such a case, scientific research would be fundamentally limited, as it would be impossible to find a suitable model for  $e$ . Our ability to understand and make predictions about  $e$  would then be constrained by the length of its incompressible representations.

## 2.6 Descriptions for Joint Representations

In Section ??, we introduced the concept of a joint representation  $ts$ , formed by combining two individual representations  $t$  and  $s$ . In this section, we aim to study how the length of the perfect description of a joint representation relates to the lengths of the perfect descriptions of the individual representations.

The length of the perfect description of a joint representation is greater than or equal to the length of the perfect description of either individual representation. In other words, the more information a representation contains, the longer it takes to describe.

**Proposition 2.6.1** Let  $t, s \in \mathcal{R}_{\mathcal{E}}$  be two representations, and let  $m_t^*$ ,  $m_s^*$ , and  $m_{ts}^*$  denote the perfect descriptions of the representations  $t$ ,  $s$ , and the joint representation  $ts$ , respectively. Then:  $l(m_{ts}^*) \geq l(m_t^*)$  and  $l(m_{ts}^*) \geq l(m_s^*)$ .

*Proof.* The inequality  $l(m_{ts}^*) \geq l(m_t^*)$  is equivalent to  $K(ts) \geq K(t)$ . The

result then follows from Proposition E.3.3. ■

Intuitively, adding more information to a representation is beneficial if the additional information is relevant to describing the entity of interest. However, including irrelevant information leads to unnecessarily long models. Recall that joining representations can serve either to concatenate two partial representations of the same entity or to enrich a representation by adding missing symbols.

If the selected representations partially overlap, we can exploit this redundancy to produce a joint description that is shorter than the mere concatenation of the individual descriptions. In the worst-case scenario, the perfect description of a joint representation would be equal in length to the sum of the perfect descriptions of the individual representations.

**Proposition 2.6.2** Let  $t, s \in \mathcal{R}_{\mathcal{E}}$  be two representations, and let  $m_t^*$ ,  $m_s^*$ , and  $m_{ts}^*$  denote the perfect descriptions of the representations  $t$ ,  $s$ , and the joint representation  $ts$ , respectively. Then:  $l(m_{ts}^*) \leq l(m_t^*) + l(m_s^*)$ .

*Proof.* The inequality  $l(m_{ts}^*) \leq l(m_t^*) + l(m_s^*)$  is equivalent to  $K(ts) \leq K(t) + K(s)$ . The result follows from Proposition E.3.2. ■

One interpretation of Proposition 2.6.2 is that including redundant information in the representation of an entity does not hinder our ability to find its shortest possible description. From the perspective of compression, redundancy can be eliminated during the modeling process. Therefore, in practice, we may prefer to work with representations that are longer but make the process of scientific discovery, i.e., finding the best model, easier, even if they contain superfluous information. In contrast, Proposition 2.6.1 highlights a different concern: adding irrelevant or non-informative symbols to a representation should be avoided, as they increase the complexity of the description without contributing useful information about the entity.

Finally, the following proposition shows that the order of the representations in the perfect description of a joint representation does not affect its length.

**Proposition 2.6.3** Let  $t, s \in \mathcal{R}_{\mathcal{E}}$  be two representations, and let  $m_{ts}^*$  and  $m_{st}^*$  be the perfect descriptions of the joint representations  $ts$  and  $st$ , respectively. Then:  $l(m_{ts}^*) = l(m_{st}^*)$ .

*Proof.* The equality  $l(m_{ts}^*) = l(m_{st}^*)$  is equivalent to  $K(ts) = K(st)$ . The result follows from Proposition E.3.1. ■

It is important to note, however, that joining representations is not a commutative operation, there is no guarantee that the strings  $ts$  and  $st$  encode the

same entity. Moreover, given only the concatenated string  $ts$ , it is generally not possible to recover the original representations  $t$  and  $s$ , since they are not self-delimiting.

Propositions 2.6.1, 2.6.2 and 2.6.3 can be generalized to any arbitrary, but finite, collection of representations  $t_1, t_2, \dots, t_n$ .

**Proposition 2.6.4** Let  $t_1, t_2, \dots, t_n \in \mathcal{R}_{\mathcal{E}}$  be a finite collection of representations. Then, we have that:

- i  $l(m_{t_1 t_2 \dots t_n}^*) \geq l(m_{t_i}^*) \forall 1 \leq i \leq n$ ,
- ii  $l(m_{t_1 t_2 \dots t_n}^*) \leq l(m_{t_1}^*) + l(m_{t_2}^*) + \dots + l(m_{t_n}^*)$ ,
- iii  $l(m_{t_1 \dots t_i \dots t_j \dots t_n}^*) = l(m_{t_1 \dots t_j \dots t_i \dots t_n}^*) + c \forall 1 \leq i \leq j \leq n$ ,
- iv  $l(m_{t_1 \dots t_{n-1}}^*) \leq l(m_{t_1 \dots t_{n-1} t_n}^*)$ .

*Proof.* Apply Propositions 2.6.1, 2.6.2 and 2.6.3 to individual pairs of representations  $i$  and  $j$ . ■

## 2.7 Conditional Descriptions

It is often cumbersome to include all the information required to reconstruct an entity within a single description, as this would typically result in very long strings for most entities. A more practical approach is to assume the existence of some background knowledge and to quantify our lack of knowledge about an entity relative to that background. In this section, we study the concept of *conditional descriptions*, that is, constructing a description given some prior description. Conditional descriptions also play a crucial role in the discovery of new knowledge: if conditioning a description on some prior knowledge significantly reduces the inaccuracy of a model, it indicates that this prior knowledge is relevant to understanding the entity.

**Definition 2.7.1** Let  $r, d, s \in \mathcal{B}^*$  be strings. We say that the string  $\langle d, s \rangle$  is a *valid conditional description* of the representation  $r$  given the string  $s$ , denoted by  $d_{r|s}$ , if  $d = \langle TM, a \rangle$  is a description, and  $TM(\langle a, s \rangle) = r$ .

The conditional description  $d_{r|s}$  relies on two distinct strings:  $a$  and  $s$ , each fulfilling a different role. The string  $a$  is provided as input to the Turing machine  $TM$  and is intended to contain the portion of the representation  $r$  that cannot be derived from prior knowledge, that is, the incompressible or novel part. In contrast, the string  $s$  represents background knowledge: it is a description or representation of another entity that is assumed to be already known and that can facilitate the reconstruction or understanding of  $r$ . For example, as we will explain in Chapter 5, when evaluating the redundancy of a conditional description, the contribution of the string  $s$  is disregarded—only the length and content of  $a$  are taken into account.

Note that the conditional description  $d_{r|s}$  does not belong to the set of valid descriptions  $\mathcal{D}$  for the representation  $r$ , since computing  $r$  requires the additional string  $s$ , which is not part of the description itself. Therefore, a new definition is needed to formally capture this concept.

**Definition 2.7.2** Let  $r \in \mathcal{B}^*$  be a representation and  $s \in \mathcal{B}^*$  an arbitrary string. We define the *set of conditional descriptions* of  $r$  given  $s$ , denoted by  $\mathcal{D}_{r|s}$ , as:

$$\mathcal{D}_{r|s} = \{d \in \mathcal{B}^*, d = \langle TM, a \rangle : TM(\langle a, s \rangle) = r\}.$$

For each representation  $r \in \mathcal{B}^*$ , there always exists a conditional description  $d_{r|s}$  that describes  $r$ , as the following proposition shows.

**Proposition 2.7.1** Let  $r \in \mathcal{B}^*$  be a representation and  $s \in \mathcal{B}^*$  an arbitrary string. If  $d \in \mathcal{D}_r$  then  $d \in \mathcal{D}_{r|s}$ .

*Proof.* We can construct a conditional description  $\langle \langle TM, a \rangle, s \rangle$  based on a Turing machine  $TM$  such that, when given the input  $\langle a, s \rangle$ , the machine safely ignores the string  $s$ . ■

The converse of Proposition 2.7.1 is not true. The fact that  $d$  is a conditional description (i.e.,  $d \in \mathcal{D}_{r|s}$ ) does not imply that  $d$  is also a valid description (i.e.,  $d \in \mathcal{D}_r$ ). Indeed, while we require that  $TM(\langle a, s \rangle) = r$ , we do not require that  $TM(a) = r$ , and in general, this may not hold.

We are interested in the concept of a perfect conditional description. The perfect conditional description of a representation, given some prior knowledge, is the shortest possible string that allows us to fully reconstruct the representation, assuming that the prior knowledge is already known.

**Definition 2.7.3** Let  $r \in \mathcal{B}^*$  be a representation, and let  $d_{r|s}^*$  be the shortest possible description of  $r$  given the string  $s$ . We call  $d_{r|s}^*$  the *perfect conditional description* of the representation  $r$  given the string  $s$ , or simply perfect conditional description of  $r$  given  $s$  for short.

Note that  $d_{r|s}^*$  is a perfect description of the representation  $r$  \*conditional on\* the string  $s$ . This does not imply that  $s$  is a perfect description itself; it may be an incomplete or partially irrelevant representation. In such a case, we would have achieved perfect knowledge with respect to the  $d$  component, but not with respect to the  $s$  component of the combined string  $\langle d, s \rangle$ .

The length of a perfect conditional description is always less than or equal to that of its unconditional counterpart. In other words, assuming the existence of some background knowledge can reduce the effort required to describe a representation.

**Proposition 2.7.2** Let  $r \in \mathcal{B}^*$  be a representation and  $s \in \mathcal{B}^*$  an arbitrary string. Then  $l(d_{r|s}^*) \leq l(d_r^*)$ .

*Proof.* The inequality  $l(d_{r|s}^*) \leq l(d_r^*)$  is equivalent to the well-known result  $K(r | s) \leq K(r)$ . The proposition follows directly by applying Proposition E.4.3. ■

The notions of unconditional, conditional, and joint descriptions are closely related. In particular, the availability of prior knowledge (as captured by a conditional description) can reduce the length of a description, while describing multiple entities jointly (via a joint description) typically requires more information than describing a single entity. The following proposition formalizes these relationships by comparing the lengths of the perfect conditional description, the perfect (unconditional) description, and the perfect joint description.

**Proposition 2.7.3** Let  $r, s \in \mathcal{B}^*$  two different representations. Then:

$$l(d_{r|s}^*) \leq l(d_r^*) \leq l(d_{rs}^*)$$

*Proof.* The inequality  $l(d_{r|s}^*) \leq l(d_r^*) \leq l(d_{rs}^*)$  is equivalent to the Kolmogorov complexity relations  $K(r | s) \leq K(r)$  and  $K(r) \leq K(rs)$ . The result follows directly by applying Proposition E.4.5. ■

As was the case with joint descriptions, the concept of conditional description can be naturally extended to finite collections of representations.

**Definition 2.7.4** Let  $r, d, s_1, s_2, \dots, s_n \in \mathcal{B}^*$  be strings. We say that the string  $\langle d, s_1, s_2, \dots, s_n \rangle$  is a *valid conditional description* of the representation  $r$  given the strings  $s_1, s_2, \dots, s_n$ , denoted by  $d_{r|s_1, s_2, \dots, s_n}$ , if  $d = \langle TM, a \rangle$  is a description, and  $TM(\langle a, s_1, s_2, \dots, s_n \rangle) = r$ .

The following definition generalizes the notion of a perfect conditional description to the case of multiple conditioning strings.

**Definition 2.7.5** Let  $r \in \mathcal{B}^*$  be a representation, and let  $d_{r|s_1, s_2, \dots, s_n}^*$  be the shortest possible description of  $r$  given the strings  $s_1, s_2, \dots, s_n$ . We call  $d_{r|s_1, s_2, \dots, s_n}^*$  the *perfect conditional description* of the representation  $r$  given the string  $s_1, s_2, \dots, s_n$ , or perfect conditional description of  $r$  given  $s_1, s_2, \dots, s_n$  for short.

The next proposition generalizes Propositions 2.7.2 and 2.7.3 to any arbitrary (but finite) collection of strings  $s_1, s_2, \dots, s_n$ . In particular, it shows

that the more background knowledge we assume for a given representation, the shorter its perfect description becomes.

**Proposition 2.7.4** Let  $r, s_1, s_2, \dots, s_n \in \mathcal{B}^*$  be a finite collection of strings. Then:

$$l(d_{r|s_1, s_2, \dots, s_n}^*) \leq l(d_r^*) \leq l(d_{r, s_1, s_2, \dots, s_n}^*)$$

*Proof.* This follows from the Kolmogorov complexity inequalities  $K(r | s_1, s_2, \dots, s_n) \leq K(r) \leq K(r, s_1, s_2, \dots, s_n)$ , which generalize the results stated in Propositions 2.7.2 and 2.7.3. ■

The following proposition further generalizes the idea that assuming additional background knowledge cannot increase the length of a perfect conditional description.

**Proposition 2.7.5** Let  $r, s_1, s_2, \dots, s_n, s_{n+1} \in \mathcal{B}^*$  be a finite collection of strings. Then:

$$l(d_{r|s_1, s_2, \dots, s_n, s_{n+1}}^*) \leq l(d_{r|s_1, s_2, \dots, s_n}^*)$$

*Proof.* This follows directly from the monotonicity property of conditional Kolmogorov complexity: adding more conditioning information cannot increase the complexity. Formally,  $K(r | s_1, s_2, \dots, s_n, s_{n+1}) \leq K(r | s_1, s_2, \dots, s_n)$ . ■

## 2.8 Research Areas

Entities can be grouped into research areas. The concept of an area is useful insofar as all the entities included in the area are related to a common subdomain of knowledge or share a common property. The specific criteria used for grouping depend on the practical application of the theory of nescience.

**Definition 2.8.1** Given a set of entities  $\mathcal{E}$ , we define a *research area*  $\mathcal{A}$  as a subset of entities,  $\mathcal{A} \subset \mathcal{E}$ .

If we want to quantify how much we do not know about a research area, we must first provide a representation for that area. In general, research areas are infinite, but the number of known representations is finite. Therefore, we can only describe an area with respect to our current state of knowledge.

**Definition 2.8.2** Let  $\mathcal{A} \subset \mathcal{E}$  be a research area. We define the *known subset of the area*  $\mathcal{A}$ , denoted by  $\hat{\mathcal{A}}$ , as the set of entities  $e_1, e_2, \dots, e_n \in \mathcal{A}$  for which at least one non-pleonastic description is known.

We must distinguish between the knowable subset of  $\mathcal{A}$ , composed of those entities for which a representation exists, and the known subset of  $\mathcal{A}$ , composed of those entities for which at least one non-pleonastic description is known, that is, entities about which some research has already been conducted. Clearly, the set of known entities is a subset of the set of knowable entities.

As our understanding of a research area evolves, the number of entities included in its known subset also changes. Throughout this book, the properties of research areas will always be considered relative to our current state of knowledge.

**Definition 2.8.3** Let  $\mathcal{A} \subset \mathcal{E}$  be a research area with known subset  $\hat{\mathcal{A}} = \{e_1, e_2, \dots, e_n\}$ , and let  $R_{\hat{\mathcal{A}}} = \{r_1, r_2, \dots, r_n\}$  be a set of representations such that  $r_i \in \mathcal{R}_{e_i}$ . We call  $R_{\hat{\mathcal{A}}}$  a *representation of the area  $\mathcal{A}$*  given the known subset  $\hat{\mathcal{A}}$ , abbreviated as *representation of  $\mathcal{A}$* .

In a similar manner to how we describe individual entities, we can also introduce the concept of a description for an entire research area. Since a research area is represented by a collection of known representations corresponding to its known subset, a description of the area must account for the generation of this entire set. Thus, we define a description of a research area as a program that, when executed, produces the sequence of representations associated with the known entities in that area.

**Definition 2.8.4** Let  $R_{\hat{\mathcal{A}}} = \{r_1, r_2, \dots, r_n\}$  be the representation of an area  $\mathcal{A}$ . We call a *description of the area  $\mathcal{A}$*  given the known subset  $\hat{\mathcal{A}}$ , abbreviated as *description of  $\mathcal{A}$* , and denoted by  $d_{\hat{\mathcal{A}}}$ , to any string in the form  $\langle TM, a \rangle$  such that the Turing machine  $TM$ , when given input  $a$ , outputs the sequence  $\langle r_1, r_2, \dots, r_n \rangle$ .

We can also consider all possible descriptions that generate the full set of known representations for a given research area. These descriptions differ in structure, length, or computational efficiency, but they all produce the same output: the sequence of representations associated with the known entities in the area.

**Definition 2.8.5** Let  $R_{\hat{\mathcal{A}}} = \{r_1, r_2, \dots, r_n\}$  be the representation of an area  $\mathcal{A}$ . We define the set of *descriptions for  $R_{\hat{\mathcal{A}}}$* , denoted by  $\mathcal{D}_{R_{\hat{\mathcal{A}}}}$ , as:

$$\mathcal{D}_{R_{\hat{\mathcal{A}}}} = \{d \in \mathcal{D} : TM(a) = \langle r_1, r_2, \dots, r_n \rangle\}.$$

Finally, we are interested in identifying the perfect model for a research area, that is, the shortest possible string that fully describes its known subset. According to Definition 2.5.7, if we are aware of the existence of an entity

$e \in A$ , then  $e$  should be included in the known subset  $\hat{A}$ , even if no research has yet been conducted on that specific topic.

**Definition 2.8.6** Let  $A \subset \mathcal{E}$  be an area with known subset  $\hat{A}$ , and let  $d_{\hat{A}}^* \in \mathcal{D}_{R_{\hat{A}}}$  be the shortest possible description of  $A$ . We call  $d_{\hat{A}}^*$  the *perfect description of the area  $A$*  given the known subset  $\hat{A}$ , abbreviated as *perfect description of  $A$* .

The following proposition shows the relationship between the description of a research area and the descriptions of the individual entities that compose its known subset. In general, the models for an area are not simply the collection of the models for each individual topic; instead, a joint model may offer a more concise description.

**Proposition 2.8.1** Let  $A \subset \mathcal{E}$  be an area with known subset  $\hat{A} = \{e_1, e_2, \dots, e_n\}$ , then we have that  $l(d_{\hat{A}}^*) \leq l(d_{e_1}^*) + l(d_{e_2}^*) + \dots + l(d_{e_n}^*)$ .

*Proof.* Apply Proposition 2.6.4-ii. ■

Moreover, as shown in Proposition 2.6.4, the order in which the representations are listed in the description of an area does not affect the length of its perfect model.

Research areas can overlap; that is, given two areas  $A$  and  $B$ , it may be the case that  $A \cap B \neq \emptyset$ . Furthermore, one area can be a subset of another, forming a hierarchy of areas. In this context, we are particularly interested in how the length of perfect models for some areas compares to the length of perfect models for related areas.

**Proposition 2.8.2** Let  $A, B \subset \mathcal{E}$  be two areas such that  $A \subset B$ , and let  $\hat{A}$  and  $\hat{B}$  be their known subsets respectively, then we have that  $l(d_{\hat{A}}^*) \leq l(d_{\hat{B}}^*)$ .

*Proof.* Since  $A \subset B$ , it follows that  $\hat{A} \subset \hat{B}$ . Let

$$R_{\hat{A}} = \{r_1, r_2, \dots, r_m\} \quad \text{and} \quad R_{\hat{B}} = \{r_1, r_2, \dots, r_m, r_{m+1}, \dots, r_n\}$$

be the sets of representations corresponding to  $\hat{A}$  and  $\hat{B}$  respectively, and let  $d_{\hat{B}}^*$  be the perfect description of  $\hat{B}$ . Then, we can construct a description  $d'$  of  $\hat{A}$  by modifying  $d_{\hat{B}}^*$  to output only the subset  $R_{\hat{A}}$ . This can be achieved by appending a simple postprocessing step that discards the extra representations. The additional cost of this truncation is at most a constant number of bits, independent of the specific contents of  $R_{\hat{B}}$ .

Formally, we have:

$$l(d_{\hat{A}}^*) \leq l(d') \leq l(d_{\hat{B}}^*) + c$$

for some constant  $c$ . But since  $d_{\hat{A}}^*$  is the shortest possible description of  $R_{\hat{A}}$ , we conclude:

$$l(d_{\hat{A}}^*) \leq l(d_{\hat{B}}^*).$$

■

The following proposition shows how the length of the shortest possible description of two areas relates to the length of the description of their union and intersection.

**Proposition 2.8.3** Let  $A, B \subset \mathcal{E}$  be two areas with known subsets  $\hat{A}$  and  $\hat{B}$  respectively, then we have that  $l(d_{\hat{A} \cup \hat{B}}^*) = l(d_{\hat{A}}^*) + l(d_{\hat{B}}^*) - l(d_{\hat{A} \cap \hat{B}}^*)$ .

*Proof.* Let  $R_{\hat{A}}$ ,  $R_{\hat{B}}$ , and  $R_{\hat{A} \cap \hat{B}}$  be the sets of representations corresponding to the known subsets  $\hat{A}$ ,  $\hat{B}$ , and  $\hat{A} \cap \hat{B}$ , respectively.

From the theory of Kolmogorov complexity, the minimal description length of the union of two finite sets of strings satisfies the following identity:

$$K(R_{\hat{A} \cup \hat{B}}) = K(R_{\hat{A}}) + K(R_{\hat{B}}) - K(R_{\hat{A} \cap \hat{B}}),$$

Since, by definition, the perfect description  $d_{\hat{A}}^*$  satisfies:

$$l(d_{\hat{A}}^*) = K(R_{\hat{A}}), \quad l(d_{\hat{B}}^*) = K(R_{\hat{B}}), \quad l(d_{\hat{A} \cap \hat{B}}^*) = K(R_{\hat{A} \cap \hat{B}}),$$

it follows that:

$$l(d_{\hat{A} \cup \hat{B}}^*) = K(R_{\hat{A} \cup \hat{B}}) = l(d_{\hat{A}}^*) + l(d_{\hat{B}}^*) - l(d_{\hat{A} \cap \hat{B}}^*).$$

■

A consequence of Proposition 2.8.3 is that  $l(d_{\hat{A} \cup \hat{B}}^*) \leq l(d_{\hat{A}}^*) + l(d_{\hat{B}}^*)$ , that is, when we combine two different research areas, how much we do not know about these areas decreases.

Just as we introduced a chain rule for entropy in Proposition D.4.5, we can also establish a chain rule for the shortest length of a description of a research area.

**Proposition 2.8.4** Let  $A, B \subset \mathcal{E}$  be two areas with known subsets  $\hat{A}$  and  $\hat{B}$ , then we have that  $l(d_{\hat{A} \cup \hat{B}}^*) = l(d_{\hat{A}}^*) + l(d_{\hat{B} \setminus \hat{A}}^*)$ .

*Proof.* Let  $R_{\hat{A}}$  be the set of representations associated with  $\hat{A}$ , and let  $R_{\hat{B} \setminus \hat{A}}$  be the set of representations corresponding to entities in  $\hat{B}$  that are not in  $\hat{A}$ .

By definition, the known subset of the union  $\hat{A} \cup \hat{B}$  corresponds to the set of representations:

$$R_{\hat{A} \cup \hat{B}} = R_{\hat{A}} \cup R_{\hat{B} \setminus \hat{A}}.$$

Let  $d_{\hat{A}}^*$  be the shortest (perfect) description that generates  $R_{\hat{A}}$ , and let  $d_{\hat{B} \setminus \hat{A}}^*$  be the shortest description that generates  $R_{\hat{B} \setminus \hat{A}}$ . Because the two subsets are disjoint, we can concatenate these two descriptions to produce a description of  $R_{\hat{A} \cup \hat{B}}$ .

Hence, the length of the shortest description of the union satisfies:

$$l(d_{\hat{A} \cup \hat{B}}^*) \leq l(d_{\hat{A}}^*) + l(d_{\hat{B} \setminus \hat{A}}^*).$$

To prove equality, assume there exists a shorter description  $d'$  for  $\hat{A} \cup \hat{B}$  such that:

$$l(d') < l(d_{\hat{A}}^*) + l(d_{\hat{B} \setminus \hat{A}}^*).$$

Then, one could extract from  $d'$  both  $R_{\hat{A}}$  and  $R_{\hat{B} \setminus \hat{A}}$ , which would imply that at least one of  $d_{\hat{A}}^*$  or  $d_{\hat{B} \setminus \hat{A}}^*$  is not minimal—contradicting the assumption that they are perfect descriptions.

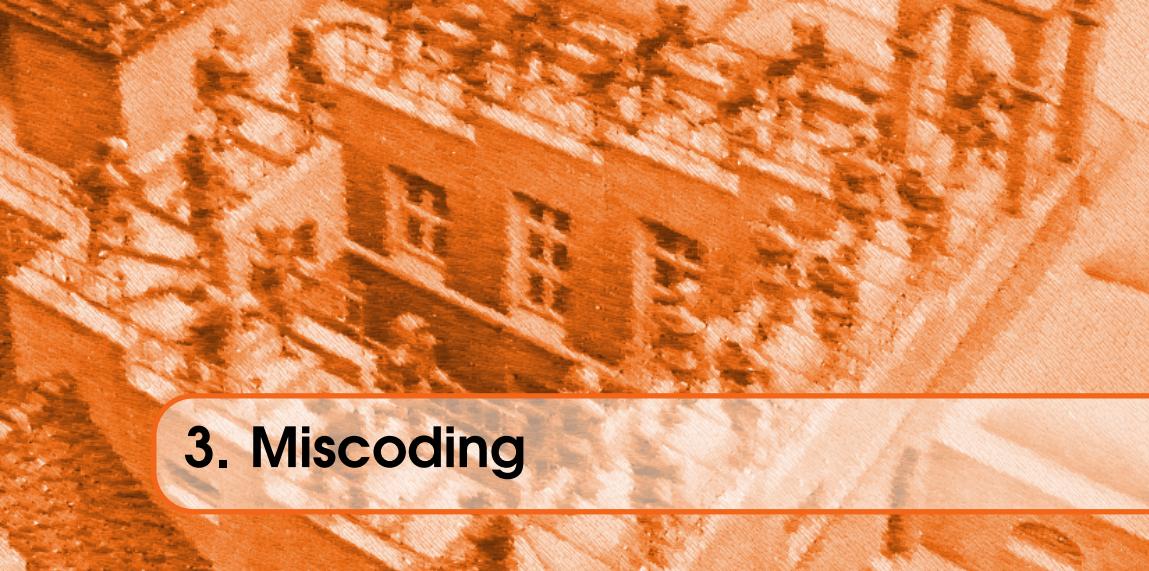
Therefore:

$$l(d_{\hat{A} \cup \hat{B}}^*) = l(d_{\hat{A}}^*) + l(d_{\hat{B} \setminus \hat{A}}^*).$$

■

## 2.9 References

For more information about Russell’s paradox, Cantor’s theorem, and universal sets, refer, for example, to [Jec13]. This book also covers the Zermelo–Fraenkel system of axioms, including the Axiom of Choice. The idea of using a function to assign to each symbol and well-formed formula of a formal language a unique natural number (called a Gödel number) was introduced by Kurt Gödel in his proof of the incompleteness theorems [Göd31]. A detailed discussion of the Berry paradox from the perspective of computability can be found in [Cha95]. For a review of the problem of representation in Kolmogorov complexity, as well as a detailed account of the implications of Kolmogorov complexity being defined only up to an additive constant, see [LV13]. That oracle machines are not mechanical was originally stated by Turing when he introduced the concept of the oracle machine in [Tur39]. For a comprehensive review of oracle Turing machines, refer to [Rob15].



## 3. Miscoding

*All great work is the fruit of patience and perseverance,  
combined with tenacious concentration on a subject  
over a period of months or years.*

Santiago Ramón y Cajal

In many areas of science, the entities we aim to study (the set  $\mathcal{E}$ ) often include abstract concepts or complex phenomena that are difficult to investigate directly. These may be ideas or processes that resist straightforward description or modeling. To gain insight into such entities, we must often rely on indirect methods. As discussed in the previous chapter, the theory of nescience suggests using representations, that is, sequences of symbols, rather than attempting to interact directly with the entities themselves.

However, this approach introduces its own challenges. Because our understanding of the elements in  $\mathcal{E}$  is incomplete (otherwise, further research would be unnecessary), the representations we construct are typically imperfect, they lack the completeness and precision we would ideally desire. These imperfections can lead to significant problems: inaccuracies in representation can propagate into the models derived from them, potentially distorting our interpretation and understanding. It is therefore essential to recognize this source of error and to carefully consider its implications.

To address this issue, we introduce the concept of *miscoding*, a measure used to quantify the error that arises from misrepresenting or inaccurately encoding entities. Miscoding is defined in terms of the length of the shortest computer program capable of transforming an incorrect representation into a correct one. In essence, it measures the amount of effort, as reflected by the program's length, required to correct a faulty representation.

However, there is a significant complication: applying this idea in practice is not straightforward, since the perfect representations of entities are unknown. As a result, we cannot directly measure the gap between a given representation and an ideal one. Nevertheless, we propose a theoretical framework based on the oracle machine (an abstract construct) used to define the set  $\mathcal{E}$ . This oracle machine is assumed to be capable of recognizing valid representations for all entities in the set. It is important to note, however, the limitations of this approach; for instance, we cannot query the oracle about a specific entity unless we already possess a valid representation of it.

Getting a better grasp on scientific representation and the challenges in reducing miscoding could really push forward scientific research, helping us create better, more thorough models of the natural world. In this chapter, we are going to properly introduce the idea of miscoding and look into its various characteristics. We will examine how miscoding behaves when it comes to combined representations and discuss methods to reduce miscoding. We will delve into the idea that there are strings that do not represent any entity. And we will dig into why miscoding is important in different research areas and talk about its potential to open up new lines of inquiry.

Gaining a deeper understanding of scientific representation and the difficulties involved in reducing miscoding can significantly advance scientific research, enabling the development of more accurate and comprehensive models of the natural world. In this chapter, we formally introduce the concept of *miscoding* and explore its key properties. We will analyze how miscoding behaves in the context of combined representations and consider strategies for reducing it. We will also examine the notion that some strings fail to represent any entity and investigate the implications of this idea. Finally, we will highlight the relevance of miscoding across various research domains and discuss its potential to inspire new lines of inquiry.

### 3.1 Miscoding

In an ideal scenario, we would ask the oracle to determine how far a particular string  $r$  is from perfectly encoding an entity  $e$ . This, however, presents a complication: we can only refer to entity  $e$  through a perfect representation from the set  $\mathcal{R}_e^*$ . Unfortunately, we typically do not have access to these

perfect representations and, therefore, cannot use them in our query.

To overcome this limitation, we propose an alternative approach. Rather than asking the oracle to assess the discrepancy between our current representation  $r$  and a valid representation of the specific entity  $e$ , we instead we identify the smallest distance between  $r$  and all valid representations of all entities in the set  $\mathcal{E}$ , that is, the elements of  $\mathcal{R}_\mathcal{E}^*$ .

**Definition 3.1.1 — Miscoding.** Let  $r \in \mathcal{B}^*$  be a representation. We define the *miscoding* of  $r$ , denoted by  $\mu(r)$ , as:

$$\mu(r) = \min_{r_e^* \in \mathcal{R}_\mathcal{E}^*} \frac{\max\{K(r | r_e^*), K(r_e^* | r)\}}{\max\{K(r), K(r_e^*)\}}$$

where  $\min^o$  indicates that the minimum is to be computed by an oracle.

Intuitively, our lack of knowledge about an entity is reflected in a higher miscoding value of its current representation. Conversely, a deeper understanding of the entity should lead to an encoding that is closer to a valid representation, and thus exhibits lower miscoding.

Miscoding is computed using a bidirectional approach: we require the oracle to determine both the length of the shortest computer program that can generate the valid representation  $r_e^*$  from our current representation  $r$  (i.e.,  $K(r_e^* | r)$ ), and the length of the shortest program that can generate  $r$  from  $r_e^*$  (i.e.,  $K(r | r_e^*)$ ). This bidirectionality captures the maximum difficulty of transforming one representation into the other. A high-quality representation should allow for an easy reconstruction of the correct representation and vice versa, implying that both of these conditional complexities are low. In other words, an ideal representation contains all the information needed to recover the correct encoding of the entity, without introducing any erroneous or irrelevant symbols.

While miscoding concerns whether the symbols in a representation are relevant, that is, whether they correctly refer to the intended entity, it is not concerned with how many symbols are used. In contrast, surfeit (as discussed in Chapter 5) focuses on whether the symbols are essential, aiming to minimize the number of unnecessary or redundant symbols. In this sense, miscoding relates to the correctness of the content, while surfeit relates to the efficiency of its expression. Although both concepts address flaws in representation, they capture fundamentally different types of representational inefficiency.

Our definition of miscoding is formulated as a relative measure rather than an absolute one. Instead of providing an isolated score for a single representation, it quantifies how far a representation is from the nearest

valid representation, normalized by their respective lengths. This relative formulation allows us to meaningfully compare the miscoding of different representations of the same entity, and also to compare the miscoding values across representations of different entities, regardless of their size or complexity.

The miscoding value of a representation  $r$  always lies within the interval  $[0, 1]$ .

**Proposition 3.1.1** For all  $r \in \mathcal{B}^*$ , it holds that  $0 \leq \mu(r) \leq 1$ .

*Proof.* This follows directly from the inequality  $0 \leq \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}} \leq 1$  which holds for all  $x, y \in \mathcal{B}^*$ , as stated in Proposition E.5.5. ■

Miscoding is zero if, and only if, the representation  $r$  is a valid representation of some entity  $e$ .

**Proposition 3.1.2** Given a representation  $r \in \mathcal{B}^*$ , we have  $\mu(r) = 0$  if, and only if,  $r \in \mathcal{R}_e^*$ .

*Proof.* If  $r \in \mathcal{R}_e^*$ , then there exists an entity  $e \in \mathcal{E}$  such that  $r = r_e^*$ . In this case,  $K(r | r_e^*) = 0$ , and thus  $\mu(r) = 0$ .

Conversely, if  $\mu(r) = 0$ , then there must exist some  $r_e^* \in \mathcal{R}_e^*$  such that  $K(r | r_e^*) = 0$  and  $K(r_e^* | r) = 0$ , which implies that  $r = r_e^*$ , and hence  $r \in \mathcal{R}_e^*$ . ■

According to Proposition 3.1.2, a miscoding value of zero implies that  $r$  perfectly encodes some entity  $e$ . However, identifying the exact entity that  $r$  represents remains a challenge. While scientific intuition may offer plausible hypotheses about the encoded entity, such guesses cannot be confirmed mathematically. Moreover, as scientific understanding evolves, our interpretation of what  $r$  encodes may also change over time (see Example 1.4).

It has been observed that an entity  $e \in \mathcal{E}$  may possess multiple valid representations, as captured by the set  $\mathcal{R}_e^*$ . Fortunately, the value of miscoding is invariant under the choice of valid representation (see the discussion on style in Section G.4).

**Proposition 3.1.3** For any valid representation  $r_e^* \in \mathcal{R}_e^*$ , it holds that  $\mu(r_e^*) \leq \mu(r)$  for all  $r \in \mathcal{B}^*$ .

*Proof.* This follows from the fact that  $\mu(r_e^*) = 0$  and  $\mu(r) \geq 0$  for all  $r \in \mathcal{B}^*$ . ■

For a given entity  $e$ , all valid representations in the set  $\mathcal{R}_e^*$  are equally adequate with respect to miscoding, as each yields a miscoding value of 0. From a practical standpoint, however, the most suitable representation

is the one that best facilitates the discovery of new knowledge about the entity—specifically, the representation that most effectively supports the construction of explanatory models.

## 3.2 Joint Miscoding

Section ?? introduced the notion of a *joint representation*, which arises when two representations  $s, t \in \mathcal{B}^*$  are concatenated to form a new string  $st$ . This section explores the properties of miscoding as they apply to such joint representations.

Since the concatenated string  $st$  is itself a valid representation, it is not necessary to introduce a separate definition of miscoding for joint representations. The miscoding of the joint representation is given by:

$$\mu(st) = \min_{r_e^* \in \mathcal{R}_E^*} \frac{\max\{K(st | r_e^*), K(r_e^* | st)\}}{\max\{K(st), K(r_e^*)\}}$$

Consistent with Proposition 3.1.1, the miscoding of a joint representation is a value between 0 and 1, i.e.,  $0 \leq \mu(st) \leq 1$ .

It is important to note that supplementing an incomplete representation, that is, one with a positive miscoding value, with additional symbols does not necessarily lead to a reduction in miscoding. This is because the added symbols may introduce irrelevant or incorrect information. Conversely, miscoding does not necessarily increase either, since incorporating relevant and accurate symbols can reduce the miscoding of an incomplete representation.

From a formal perspective, given two arbitrary representations  $s, t \in \mathcal{B}^*$ , there is no guarantee that any of the following inequalities will always hold:  $\mu(ts) \geq \mu(t)$ ,  $\mu(ts) \leq \mu(t)$ ,  $\mu(ts) \geq \mu(s)$ , or  $\mu(ts) \leq \mu(s)$ .

■ **Example 3.1** Consider the text  $r$  of a biology research article that succinctly and accurately describes a newly discovered specimen, resulting in a low miscoding value  $\mu(r)$ . However, when additional text  $s$ , taken from a completely unrelated article about a second specimen from a different species, is appended, the resulting concatenated representation  $rs$  muddles the original focus. This inclusion of unrelated content increases the miscoding value to  $\mu(rs)$ , illustrating a case in which  $\mu(rs) \geq \mu(r)$ . Although the added text is scientifically valid on its own, it dilutes the clarity and precision of the original article, thereby increasing its miscoding.

Conversely, in a second scenario, consider another research article  $r'$  that exhibits a relatively high miscoding value  $\mu(r')$ , due to an incomplete description of the specimen. If this article is supplemented with additional text  $s'$ , containing essential information previously omitted, the resulting concatenated representation  $r's'$  becomes significantly more informative

and coherent. As a result, the miscoding value decreases, demonstrating a case where  $\mu(r's') \leq \mu(r')$ . This highlights that enriching an incomplete representation with relevant and focused content can yield a more accurate and lower-miscoding representation. ■

Furthermore, it should not be assumed that the miscoding of a joint representation is less than or equal to the sum of the miscodings of the individual representations, i.e.,  $\mu(st) \leq \mu(s) + \mu(t)$ . This inequality does not necessarily hold, even when both  $s$  and  $t$  encode the same entity. The reason is that the joint representation  $\$st\$$  may end up encoding an entirely different entity from those represented by  $s$  and  $t$  individually.

■ **Example 3.2** In a medical research context, representation  $r$  describes a new drug compound  $A$  for treating a specific type of cancer, while representation  $s$  outlines a genetic mutation  $B$  associated with that cancer type. Both  $r$  and  $s$  have low miscoding values, denoting accurate, focused content. However, when concatenated, the combined representation  $rs$  unintentionally suggests a third entity  $C$ , implying a causal relationship between  $A$  and  $B$ . This unintended implication stems from the mixture of distinct information, leading to a higher miscoding value. In this instance, each of  $r$ ,  $s$ , and  $rs$  essentially represents different research entities.  $r$  and  $s$  are clear in their individual contexts, but  $rs$  is ambiguous, exemplifying a case where concatenated information can inadvertently create a representation of an entirely different, unintended entity, thus increasing miscoding. ■

Given the non-commutative nature of joining two representations, it is not guaranteed that  $\mu(ts) = \mu(st)$ . This is because the composite strings  $ts$  and  $st$  may encode entirely different entities. This property holds true even when we restrict our attention to valid representations of a specific entity  $e$ . For example, if  $r_e^*, s_e^* \in \mathcal{R}_e^*$  are two valid representations of the entity  $e$ , there is no assurance that the concatenated string  $r_e^*s_e^*$  will itself be a valid representation of  $e$ .

■ **Example 3.3** In environmental science, representation  $r$  thoroughly examines microplastics, their nature and harmful impacts on marine ecosystems, while representation  $s$  focuses on specific technologies and methodologies for detecting these pollutants in water. The concatenation  $rs$  presents a logical progression: first describing the pollutant and its effects, then outlining detection methods. This creates a coherent representation of a broader entity, one concerned with the pollutant, its impact, and how to detect it, an entity which may not yet be fully understood, hence yielding a higher miscoding value.

Conversely, the concatenation  $sr$  begins with technical methodologies

for detection and then provides the contextual background on microplastics and their environmental effects. This ordering may be more familiar within a methodological framework and results in a more coherent and well-understood entity, yielding a lower miscoding value.

In this particular case, we observe that  $\mu(rs) > \mu(sr)$ . ■

The concept of miscoding can be naturally extended to joint representations formed from any finite collection of representations. Let  $r_1, r_2, \dots, r_n \in \mathcal{B}^*$  be a finite collection of representations. The miscoding of their concatenation, denoted by  $r_1 r_2 \dots r_n$ , is defined as:

$$\mu(r_1 r_2 \dots r_n) = \min_{r_e^* \in \mathcal{R}_e^*} \frac{\max\{K(r_1 r_2 \dots r_n | r_e^*), K(r_e^* | r_1 r_2 \dots r_n)\}}{\max\{K(r_1 r_2 \dots r_n), K(r_e^*)\}}$$

As in the case of concatenating two representations, the miscoding of a joint representation composed of  $\$n\$$  elements remains bounded within the unit interval  $0 \leq \mu(r_1 r_2 \dots r_n) \leq 1$ . However, several caveats must be noted. There is no guarantee that  $\mu(r_1 r_2 \dots r_n) \leq \mu(r_i)$  or  $\mu(r_1 r_2 \dots r_n) \geq \mu(r_i)$  for any  $i = 1, \dots, n$ . Moreover, the miscoding of the joint representation is not necessarily less than or equal to the sum of the individual miscodings  $\mu(r_1 r_2 \dots r_n) \not\leq \mu(r_1) + \mu(r_2) + \dots + \mu(r_n)$ . Finally, no general conclusions can be drawn regarding how the miscoding of a joint representation is affected by permutations of its components. That is, reordering the representations may increase, decrease, or leave unchanged the overall miscoding.

### 3.3 Decreasing Miscoding

A valid representation of an entity is a string that contains all the essential information required by the oracle to reproduce the entity, no more, no less. In contrast, an invalid representation, indicated by a miscoding value greater than zero, may result from missing critical information, the inclusion of incorrect symbols, or the presence of irrelevant symbols. To reduce the miscoding of a representation, one may attempt to supply the missing information or remove erroneous or non-pertinent symbols. However, it is generally not possible to determine in advance which of these actions will be effective. Nevertheless, as the following theorem shows, at least one of these approaches must be applicable.

**Theorem 3.3.1** Let  $r \in \mathcal{B}^*$  be a representation such that  $\mu(r) > 0$ , then one or both of the following conditions must hold:

- (i) There exists a  $s \in \mathcal{B}^*$  such that  $\mu(rs) < \mu(r)$  or  $\mu(sr) < \mu(r)$ ,
- (ii) There exists an  $s \in \mathcal{B}^*$  in the form  $r = \alpha s \beta$  with  $\alpha, \beta \in \mathcal{B}^*$  such that

$$\mu(s) < \mu(r).$$

*Proof.* Let  $r \in \mathcal{B}^*$  be a representation with

$$\mu(r) = \min_{r_e^* \in \mathcal{R}_e^*} \frac{\max\{K(r \mid r_e^*), K(r_e^* \mid r)\}}{\max\{K(r), K(r_e^*)\}} > 0.$$

Fix one valid representation  $r^* \in \mathcal{R}_e^*$  at which the minimum is attained, and write

$$A := K(r \mid r^*), \quad B := K(r^* \mid r), \quad L := \max\{K(r), K(r^*)\},$$

so that  $\mu(r) = \frac{\max\{A, B\}}{L}$ .

We distinguish two (exhaustive) cases. Case i).  $B > 0$ , that is, some information is missing in  $r$ . Because  $B > 0$ , a shortest self-delimiting program  $p$  of length  $B$  exists that, given  $r$ , outputs  $r^*$ . Set  $s := p$  and consider the concatenation  $r' := rs$ . Because  $s$  already is the program that converts  $r$  into  $r^*$ ,

$$K(r^* \mid r') = O(1), \quad K(r' \mid r^*) \leq A + B + O(1).$$

Hence

$$\max\{K(r^* \mid r'), K(r' \mid r^*)\} \leq A + B + O(1) < A + L.$$

The description length of  $r'$  satisfies  $K(r') \leq K(r) + B + O(\log B)$ , so

$$\max\{K(r'), K(r^*)\} \geq \max\{K(r), K(r^*)\} = L.$$

Thus

$$\mu(r') = \frac{\max\{K(r^* \mid r'), K(r' \mid r^*)\}}{\max\{K(r'), K(r^*)\}} < \frac{A + B + O(1)}{L} \leq \frac{\max\{A, B\}}{L} = \mu(r),$$

so either  $\mu(rs) < \mu(r)$  (if we append) or, by symmetric reasoning,  $\mu(sr) < \mu(r)$  (if we prepend). Hence condition (i) is satisfied.

Case ii).  $B = 0$ , that is, all information needed to produce  $r^*$  is contained in  $r$ ; extra symbols remain. Now  $A > 0$  because  $\mu(r) > 0$ . Since  $K(r^* \mid r) = 0$ , a deterministic algorithm can scan  $r$  from left to right and halt as soon as it has seen a \*\*shortest prefix\*\* that already allows it to output  $r^*$ . Let  $s$  be that prefix and write  $r = \alpha s \beta$  with  $\beta \neq \lambda$  (otherwise  $r = s$  and  $A = 0$ , contradicting  $\mu(r) > 0$ ).

$$K(r^* \mid s) = 0, \quad K(s \mid r^*) \leq K(r \mid r^*) = A.$$

Because  $s$  is a proper prefix of  $r$ ,  $K(s) \leq K(r)$  (up to an  $O(\log K(r))$  term). Therefore

$$\mu(s) = \frac{\max\{K(s \mid r^*), K(r^* \mid s)\}}{\max\{K(s), K(r^*)\}} \leq \frac{A}{\max\{K(s), K(r^*)\}} < \frac{\max\{A, B\}}{L} = \mu(r),$$

so condition (ii) holds with that substring  $s$ . ■

Refer to Example ?? for a case in which appending additional information to a representation improves its quality, that is, reduces its miscoding. Also, the following example illustrates the opposite situation: a case in which removing certain symbols from a representation can lead to a decrease in miscoding.

■ **Example 3.4** A historian compiles a representation  $r$  intended to narrate the events surrounding the signing of a pivotal treaty. However, the inclusion of speculative statements and personal interpretations about the intentions of the involved parties, statements not supported by primary sources, results in a high miscoding value. Although the document is rich in content, it is contaminated by erroneous or unverifiable information, rendering it an invalid representation of the historical event.

By removing these speculative and unsubstantiated segments, a revised version  $r'$  is produced. This refined representation presents a concise and factual account based solely on verifiable data and primary sources. The removal of misleading content reduces the miscoding value, making  $r'$  a more valid representation that more accurately reflects the historical event without the noise of uncorroborated interpretations. ■

## 3.4 Targetless Representations

In the most extreme scenario, the miscoding value can reach its upper bound of 1. This occurs when the current representation  $r$  does not contain a single symbol that contributes to the encoding of any entity in the set  $\mathcal{E}$ . In such a case,  $r$  is said to be a targetless representation, an abstract string with no concrete or meaningful interpretation.

**Definition 3.4.1** Let  $r \in \mathcal{B}^*$  be a representation. If  $\mu(r) = 1$  we say that  $r$  is a *targetless* representation.

Although one could, in principle, assign a targetless representation to an arbitrary entity, doing so would violate the essential requirement of surrogate reasoning (see Section G.4). The following proposition formalizes this idea.

**Proposition 3.4.1** Given a representation  $r \in \mathcal{B}^*$ , the miscoding  $\mu(r)$  reaches its maximum value of 1 if and only if no symbol in  $r$  contributes to the encoding of any entity in  $\mathcal{E}$ .

*Proof.* Assume that  $r$  contains no symbol that contributes to the encoding of any entity  $e \in \mathcal{E}$ . Let  $r_e^*$  be a valid representation of some  $e \in \mathcal{E}$ . Then, the shortest program capable of generating  $r_e^*$  from  $r$  must encode the entirety of  $r_e^*$  without reuse of any symbols from  $r$ . Thus,  $K(r_e^* | r) = K(r_e^*)$ . Similarly, since no symbol in  $r_e^*$  contributes to the generation of  $r$ , we have  $K(r | r_e^*) = K(r)$ . Substituting into the definition of miscoding:

$$\mu(r) = \frac{\max\{K(r | r_e^*), K(r_e^* | r)\}}{\max\{K(r), K(r_e^*)\}} = \frac{\max\{K(r), K(r_e^*)\}}{\max\{K(r), K(r_e^*)\}} = 1$$

assuming  $K(r) \neq 0$  and  $K(r_e^*) \neq 0$ . Conversely, suppose that  $\mu(r) = 1$ . Then:

$$\max\{K(r | r_e^*), K(r_e^* | r)\} = \max\{K(r), K(r_e^*)\}$$

which implies that neither  $r$  can help generate  $r_e^*$  nor  $r_e^*$  can help generate  $r$ . Therefore, there is no overlap in information, and  $r$  contains no symbol that contributes to the encoding of any entity in  $\mathcal{E}$ .  $\blacksquare$

For every finite or countably infinite set of entities  $\mathcal{E}$ , there exists an infinite number of targetless representations. We will prove this result constructively by showing that at least one targetless representation exists and then defining a method to generate infinitely many more from it.

**Proposition 3.4.2** For every finite or countably infinite set of entities  $\mathcal{E}$ , there exists an uncountably infinite set of targetless representations.

*Proof.* Let  $\mathcal{R}$  denote the set of all valid representations of entities in  $\mathcal{E}$ , such that for each entity  $e \in \mathcal{E}$ , there exists a representation  $r \in \mathcal{R}$  and vice versa. Since  $\mathcal{E}$  is finite or countably infinite,  $\mathcal{R}$  is also countable.

Now consider the set  $\mathcal{B}^*$  of all finite binary strings. This set is countably infinite, as it includes all possible finite sequences over the alphabet 0, 1. Because  $\mathcal{R}$  is a subset of  $\mathcal{B}^*$  and is countable, and  $\mathcal{B}^* \setminus \mathcal{R}$  is non-empty and infinite, there must exist binary strings that are not valid representations of any entity in  $\mathcal{E}$ .

Let  $r_0 \in \mathcal{B}^* \setminus \mathcal{R}$  be any such string. By definition,  $r_0$  is a targetless representation.

Given any targetless representation  $r_i$ , we can construct a new targetless representation  $r_{i+1}$  by appending a symbol (either '0' or '1') to  $r_i$ . Since  $r_i$  does not encode any entity in  $\mathcal{E}$ , and  $\mathcal{R}$  is prefix-free with respect to targetless extensions (i.e., extending a non-representation cannot turn it into a valid

one), the new string  $r_{i+1}$  also does not correspond to any entity. Therefore,  $r_{i+1}$  is also targetless.

By iterating this process, we generate an infinite sequence of distinct targetless representations  $r_0, r_1, r_2, \dots \subset \mathcal{B}^* \setminus \mathcal{R}$ . This proves that there exists a countably infinite set of targetless representations corresponding to any finite or countably infinite set of entities  $\mathcal{E}$ . ■

What constitutes a targetless representation for one oracle may not necessarily be targetless for another, as the following example illustrates.

■ **Example 3.5** Imagine two machines, A and B, each controlling a robotic arm capable of manufacturing nuts and bolts. Machine A operates using a low-level assembly language, whereas Machine B uses a more sophisticated high-level programming language. As a result, a particular set of instructions that fails to produce a valid output on Machine A—thus being considered a targetless representation—could be perfectly interpretable by Machine B, successfully yielding a finished bolt. ■

The normalized compression distance between a targetless representation and the closest valid (non-targetless) representation may be less than one. This indicates that the targetless representation shares some information with a valid one. However, this is not sufficient to ensure scientific progress, as the computational effort, reflected in the size or complexity of the Turing machine required to extract useful knowledge, may be larger.

## 3.5 Miscoding of Areas

The concept of miscoding can be extended to research areas to quantitatively measure the amount of effort required to correct an inaccurate representation of an area. Unfortunately, as discussed in Section 2.8, there is no reliable way to verify whether the strings included in an  $n$ -fold representation actually correspond to entities within that area. Moreover, it is not possible to prevent cases in which some of these strings represent the same entity.

Consider the strings  $r_1, r_2, \dots, r_n$ , where each  $r_i \in \mathcal{B}^*$  for  $i = 1, 2, \dots, n$ . Recall that the expression  $r_1 r_2 \dots r_n$  refers to the concatenation of these strings into a single binary string. This operation may merge the individual components in a way that makes them inseparable. In contrast, the notation  $\langle r_1, r_2, \dots, r_n \rangle$  denotes a re-encoding of the individual strings into a single string that preserves the ability to recover each original component.

Furthermore, the joint representation  $r_1 r_2 \dots r_n$  is assumed to encode a single entity, while the  $n$ -fold representation  $\langle r_1, r_2, \dots, r_n \rangle$  may encode up to  $n$  distinct entities.

The following definition extends the concept of miscoding to  $n$ -fold representations.

**Definition 3.5.1** Let  $R = (r_1, r_2, \dots, r_n) \in \mathcal{B}^* \times \mathcal{B}^* \times \dots \times \mathcal{B}^*$  be an  $n$ -fold representation. We define the *miscoding* of  $R$ , denoted by  $\mu(R)$ , as:

$$\mu(R) = \min_{(r_{e_1}^*, \dots, r_{e_n}^*) \in \mathcal{R}_{\mathcal{E}}^* \times \dots \times \mathcal{R}_{\mathcal{E}}^*} \frac{\max \{ K(\langle r_1, \dots, r_n \rangle \mid \langle r_{e_1}^*, \dots, r_{e_n}^* \rangle), K(\langle r_{e_1}^*, \dots, r_{e_n}^* \rangle \mid \langle r_1, \dots, r_n \rangle) \}}{\max \{ K(\langle r_1, \dots, r_n \rangle), K(\langle r_{e_1}^*, \dots, r_{e_n}^* \rangle) \}}$$

where  $\min^o$  indicates that the minimum is to be computed by an oracle.

The miscoding of the representation of an area falls within the range  $[0, 1]$ , as demonstrated by the following proposition.

**Proposition 3.5.1** For all known subsets  $R = (r_1, r_2, \dots, r_n) \in \mathcal{B}^* \times \mathcal{B}^* \times \dots \times \mathcal{B}^*$ , it holds that  $0 \leq \mu(R) \leq 1$ .

*Proof.* Since  $\langle r_1, r_2, \dots, r_n \rangle$  is a string in  $\mathcal{B}^*$ , we can apply Proposition E.5.5, which guarantees that the normalized compression-based distance lies between 0 and 1. ■

By extending the concept of miscoding to cover research areas, we gain a quantitative means of evaluating the quality of representations for specific subsets of entities. This mathematical framework provides a rigorous tool to assess and correct inaccuracies, both at the level of individual entities and across broader scientific domains.

## References

Misrepresentation or inaccuracies in scientific representation have significant implications for scientific discovery, technological progress, policymaking, and other domains. However, no book or paper explicitly addresses the topic of "incorrect representations in science" from the perspective adopted in this work. Here, we decompose the problem of scientific representation into two complementary subproblems: the representation of entities and the description of those representations.

[Sup02] provides a formal foundation for scientific representation and emphasizes the role of structure-preserving mappings; his analysis is crucial for understanding when and how representations fail to capture relevant features of phenomena. [Van80] introduces the concept of "constructive empiricism" and highlights the model-dependent nature of scientific representation.

[LSW13] offers an ethnographic study of how scientific representations are socially constructed—and, at times, distorted—in scientific practice.

Although the specific topic of incorrect representations has not been directly examined in this way, various researchers have addressed related concerns indirectly. Their discussions often focus on issues such as scientific fraud, the replication crisis, and the use of incorrect or misleading models. For example, [Ioa05] presents a widely cited empirical and philosophical analysis of how methodological biases, poor model design, and selective reporting contribute to unreliable or misleading scientific results.





## 4. Inaccuracy

*A little inaccuracy sometimes saves tons of explanations.*

Saki

In Section 2.5, we introduced the notion of a description, or model, of an entity as a computer program. When executed, this program reproduces one of the representations encoding the entity in question. More precisely, a description  $d$  for a representation  $r$  of an entity  $e$  is a Turing machine that produces the string  $r$  when interpreted by a universal Turing machine  $\delta$ . However, due to our typically incomplete understanding of the entity  $e$ , the actual output of the description, denoted as  $r' = \delta(d)$ , will generally resemble but not exactly match  $r$ .

In this chapter, we investigate the error introduced by flawed models, specifically, how closely the output  $r'$  approximates the intended representation  $r$ . We refer to this type of error as the *inaccuracy* of the description  $d$ .

Inaccuracy constitutes the second metric for assessing our understanding of a research entity. The underlying idea is that the more accurate our model, the closer  $r'$  is to  $r$ , and thus the better our understanding of the entity. Formally, the inaccuracy of a description  $d$  is defined as the normalized

information distance between the original representation  $r$  and the output representation  $r'$  produced by the description. That is, inaccuracy measures the length of the shortest computer program needed to transform the erroneous output  $r'$  into the correct representation  $r$ .

Inaccuracy evaluates how well the output of a description aligns with the selected representation encoding the entity. However, as discussed in the preceding chapter, the representation itself may be flawed. Inaccuracy focuses exclusively on the description  $d$ , without accounting for potential miscoding in the representation  $r$ . Moreover, although its computation does not require an oracle, inaccuracy cannot always be calculated exactly in practice; instead, it must often be estimated—a topic we will address in Part II of this book.

In this chapter, we formally define inaccuracy and examine its key properties. We also analyze how inaccuracy behaves when conditional descriptions are used in place of unconditional ones. Finally, we extend the notion of inaccuracy from individual entities to entire research areas.

This investigation is not purely theoretical, it has significant practical relevance. Accurate models are essential across a wide range of domains, from climate prediction to the development of artificial intelligence systems. Understanding and quantifying inaccuracy can thus lead to better models, ultimately improving our ability to make reliable predictions and informed decisions.

## 4.1 Inaccuracy

In the process of studying an entity  $e \in \mathcal{E}$  through a representation  $r \in \mathcal{R}_e$ , we may encounter situations in which our proposed description  $d$  fails to accurately produce  $r$ . That is,  $d \notin \mathcal{D}_r$  (see Definition 2.5.2). In such cases, when the universal Turing machine  $\delta$  receives  $d$  as input, it produces a string  $r'$  that differs from the original representation  $r$ .

Intuitively, one might say that  $d$  is an inaccurate description of the entity  $e$ . However, because descriptions refer to entities only indirectly, via representations, our formal notion of inaccuracy must be defined in terms of the representation, not the entity itself. Furthermore, we must account for the possibility that the representation  $r$  is itself flawed, as previously discussed through the concept of miscoding.

With these considerations in mind, we introduce the following definition of an inaccurate description.

**Definition 4.1.1** Let  $r \in \mathcal{B}^*$  be a representation, and let  $d \in \mathcal{D}$  be a description, where  $d = \langle TM, a \rangle$ . If the output of  $TM(a)$  is a string  $r'$  such that  $r \neq r'$ , we say that  $d$  is an *inaccurate* description for  $r$ .

Our proposed description  $d$  may fall outside the set of valid descriptions  $\mathcal{D}_r$  for  $r$  (indicating positive inaccuracy), and the representation  $r$  may not belong to the set of valid representations  $\mathcal{R}_e^*$  for the entity  $e$  (indicating positive miscoding).

When a description is inaccurate, we aim to quantify the degree of inaccuracy. Within the computational framework, a natural approach is to measure how difficult it is to transform the incorrect representation  $r'$ , obtained by executing  $d$  on the universal Turing machine, into the original representation  $r$ . This difficulty is captured by the normalized information distance between  $r'$  and  $r$ .

**Definition 4.1.2 — Inaccuracy.** Let  $r \in \mathcal{B}^*$  be a representation, and let  $d \in \mathcal{D}$  be a description, where  $d = \langle TM, a \rangle$ . We define the *inaccuracy* of the description  $d$  with respect to the representation  $r$ , denoted by  $\iota(d, r)$ , as:

$$\iota(d, r) = \frac{\max \{K(r \mid \delta(d)), K(\delta(d) \mid r)\}}{\max \{K(r), K(\delta(d))\}}$$

The use of a relative measure of inaccuracy, rather than an absolute one, enables meaningful comparisons between inaccuracies across different descriptions of the same representation, as well as between descriptions of different representations.

Similar to miscoding (see Definition 3.1.1), inaccuracy is computed using a bidirectional approach: we calculate the length of the shortest computer program that can generate the correct representation  $r$  from the erroneous one  $r'$ , and vice versa, that is, the shortest program that can generate  $r'$  from  $r$ . In essence, a valid description should produce a representation that contains all the necessary information to reconstruct the intended entity, while excluding any erroneous or irrelevant content.

■ **Example 4.1** Inaccuracy primarily concerns the difficulty of correcting the *output* of a description, that is, the result produced by a computable model, rather than the difficulty of modifying the description itself.

For example, suppose we have a dataset generated by a system that is perfectly described by a quadratic function, but we choose a linear function as our model. In this case, inaccuracy evaluates how different the predicted dataset (produced by the linear model) is from the original quadratic dataset. It does not measure how difficult it is to transform the incorrect linear model into the correct quadratic one.

In this sense, if the original dataset consists of 10 points, a polynomial of degree ten that perfectly fits the data would also yield an inaccuracy of zero. Determining which model is preferable, the quadratic model with

zero inaccuracy or the more complex degree-ten polynomial, also with zero inaccuracy, is a matter addressed by the surfeit metric (see Chapter 5). ■

Given its basis in Kolmogorov complexity, inaccuracy is a quantity that, in general, cannot be computed exactly and must instead be approximated. The method used to approximate inaccuracy depends on the specific characteristics of the entities under study and the nature of their representations.

Conveniently, the inaccuracy of a description always falls within the range  $[0, 1]$ , as established by the following proposition.

**Proposition 4.1.1** For all representations  $r \in \mathcal{B}^*$  and all descriptions  $d \in \mathcal{D}$ , it holds that  $0 \leq \iota(d, r) \leq 1$ .

*Proof.* This follows from the general inequality

$$0 \leq \frac{\max\{K(x | y), K(y | x)\}}{\max\{K(x), K(y)\}} \leq 1$$

for all  $x, y \in \mathcal{B}^*$ , as stated in Proposition E.5.5. ■

The proposition above applies to all pairs of descriptions  $d$  and representations  $r$ , even in cases where  $d$  is not intended to model  $r$ . In such instances, the inaccuracy  $\iota(d, r)$  will typically be close to one.

Inaccuracy is exactly zero if and only if the description  $d$  is one of the valid descriptions of the representation  $r$ .

**Proposition 4.1.2** Given a description  $d \in \mathcal{D}$  and a representation  $r \in \mathcal{B}^*$ , we have that  $\iota(d, r) = 0$  if and only if  $d \in \mathcal{D}_r$ , i.e.,  $d$  is a valid description of  $r$ .

*Proof.* If  $d \in \mathcal{D}_r$ , then by definition,  $\delta(d) = r$ . Consequently, we have  $K(r | \delta(d)) = K(\delta(d) | r) = 0$ , and thus  $\iota(d, r) = 0$ .

Conversely, suppose  $\iota(d, r) = 0$ . Then,

$$\max\{K(r | \delta(d)), K(\delta(d) | r)\} = 0,$$

which implies that both conditional complexities are zero. Therefore,  $r = \delta(d)$ , and it follows that  $d \in \mathcal{D}_r$ . ■

Finally, given two representations  $r$  and  $s$ , we may also be interested in evaluating the inaccuracy of a description  $d$  when it is used to describe their joint representation  $rs$ . Since we require that  $rs$  is itself a valid representation, the extension of the inaccuracy concept to joint representations is straightforward and does not require a new definition:

$$\iota(d, rs) = \frac{\max\{K(rs | \delta(d)), K(\delta(d) | rs)\}}{\max\{K(rs), K(\delta(d))\}}$$

As a direct consequence of Proposition 4.5.1, for any representations  $r, s \in \mathcal{B}^*$  and any description  $d \in \mathcal{D}$ , we have:

$$0 \leq \iota(d, rs) \leq 1.$$

## 4.2 Conditional Inaccuracy

In this section, we delve deeper into the concept of inaccuracy by considering its application to conditional descriptions. Specifically, we explore the inaccuracy of a description when evaluated in conjunction with pre-existing background knowledge, a notion we term *conditional inaccuracy*. As we will see, the inaccuracy of a conditional description can never exceed that of its unconditional counterpart; at worst, it remains unchanged. This property makes conditional inaccuracy a useful tool for evaluating new concepts or models and for assessing their explanatory power with respect to the entity of interest.

In Definition 2.7.1, we introduced the concept of a conditional description  $d$  for a representation  $r$ , given an arbitrary background string  $s$ . This is denoted by  $d | s$  and defined as the self-delimiting concatenated string  $\langle d, s \rangle$ , where  $d = \langle TM, a \rangle$  and  $TM(\langle a, s \rangle) = r$ . If  $TM(\langle a, s \rangle) = r'$  with  $r \neq r'$ , then  $d | s$  is referred to as an *inaccurate* conditional description of  $r$ .

It is important to note that  $d | s$  must be defined for all possible background strings  $s$ . Additionally, we refer to the case in which  $s$  is the empty string, denoted  $d | \lambda$ , as the *unconditional* version of the description.

Building on the concept of inaccuracy introduced in Definition 4.1.2, we now define the notion of *conditional inaccuracy* to capture the error introduced when using an inaccurate conditional description.

**Definition 4.2.1** Let  $r \in \mathcal{B}^*$  be a representation,  $s \in \mathcal{B}^*$  a background string, and  $d | s$  an inaccurate conditional description. We define the *conditional inaccuracy* of the description  $d$  for the representation  $r$  given the string  $s$ , denoted by  $\iota(d | s, r)$ , as:

$$\iota(d | s, r) = \frac{\max \{K(r | \delta(d | s)), K(\delta(d | s) | r)\}}{\max \{K(r), K(\delta(d | s))\}}$$

Conditional inaccuracy is thus defined as the normalized information distance between the original representation  $r$  and the string produced by the conditional description  $d | s$ .

As a normalized measure, the conditional inaccuracy of a description lies within the interval  $[0, 1]$ .

**Proposition 4.2.1** Let  $r \in \mathcal{B}^*$  be a representation,  $s \in \mathcal{B}^*$  a string, and  $d | s$  a conditional description of  $r$  given  $s$ . Then  $0 \leq \iota(d | s, r) \leq 1$ .

*Proof.* This follows directly from the general inequality:

$$0 \leq \frac{\max\{K(x | y), K(y | x)\}}{\max\{K(x), K(y)\}} \leq 1$$

for all  $x, y \in \mathcal{B}^*$ , as established in Proposition E.5.5. ■

The conditional inaccuracy assumes a value of zero if and only if the conditional description  $d | s$  is a valid model that correctly produces the representation  $r$ .

**Proposition 4.2.2** Let  $r \in \mathcal{B}^*$  be a representation,  $s \in \mathcal{B}^*$  a string, and  $d | s$  a conditional description of  $r$  given  $s$ , where  $d = \langle TM, a \rangle$ . Then  $\iota(d | s, r) = 0$  if and only if  $TM(\langle a, s \rangle) = r$ .

*Proof.* If  $TM(\langle a, s \rangle) = r$ , then  $\delta(d | s) = r$ , which implies that

$$K(r | \delta(d | s)) = K(\delta(d | s) | r) = 0.$$

Hence,  $\iota(d | s, r) = 0$ . Conversely, if  $\iota(d | s, r) = 0$ , then

$$\max\{K(r | \delta(d | s)), K(\delta(d | s) | r)\} = 0,$$

which implies both conditional complexities are zero. Therefore,  $\delta(d | s) = r$ , which means  $TM(\langle a, s \rangle) = r$ . ■

Incorporating established prior knowledge into research does not increase the inaccuracy of a description. If this background knowledge is relevant to the representation being described, the oracle will use it accordingly. Conversely, if the prior knowledge is irrelevant, the oracle will simply disregard it. The following theorem formalizes this idea.

**Theorem 4.2.3** Let  $r \in \mathcal{B}^*$  be a representation, and  $d \in \mathcal{D}$  a conditional description of  $r$ . Then

$$\iota(d | s, r) \leq \iota(d, r)$$

for all strings  $s \in \mathcal{B}^*$ .

*Proof.* Since  $\iota(d, r)$  is equivalent to  $\iota(d | \lambda, r)$ , we need to prove that

$$\frac{\max\{K(r | \delta(d | s)), K(\delta(d | s) | r)\}}{\max\{K(r), K(\delta(d | s))\}} \leq \frac{\max\{K(r | \delta(d | \lambda)), K(\delta(d | \lambda) | r)\}}{\max\{K(r), K(\delta(d | \lambda))\}}.$$

This inequality follows from the fact that

$$K(r | \langle \delta(d), s \rangle) \leq K(r | \delta(d)),$$

as shown in Proposition E.4.3. ■

Theorem 4.2.3 represents a foundational result in the theory of nescience. It establishes the basis for developing a robust methodology aimed at deepening our understanding (i.e., reducing inaccuracy) of a research entity. In practical applications, our primary focus will typically be on prior knowledge directly related to the subject of study. However, the core insight of Theorem 4.2.3 is that incorporating concepts from seemingly unrelated domains will not compromise the accuracy of our investigation. This theorem becomes particularly powerful when such exploratory processes are automated (see Chapter 10).

■ **Example 4.2** The P vs. NP problem stands as one of the most significant unresolved questions in computer science. It asks whether every problem whose solution can be verified in polynomial time (class NP) can also be solved in polynomial time (class P). The relationship between these two complexity classes remains unsolved. Constructing a comprehensive, self-contained solution to this problem in a formal language is an immense challenge. However, leveraging relevant prior knowledge can significantly reduce the complexity of the required description. For instance, insights from Algorithm Theory, which deals with the classification and efficiency of algorithms, and from Formal Language Theory, which addresses the structure of computational problems (e.g., regular and context-free languages) and highlights the role of Turing machines, can be instrumental. Drawing upon such established knowledge may not only simplify our descriptions but also facilitate a deeper understanding and potentially contribute to resolving the P vs. NP problem. ■

Finally, given two representations  $r$  and  $t$ , the formalization of the concept of conditional inaccuracy, when applied to the joint representation  $rt$ , is straightforward and does not require a new definition:

$$\iota(d | s) = \frac{\max\{K(rt | \delta(d | s)), K(\delta(d | s) | rt)\}}{\max\{K(rt), K(\delta(d | s))\}}$$

As a normalized measure,  $\iota(d | s)$  always takes a value in the interval  $[0, 1]$ .

## 4.3 Decreasing Inaccuracy

Our objective is to reduce the inaccuracy of the current description  $d_1$ , thereby improving our understanding of the original entity. This improvement may involve either modifying  $d_1$  to correct or eliminate its inaccuracies, or developing a completely new description based on a different approach to modeling the entity. In either case, the result is a new description  $d_2$ . In this section, we aim to analyze how the introduction of a new description  $d_2$  affects the inaccuracy compared to the original description  $d_1$ .

**Definition 4.3.1** Let  $r \in \mathcal{B}^*$  be a representation, and let  $d_1, d_2 \in \mathcal{D}$  be two descriptions. We define the *variation of inaccuracy* between the descriptions  $d_1$  and  $d_2$ , with respect to  $r$ , denoted by  $\Delta_t^a(d_1, d_2, r)$ , as:

$$\Delta_t^a(d_1, d_2, r) = \iota(d_1, r) - \iota(d_2, r)$$

Since inaccuracy is bounded between 0 and 1, the maximum possible variation in inaccuracy is  $\pm 1$ . A positive value of  $\Delta_t$  indicates that description  $d_2$  is preferable to  $d_1$  in terms of accuracy. Conversely, a negative value suggests that  $d_1$  is more accurate than  $d_2$ . It is important to note that the new description may also introduce a substantial increase in surfeit, potentially outweighing the improvement in inaccuracy. For a detailed discussion of surfeit, refer to Chapter 5, and for an explanation of how inaccuracy and surfeit combine into the unified metric of nescience, see Chapter 6.

We can also introduce a relative measure of the variation in inaccuracy, when moving from description  $d_1$  to  $d_2$

**Definition 4.3.2** Let  $r \in \mathcal{B}^*$  be a representation, and  $d_1, d_2 \in \mathcal{D}$  be two descriptions. We define the *relative variation of inaccuracy* between descriptions  $d_1$  and  $d_2$ , denoted by  $\Delta_t^r(d_1, d_2, r)$ , as:

$$\Delta_t^r(d_1, d_2, r) = \frac{\iota(d_1, r) - \iota(d_2, r)}{\iota(d_1, r)}$$

provided that  $\iota(d_1, r) \neq 0$ .

A value of 0 indicates no change, while a value of 1 corresponds to a complete elimination of inaccuracy. Negative values, on the other hand, signify an increase in inaccuracy, indicating that the new description  $d_2$  is less accurate than the original  $d_1$ . Note that the relative variation can be arbitrarily negative, diverging to  $-\infty$  as  $\iota(d_2, r)$  increases and  $\iota(d_1, r)$  approaches zero.

As inaccuracy approaches zero, relative variations become increasingly unstable. A small absolute change can produce a large relative variation if the initial inaccuracy is very low. For instance, if  $\iota(d_1, r) = 0.1$  and the inaccuracy decreases by 0.05, this corresponds to a relative improvement of 50%. In contrast, if  $\iota(d_1, r) = 0.9$  and the same absolute improvement occurs, the relative reduction is only about 5.6%. Both absolute and relative variations are essential for evaluating the significance and magnitude of improvements.

An alternative strategy for reducing uncertainty about an entity involves modifying its representation rather than altering its description. While such a change might increase the miscoding of the entity, the potential reduction in inaccuracy could outweigh this drawback.

**Definition 4.3.3** Let  $r_1, r_2 \in \mathcal{B}^*$  be two representations, and let  $d \in \mathcal{D}$  be a description. We define the *variation of inaccuracy* between representations  $r_1$  and  $r_2$ , denoted by  $\Delta_t^a(d, r_1, r_2)$ , as:

$$\Delta_t^a(d, r_1, r_2) = \iota(d, r_1) - \iota(d, r_2)$$

As before, since inaccuracy is bounded between 0 and 1, the maximum possible variation is 1. A positive value of  $\Delta_t^a$  indicates a preference for representation  $r_2$  over  $r_1$  with respect to the fixed description  $d$ . Conversely, a negative value suggests that  $r_1$  is more accurate than  $r_2$ . This comparison is based solely on inaccuracy, as miscoding is not considered here. Additionally, since the description remains unchanged, there is no risk of variation in surfeit.

Finally, we can also introduce a relative variation of inaccuracy with respect to a change in representation

**Definition 4.3.4** Let  $r_1, r_2 \in \mathcal{B}^*$  be two representations, and  $d \in \mathcal{D}$  be a description. We define the *relative variation of inaccuracy* of the representations  $r_1, r_2$ , denoted by  $\Delta_t^r(d, r_1, r_2)$ , as:

$$\Delta_t^r(d, r_1, r_2) = \frac{\iota(d, r_1) - \iota(d, r_2)}{\iota(d, r_1)}$$

provided that  $\iota(d, r_1) \neq 0$ .

This quantity measures the proportional reduction in inaccuracy resulting from replacing representation  $r_1$  with  $r_2$ , while keeping the description fixed. A value of 0 indicates no change in inaccuracy, and a value of 1 corresponds to a complete elimination of inaccuracy. However,  $\Delta_t^r$  can also take negative values, potentially diverging to  $-\infty$ , when the new representation  $r_2$  performs worse than  $r_1$ . As the inaccuracy of the initial representation  $r_1$  approaches zero, even minor absolute changes in inaccuracy can lead to large swings in the relative variation, making it increasingly volatile.

## 4.4 Inaccuracy-Miscoding Rate of Change

In the preceding section, we examined how the inaccuracy of a representation can be reduced by selecting a different description. We also explored an alternative strategy in which inaccuracy is minimized not by modifying the description, but by changing the representation itself. In this section, we turn our attention to a more general approach for reducing the inaccuracy associated with an entity. Rather than altering the description or the representation in isolation, it may be more effective to modify both simultaneously.

The balance between the amount of miscoding we are willing to accept in order to achieve a reduction in inaccuracy is referred to as the *miscoding-inaccuracy trade-off*. For a broader discussion of trade-offs in multi-objective optimization, refer to Section F.6.2.

**Definition 4.4.1** Let  $e \in \mathcal{E}$  be an entity, and  $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{R}_e \times \mathcal{D}$  be two hypotheses, with  $\mathbf{x}_1 = (r_1, d_1)$  and  $\mathbf{x}_2 = (r_2, d_2)$ . We define the *rate of change between the inaccuracy and the miscoding* of the hypothesis  $\mathbf{x}_1, \mathbf{x}_2$ , denoted by  $\Delta_{l\mu}(\mathbf{x}_1, \mathbf{x}_2)$  as:

$$\Delta_{l\mu}(\mathbf{x}_1, \mathbf{x}_2) = \frac{\iota(d_2, r_2) - \iota(d_1, r_1)}{\mu(r_2) - \mu(r_1)}$$

provided that  $\mu(r_2) - \mu(r_1) \neq 0$ .

The ratio  $\Delta_{l\mu}$  represents the rate of change between inaccuracy and miscoding when transitioning from the first hypothesis to the second. A positive value of  $\Delta_{l\mu}$  implies that both quantities, miscoding and inaccuracy, either decrease (which is desirable) or increase (which is undesirable). The interpretation becomes more nuanced when  $\Delta_{l\mu}$  is negative, indicating that one of the quantities decreases while the other increases. In such cases, two scenarios must be considered:

- (i) Inaccuracy decreases and miscoding increases, we aim for  $\Delta_{l\mu}(\mathbf{x}_1, \mathbf{x}_2) < M$ , where  $M < -1$ , thereby ensuring that the reduction in inaccuracy compensates for the increase in miscoding.
- (ii) Inaccuracy increases and miscoding decreases, we aim for  $\Delta_{l\mu}(\mathbf{x}_1, \mathbf{x}_2) > M$ , where  $-1 < M < 0$ , thereby ensuring that the reduction in miscoding justifies the increase in inaccuracy.

In both cases, caution is warranted when the change in miscoding is small, as it can disproportionately affect the ratio and potentially lead to misleading conclusions.

■ **Example 4.3** Let  $\mathbf{x}_1$  and  $\mathbf{x}_2$  be two hypotheses. For  $\mathbf{x}_1$ , the inaccuracy  $\iota(d_1, r_1)$  is 0.40, and the miscoding  $\mu(r_1)$  is 0.15. For  $\mathbf{x}_2$ , the inaccuracy  $\iota(d_2, r_2)$  is 0.20 (a decrease from 0.40), and the miscoding  $\mu(r_2)$  is 0.25 (an increase from 0.15). Using the definition of the rate of change, we compute:

$$\Delta_{l\mu}(\mathbf{x}_1, \mathbf{x}_2) = \frac{0.40 - 0.20}{0.15 - 0.25} = \frac{0.20}{-0.10} = -2$$

In this case, transitioning from hypothesis  $\mathbf{x}_1$  to  $\mathbf{x}_2$  results in a decrease of inaccuracy by 0.20 units and an increase in miscoding by 0.10 units. The rate of change is  $-2$ . ■

Having a very small change in miscoding can significantly amplify the

value of the rate of change, potentially giving the misleading impression that inaccuracy and miscoding are varying at an extreme rate, even when the actual changes are minor. This phenomenon is illustrated in the following example.

■ **Example 4.4** Let  $\mathbf{x}_1$  be a hypothesis with an inaccuracy  $\iota(d_1, r_1)$  of 0.35 and a miscoding  $\mu(r_1)$  of 0.20. Let  $\mathbf{x}_2$  be a second hypothesis with an inaccuracy  $\iota(d_2, r_2)$  of 0.30 (a slight decrease from 0.35) and a miscoding  $\mu(r_2)$  of 0.2001 (a very small increase from 0.20). Applying the definition of the rate of change:

$$\Delta_{\iota\mu}(\mathbf{x}_1, \mathbf{x}_2) = \frac{0.35 - 0.30}{0.20 - 0.2001} = \frac{0.05}{-0.0001} = -500$$

The rate of change is  $-500$ , which may misleadingly suggest a dramatic shift. In reality, the inaccuracy only decreased by 0.05 units, and the miscoding increased by a negligible 0.0001 units. The extremely small denominator inflates the result, making the change appear far more significant than it actually is. ■

As we attempt to optimize both inaccuracy and miscoding, we inevitably encounter configurations (hypotheses) where improving one objective necessitates compromising the other. The set of such "best trade-off" configurations constitutes the Pareto frontier (see Section F.6). Points on the Pareto frontier are said to be Pareto optimal because any attempt to improve one objective leads to a deterioration in the other.

**Definition 4.4.2** Let  $e \in \mathcal{E}$  be an entity, and let  $\mathbf{x} = (r, d) \in \mathcal{R}_e \times \mathcal{D}$  be a hypothesis. The hypothesis  $\mathbf{x}$  is said to be a Pareto point with respect to inaccuracy  $\iota$  and miscoding  $\mu$  if there does not exist another hypothesis  $\mathbf{x}' = (r', d')$  such that:

- 1  $\iota(d', r') \leq \iota(d, r)$  and  $\mu(r') \leq \mu(r)$ , and
- 2  $\iota(d', r') < \iota(d, r)$  or  $\mu(r') < \mu(r)$ .

In simpler terms, a hypothesis  $\mathbf{x}$  is Pareto optimal if: (i) no other hypothesis is better in both inaccuracy and miscoding, and (ii) any improvement in one metric necessarily results in a worsening of the other.

The rate of change  $\Delta_{\iota\mu}$  between any two Pareto optimal points provides insight into how the trade-off between inaccuracy and miscoding evolves along the Pareto frontier. If decision-makers are more sensitive to changes in inaccuracy than to miscoding, they may prefer configurations with a less negative  $\Delta_{\iota\mu}$ . Conversely, if miscoding is of greater concern, they may accept solutions where  $\Delta_{\iota\mu}$  indicates a larger increase in inaccuracy in exchange for a smaller gain in miscoding.

## 4.5 Inaccuracy of Areas

An area  $\mathcal{A}$  (see Section 2.8) is a subset  $\mathcal{A} \subset \mathcal{E}$  of entities that are related or share a common property. The concept of inaccuracy can be extended to research areas in order to quantitatively measure the effort required to correct an inaccurate description of an area.

Given the strings  $r_1, r_2, \dots, r_n$ , where each  $r_i \in \mathcal{B}^*$  for  $i = 1, 2, \dots, n$ , recall that we use  $\langle r_1, r_2, \dots, r_n \rangle$  to denote a self-delimited encoding of the individual strings  $r_i$  into a unified string, such that the original components can be fully recovered.

The following definition extends the concept of inaccuracy to research areas.

**Definition 4.5.1** Let  $\mathcal{A} \subset \mathcal{E}$  be an area with a known subset  $\hat{\mathcal{A}} = r_1, r_2, \dots, r_n$ , and let  $d \in \mathcal{D}$  be a description. We define the *inaccuracy of the area* given the description  $d$ , denoted by  $\iota(d, \hat{\mathcal{A}})$ , as:

$$\iota(d, \hat{\mathcal{A}}) = \frac{\max\{K(\langle r_1, r_2, \dots, r_n \rangle \mid \delta(d)), K(\delta(d) \mid \langle r_1, r_2, \dots, r_n \rangle)\}}{\max\{K(\langle r_1, r_2, \dots, r_n \rangle), K(\delta(d))\}}$$

The inaccuracy of a description for an area falls within the interval from 0 to 1, as established by the following proposition.

**Proposition 4.5.1** For all known subsets  $\hat{\mathcal{A}} = r_1, r_2, \dots, r_n$  and all descriptions  $d \in \mathcal{D}$ , we have that  $0 \leq \iota(d, \hat{\mathcal{A}}) \leq 1$ .

*Proof.* The result follows directly from the fact that  $\langle r_1, r_2, \dots, r_n \rangle$  is a string in  $\mathcal{B}^*$ , and from Proposition E.5.5. ■

By extending the concept of inaccuracy to cover areas, we can quantitatively evaluate the quality of a description for a specific subset of entities. This mathematical framework provides a rigorous tool for assessing and correcting inaccuracies not only at the level of individual entities but also across broader research domains.

## References

A good introduction to the study of uncertainties (i.e., error analysis in models) in science (particularly in physics, chemistry, and engineering) is the best-selling textbook by Taylor [TAY22], which also features the same image of a crashed train used in the introduction to this chapter. Another excellent entry-level reference on error analysis, aimed at undergraduate students in science and technology, is the book by Hughes and Hase [HH10].

From a more philosophical perspective, Popper's work [Pop14] is highly influential. In it, he introduces the concept of falsifiability, asserting that for a theory to be regarded as scientific, it must be testable and subject to refutation.





## 5. Surfeit

*Everything should be made as simple as possible,  
but not simpler.*  
Albert Einstein

Surfeit is the final metric we introduce to quantitatively assess our understanding of a research entity. It measures the presence of superfluous symbols in the description used to model that entity. Intuitively, our lack of knowledge is often reflected in the length (i.e., number of symbols) of our current description. Lengthy descriptions tend to include erroneous or redundant elements. As our understanding of the subject improves, we should be able to identify and remove these unnecessary symbols, resulting in a more concise and accurate description.

We define the surfeit of a description for an entity as the difference in length between the given description and the optimal (i.e., shortest) one for that entity. Within the framework of the theory of nescience, we assume that the pinnacle of knowledge about an entity, its perfect description, is represented by the shortest description capable of fully reconstructing the entity's representation. This notion of perfection relies on both the validity of the representation and the accuracy of the description.

The length of the most concise description of an entity is determined by the Kolmogorov complexity of a representation of that entity. In practical scenarios, given that our knowledge of entities is typically incomplete, the most concise possible description remains unknown. Moreover, as previously discussed, Kolmogorov complexity is not computable in general. Consequently, surfeit is a metric that must be approximated in practice.

If we could construct a perfect description of an entity, it would necessarily be a random string; otherwise, it would contain redundant elements that could be eliminated. Within the framework of the theory of nescience, attaining perfect knowledge corresponds to reaching a state of randomness. This inherent randomness defines a boundary on the depth of understanding achievable for a given research topic. However, rather than constituting a limitation, recognizing and understanding this boundary opens new opportunities in both science and technology. For example, by assessing how far our current description deviates from a random string, we can estimate our proximity to realizing a perfect description.

In this chapter, we formally introduce the concept of surfeit and examine its properties, including conditional surfeit. We will also present the notion of redundancy as a practical approximation of surfeit. Strategies for reducing both surfeit and redundancy will be discussed, as well as the relationship between reductions in surfeit and changes in inaccuracy or miscoding. Finally, we extend the concept of surfeit to support the analysis of entire research areas.

## 5.1 Surfeit

Given the length of a description of a representation for an entity and the length of its shortest possible description, we can introduce a relative measure to quantify the unnecessary effort involved in explaining the entity using that particular description. We call this quantity *surfeit*. Surfeit is a key component of our definition of nescience, as it reflects the degree to which our current understanding of the research entity includes superfluous or redundant information.

**Definition 5.1.1 — Surfeit.** Given a representation  $r \in \mathcal{B}^*$ , and a description  $d \in \mathcal{D}$  for  $r$ , we define the *surfeit of the description  $d$  for the representation  $r$* , denoted by  $\sigma(d, r)$ , as

$$\sigma(d, r) = \frac{|l(d) - K(r)|}{l(d)}$$

For most descriptions, the length of the description  $l(d)$  for  $r$  will exceed the length of its shortest possible description  $K(r)$ . Intuitively, the less we

know about an entity, the longer our description tends to be. As our understanding of the entity improves, we should be able to remove all redundant elements from the description. There may also be cases where the description is shorter than the optimal one. In such instances, the description oversimplifies the problem, which can be equally problematic. This justifies the use of the absolute value  $|l(d) - K(r)|$  rather than simply  $l(d) - K(r)$ . Naturally, the current description might also be inaccurate, or the representation may be invalid. These concerns are addressed separately by the metrics of inaccuracy and miscoding.

In our definition of surfeit, we chose a relative measure rather than an absolute one (i.e.,  $|l(d) - K(r)|$ ) because we aim to compare the surfeit not only among different models of the same entity but also across models of different entities. We prefer to use  $K(r)$  instead of the equivalent  $l(r^*)$  to maintain consistency with the definition of inaccuracy provided in Section 4.1.

■ **Example 5.1** In a practical machine learning scenario, consider the task of classifying images of cats and dogs, where the representation consists of a set of training images. An initial complex model, burdened with excessive parameters and irrelevant features, can be seen as a lengthy "description" of the problem. The optimal model, by contrast, achieves a balance between accuracy and simplicity. Surfeit quantifies the excess complexity of the initial model relative to this optimal one. It measures the "extra" components that are not essential for accurate classification. A high surfeit indicates that the model is overly complicated, highlighting the need for simplification to improve both efficiency and generalization, a fundamental principle in machine learning for building models that perform well on unseen data. ■

The surfeit of a description is a number between 0 and 1.

**Proposition 5.1.1** Let  $r \in \mathcal{B}^*$  be a representation, and  $d \in \mathcal{D}_r^*$  one of its valid descriptions, then we have that  $0 \leq \sigma(d, r) \leq 1$ .

*Proof.* The numerator  $|l(d) - K(r)|$  is non-negative for all  $d$  and  $r$ , and the denominator  $l(d) > 0$  (as descriptions must be non-empty). Hence, the entire expression is non-negative  $\sigma(d, r) \geq 0$ .

To show that  $\sigma(d, r) \leq 1$ , observe that:

$$|l(d) - K(r)| = l(d) - K(r) \leq l(d).$$

since  $d$  is a valid description of  $r$ , and so  $l(d) \geq K(r)$ . Thus:

$$\sigma(d, r) = \frac{|l(d) - K(r)|}{l(d)} \leq \frac{l(d)}{l(d)} = 1.$$

■

The surfeit is zero when the length of the description  $l(d)$  equals the Kolmogorov complexity  $K(r)$  of the representation  $r$  of the entity, indicating that the description has achieved theoretical conciseness.

**Proposition 5.1.2** Let  $r \in \mathcal{B}^*$  be a representation, and  $d \in \mathcal{D}_r^*$  a valid description for  $r$ , then we have that  $\sigma(d, r) = 0$  if and only if  $l(d) = l(d^*)$ .

*Proof.* Assume  $l(d) = l(d^*) = K(r)$ . Then:

$$\sigma(d, r) = \frac{|l(d) - K(r)|}{l(d)} = \frac{|K(r) - K(r)|}{K(r)} = \frac{0}{K(r)} = 0.$$

Assume  $\sigma(d, r) = 0$ . Then:

$$\frac{|l(d) - K(r)|}{l(d)} = 0.$$

This implies that the numerator must be zero, i.e.:

$$|l(d) - K(r)| = 0 \Rightarrow l(d) = K(r) = l(d^*).$$

■

It is important to distinguish between conciseness and correctness. A surfeit of zero indicates that the description is as brief as theoretically possible, but not necessarily accurate. The accuracy of a description is evaluated using the inaccuracy metric. Thus, while a zero surfeit reflects the optimal compactness of a description, its correctness and reliability must be assessed separately through inaccuracy. Together, surfeit and inaccuracy provide a more complete assessment of the description's efficiency and validity.

The following example underscores the need to balance minimal surfeit with low inaccuracy in order to develop models that are both efficient and reliable.

■ **Example 5.2** In a machine learning scenario, consider a model designed to classify emails as spam or not. Suppose this model achieves a surfeit of zero, indicating optimal conciseness with no superfluous elements in its description. However, despite this streamlined complexity, the model exhibits high inaccuracy, frequently misclassifying emails. This illustrates the distinction between surfeit and inaccuracy: while the model is theoretically as concise as possible, reflected in its zero surfeit, its practical utility is compromised by incorrect classifications, as captured by the inaccuracy metric.

■

It is important to note that, in theory, more than one description may yield a surfeit of zero for the same representation. This occurs when two distinct, yet incompressible, descriptions have exactly the same length. In such cases, both descriptions are equally concise and minimal, meaning that neither contains redundant information that could be further compressed. Since surfeit measures the relative difference between the actual length of a description and the shortest possible one, any description whose length equals the Kolmogorov complexity  $K(r)$  of the representation will result in  $\sigma(d, r) = 0$ . Therefore, while the perfect description is often referred to as unique, from a theoretical standpoint, multiple descriptions of equal minimal length and incompressibility may coexist.

## 5.2 Redundancy

Our definition of surfeit compares the length of a description with the Kolmogorov complexity of the representation, not with the Kolmogorov complexity of the description itself (i.e.,  $K(d)$ ). In other words, surfeit is not a measure of the redundancy within a description. It is possible to construct an incompressible description (i.e., one without any internal redundancy) that is not the shortest possible description of the representation it refers to (see Example 2.22). Such a description would not be redundant in the traditional sense, yet it would still exhibit surfeit according to the theory of nescience.

Moreover, it might happen that the description  $d$  under consideration does not describe the representation  $r$ ; in other words,  $d \notin \mathcal{D}_r$ . For practical applications, it is useful to introduce an alternative, and arguably weaker, notion of redundancy that applies solely to the description itself, independently of any particular representation.

**Definition 5.2.1 — Redundancy.** Given a description  $d \in \mathcal{D}$ , we define the *redundancy* of the description  $d$ , denoted by  $\rho(d)$ , as

$$\rho(d) = 1 - \frac{K(d)}{l(d)}$$

The redundancy of a description  $d$  is a quantity related to the description itself, and it does not depend on the representation  $r$  being described. The following example shows that a description can have low redundancy (being incompressible) and still have high surfeit if it is longer than necessary to describe the underlying task.

■ **Example 5.3** Consider the task of computing the first  $n$  digits of  $\pi$ . Description  $d_1$  is a program that implements an algorithm to generate and print these  $n$  digits. Description  $d_2$  is a self-extracting compressed program that

simply stores the first  $n$  digits of  $\pi$  as a hard-coded string, followed by a basic routine to print them.

Description  $d_2$  does not exploit the mathematical structure of  $\pi$ , it merely reproduces the result. Therefore, relative to the representation of the task (computing  $\pi$ ), the surfeit  $\sigma(d_2, r)$  is high because  $l(d_2) \gg K(r)$ .

In contrast, description  $d_1$  is significantly shorter since it captures the generative process behind  $\pi$ . While  $d_1$  may still contain some redundancy  $\rho(d_1) > 0$ , for instance due to suboptimal coding practices, its surfeit remains low since it is near the minimal length required to describe the task effectively.

■

The redundancy of a description always falls within the range  $[0, 1]$ .

**Proposition 5.2.1** We have that  $0 \leq \rho(d) \leq 1$  for all  $d \in \mathcal{D}$ .

*Proof.* Since Kolmogorov complexity is always less than or equal to the length of the string, we have  $0 \leq K(d) \leq l(d)$ , which implies that  $0 \leq \frac{K(d)}{l(d)} \leq 1$ . Subtracting from 1 reverses the inequalities. ■

Finally, next proposition formalizes our intuition that the surfeit of a description is greater or equal than its redundancy.

**Proposition 5.2.2** Let  $r \in \mathcal{B}^*$  be a representation, and  $d \in \mathcal{D}_r^*$  one of its valid descriptions, then we have that  $\rho(d) \leq \sigma(d, r)$ .

*Proof.* Proving that  $\rho(d) \leq \sigma(d, r)$  is equivalent to prove that  $K(d) \geq K(r)$  for all  $d$ . Lets assume that there exist a  $d$  such that  $K(d) < K(r)$ , that would mean there exists a Turing machine  $\langle TM, a \rangle$  such that  $TM(a) = r$  but  $l(\langle TM, a \rangle) < K(r)$ . That is a contradiction with the fact that  $K(r)$  is the length of the shortest possible Turing machine that prints  $r$ . ■

It would be very nice if Proposition 5.2.2 applies to all possible description. Unfortunately, the proposition is true only when we deal with valid descriptions (from  $\mathcal{D}_r^*$ ).

### 5.3 Conditional Surfeit

We are interested in studying how the surfeit of a description for a representation is affected when some background knowledge is assumed. In particular, we examine the case where a description is constructed under the assumption that some prior information is already known. The surfeit of such a description is referred to as the *conditional surfeit*.

**Definition 5.3.1** Let  $r \in \mathcal{B}^*$  be a representation,  $s \in \mathcal{B}^*$  a string, and  $d \in \mathcal{D}$  a description of  $r$  given  $s$ . We define the *conditional surfeit* of the conditional description  $d_{r|s}$ , denoted by  $\sigma(d_{r|s})$ , as:

$$\sigma(d_{r|s}) = 1 - \frac{K(r|s)}{l(d_{r|s})}$$

This definition is primarily motivated by practical considerations. When we assume perfect knowledge of  $s$ , we can isolate and study the informational content of  $r$  that is not already covered by  $s$ , that is, the new knowledge introduced by  $r$  relative to  $s$ . Conditional surfeit thus allows us to quantify the conciseness of a description in light of what is already known.

Conditional surfeit, being a relative measure, is a number between 0 and 1.

**Proposition 5.3.1** Let  $r \in \mathcal{B}^*$  be a representation,  $s \in \mathcal{B}^*$  be a string, and  $d \in \mathcal{D}$  be a description of  $r$  given  $s$ . We have that  $0 \leq \sigma(d_{r|s}) \leq 1$ .

*Proof.* Given that  $l(d_{r|s}) > 0$  and that  $K(r|s) > 0$ , since they are the lengths of non-empty strings, and that  $l(d_{r|s}) \geq K(r|s)$ . ■

Intuition tell us that the surfeit of a description could only decrease if we assume the background knowledge given by the description of another topic. This is because we require that this background knowledge must be a perfect description (it presents no surfeit). However, as it was the case of joint surfeit, we have to wait until Chapter 6 to formalize this intuition.

In the same way we introduced the concept of redundancy of a description as a weaker version of the concept of surfeit, we can also introduce the concept of conditional redundancy as a weaker version of the concept of conditional surfeit.

**Definition 5.3.2** Let  $s \in \mathcal{B}^*$  be a string, and  $d \in \mathcal{D}$  be a conditional description given  $s$ . We define the *conditional surfeit* of the conditional description  $d_{t|s^*}$ , denoted by  $\rho(d_{t|s^*})$ , as:

$$\rho(d_{t|s^*}) = 1 - \frac{K(d_{t|s^*})}{l(d_{t|s^*})}$$

Conditional surfeit is a relative measure, and so, a number between 0 and 1.

**Proposition 5.3.2** We have that  $0 \leq \rho(d_{t|s^*}) \leq 1$  for all  $t, s$  and all  $d_{t|s^*}$ .

*Proof.* Given that  $K(d_{t|s^*}) \leq l(d_{t|s^*})$  we have that  $\frac{K(d_{t|s^*})}{l(d_{t|s^*})} \leq 1$  and so,  $1 -$

$\frac{K(d_{t|s^*})}{l(d_{t|s^*})} \geq 0$ . Also, since  $\frac{K(d_{t|s^*})}{l(d_{t|s^*})} > 0$  (both quantities are positive integers), we have that  $1 - \frac{K(d_{t|s^*})}{l(d_{t|s^*})} \leq 1$ . ■

Finally, we can extend our concepts of conditional surfeit and conditional redundancy to multiple, but fine, number of topics.

**Definition 5.3.3** Let  $t, s_1, s_2, \dots, s_n \in \mathcal{T}$  be a finite collection of topics, and let  $d_{t|s_1^*, s_2^*, \dots, s_n^*}$  any conditional description of  $t$  given  $s_1, s_2, \dots, s_n$ . We define the *conditional surfeit* of the description  $d_{t|s_1^*, s_2^*, \dots, s_n^*}$ , denoted by  $\sigma(d_{t|s_1^*, s_2^*, \dots, s_n^*})$ , as:

$$\sigma(d_{t|s_1^*, s_2^*, \dots, s_n^*}) = 1 - \frac{K(t \mid s_1^*, s_2^*, \dots, s_n^*)}{l(d_{t|s_1^*, s_2^*, \dots, s_n^*})}$$

And the *conditional redundancy* of the description  $d_{t_1, t_2, \dots, t_n}$ , denoted by  $\rho(d_{t_1, t_2, \dots, t_n})$ , as:

$$\rho(d_{t_1, t_2, \dots, t_n}) = 1 - \frac{K(d_{t_1, t_2, \dots, t_n})}{l(d_{t_1, t_2, \dots, t_n})}$$

It is easy to show that the properties of conditional surfeit and conditional redundancy apply to the case of multiple topics as well.

## 5.4 Decreasing Surfeit

Our objective is to reduce the surfeit of our current best description  $d_1$ , thereby enhancing our understanding of the original entity. This improvement may involve either refining  $d_1$  to eliminate redundant symbols or developing an entirely new description based on a different modeling approach. In either case, the result is a new description  $d_2$ . In this section, we analyze how the introduction of a new description  $d_2$  affects the surfeit relative to that of the original description  $d_1$ .

**Definition 5.4.1** Let  $r \in \mathcal{B}^*$  be a representation, and let  $d_1, d_2 \in \mathcal{D}$  be two descriptions. We define the *variation of surfeit* between the descriptions  $d_1$  and  $d_2$ , with respect to  $r$ , denoted by  $\Delta_\sigma^a(d_1, d_2, r)$ , as:

$$\Delta_\sigma^a(d_1, d_2, r) = \sigma(d_1, r) - \sigma(d_2, r)$$

Since surfeit is bounded between 0 and 1, the maximum possible variation is  $\pm 1$ . A positive value of  $\Delta_\sigma$  indicates that  $d_2$  is preferable to  $d_1$  in terms of surfeit. Conversely, a negative value suggests that  $d_1$  is more concise than  $d_2$ .

It is important to emphasize that a new description may also introduce a substantial increase in inaccuracy, potentially offsetting the improvement in surfeit. For a detailed discussion of inaccuracy, refer to Chapter 4, and for an explanation of how inaccuracy and surfeit combine into the unified metric of nescience, see Chapter 6.

We can also introduce a relative measure of the variation in surfeit when transitioning from description  $d_1$  to  $d_2$ .

**Definition 5.4.2** Let  $r \in \mathcal{B}^*$  be a representation, and  $d_1, d_2 \in \mathcal{D}$  be two descriptions. We define the *relative variation of surfeit* between descriptions  $d_1$  and  $d_2$ , denoted by  $\Delta_\sigma^r(d_1, d_2, r)$ , as:

$$\Delta_\sigma^r(d_1, d_2, r) = \frac{\sigma(d_1, r) - \sigma(d_2, r)}{\sigma(d_1, r)}$$

provided that  $\sigma(d_1, r) \neq 0$ .

A value of 0 indicates that there is no change in surfeit between the two descriptions. A value of 1 corresponds to a complete elimination of surfeit, meaning that the new description  $d_2$  has zero surfeit, while the original description  $d_1$  had a nonzero surfeit. Negative values indicate that surfeit has increased, suggesting that the new description  $d_2$  either includes more irrelevant or redundant symbols than  $d_1$ , or is too short to adequately represent the entity, thus omitting important information. It is important to note that the relative variation can become arbitrarily negative. This occurs when the surfeit of the original description  $\sigma(d_1, r)$  is close to zero, while the surfeit of the new description  $\sigma(d_2, r)$  is significantly larger. In such cases, the denominator becomes very small, causing the relative variation to diverge toward  $-\infty$ .

As surfeit approaches zero, relative variations become increasingly unstable. A small absolute change can result in a large relative variation when the initial surfeit is very low. For example, if  $\sigma(d_1, r) = 0.1$  and surfeit decreases by 0.05, the relative improvement is 50%. However, if  $\sigma(d_1, r) = 0.9$  and the same absolute reduction occurs, the relative improvement is only about 5.6%. Therefore, both absolute and relative variations are important for assessing the significance and scale of changes in surfeit.

## 5.5 Surfeit-inaccuracy rate of Change

In the preceding section, we examined how the surfeit of a description can be reduced by modifying that description, either by adding missing symbols, or by removing redundant, or irrelevant symbols. In this section, we turn our attention to a more general approach for reducing the surfeit associated with

an entity, by allowing a increase in the inaccuracy of the description. Rather than reducing the surfeit in isolation, it may be more effective to modify both, surfeit and inaccuracy, simultaneously.

The balance between the amount of inaccuracy we are willing to accept in order to achieve a reduction in surfeit is referred to as the *surfeit-inaccuracy trade-off*. For a broader discussion of trade-offs in multi-objective optimization, refer to Section F.6.2.

**Definition 5.5.1** Let  $d_1, d_2 \in \mathcal{D}$  be two descriptions, and  $r \in \mathcal{B}^*$  a representation. We define the *rate of change between the surfeit and inaccuracy* of the descriptions  $d_1, d_2$  given the representation  $r$ , denoted by  $\Delta_{\sigma\iota}(d_1, d_2, r)$  as:

$$\Delta_{\sigma\iota}(x_1, x_2) = \frac{\sigma(d_2, r) - \sigma(d_1, r)}{\iota(d_2, r) - \iota(d_1, r)}$$

provided that  $\iota(d_2, r) - \iota(d_1, r) \neq 0$ .

The ratio  $\Delta_{\sigma\iota}$  represents the rate of change between surfeit and inaccuracy when transitioning from the first description to the second. A positive value of  $\Delta_{\sigma\iota}$  implies that both quantities, surfeit and inaccuracy, either decrease (which is desirable) or increase (which is undesirable). The interpretation becomes more nuanced when  $\Delta_{\sigma\iota}$  is negative, indicating that one of the quantities decreases while the other increases. In such cases, two scenarios must be considered:

- (i) Surfeit decreases and inaccuracy increases, we aim for  $\Delta_{\sigma\iota}(d_1, d_2, r) < M$ , where  $M < -1$ , thereby ensuring that the reduction in surfeit compensates for the increase in inaccuracy.
- (ii) Surfeit increases and inaccuracy decreases, we aim for  $\Delta_{\sigma\iota}(d_1, d_2, r) > M$ , where  $-1 < M < 0$ , thereby ensuring that the reduction in inaccuracy justifies the increase in surfeit.

In both cases, caution is warranted when the change in inaccuracy is small, as it can disproportionately affect the ratio and potentially lead to misleading conclusions.

**■ Example 5.4** Let  $d_1, d_2 \in \mathcal{D}$  be two descriptions, and  $r \in \mathcal{B}^*$  a representation. For  $d_1$ , the surfeit  $\sigma(d_1, r_1)$  is 0.40, and the inaccuracy  $\iota(d_1, r)$  is 0.15. For  $d_2$ , the surfeit  $\sigma(d_2, r)$  is 0.20 (a decrease from 0.40), and the inaccuracy  $\iota(d_2, r)$  is 0.25 (an increase from 0.15). Using the definition of the rate of change, we compute:

$$\Delta_{\sigma\iota}(d_1, d_2, r) = \frac{0.40 - 0.20}{0.15 - 0.25} = \frac{0.20}{-0.10} = -2$$

In this case, transitioning from description  $d_1$  to  $d_2$  results in a decrease of

surfeit by 0.20 units and an increase in inaccuracy by 0.10 units. The rate of change is  $-2$ . ■

Having a very small change in inaccuracy can significantly amplify the value of the rate of change, potentially giving the misleading impression that surfeit and inaccuracy are varying at an extreme rate, even when the actual changes are minor. This phenomenon is illustrated in the following example.

■ **Example 5.5** Let  $d_1 \in \mathcal{D}$  be a description and  $r \in \mathcal{B}^*$  a representation with an surfeit  $\sigma(d_1, r)$  of 0.35 and a inaccuracy  $\iota(d_1, r)$  of 0.20. Let  $d_2 \in \mathcal{D}$  be a second description with an surfeit  $\sigma(d_2, r)$  of 0.30 (a slight decrease from 0.35) and a inaccuracy  $\iota(d_2, r)$  of 0.2001 (a very small increase from 0.20). Applying the definition of the rate of change:

$$\Delta_{\sigma\iota}(d_1, d_2, r) = \frac{0.35 - 0.30}{0.20 - 0.2001} = \frac{0.05}{-0.0001} = -500$$

The rate of change is  $-500$ , which may misleadingly suggest a dramatic shift. In reality, the surfeit only decreased by 0.05 units, and the inaccuracy increased by a negligible 0.0001 units. The extremely small denominator inflates the result, making the change appear far more significant than it actually is. ■

As we attempt to optimize both surfeit and inaccuracy, we inevitably encounter descriptions where improving one objective necessitates compromising the other. The set of such "best trade-off" configurations constitutes the Pareto frontier (see Section F.6). Points on the Pareto frontier are said to be Pareto optimal because any attempt to improve one objective leads to a deterioration in the other.

**Definition 5.5.2** Let  $d_1 \in \mathcal{D}$  be a description and  $r \in \mathcal{B}^*$  a representation. The description  $d_1$  is said to be a Pareto point with respect to surfeit  $\sigma$  and inaccuracy  $\iota$  if there does not exist another description  $d_2$  such that:

- 1  $\sigma(d_2, r) \leq \sigma(d_1, r)$  and  $\iota(d_2, r) \leq \iota(d_1, r)$ , and
- 2  $\sigma(d_2, r) < \sigma(d_1, r)$  or  $\iota(d_2, r) < \iota(d_1, r)$ .

In simpler terms, a description is Pareto optimal if: (i) no other description is better in both surfeit and inaccuracy, and (ii) any improvement in one metric necessarily results in a worsening of the other.

The rate of change  $\Delta_{\sigma\iota}$  between any two Pareto optimal points provides insight into how the trade-off between surfeit and inaccuracy evolves along the Pareto frontier. If decision-makers are more sensitive to changes in surfeit than to inaccuracy, they may prefer configurations with a less negative  $\Delta_{\sigma\iota}$ . Conversely, if inaccuracy is of greater concern, they may accept solutions where  $\Delta_{\sigma\iota}$  indicates a larger increase in surfeit in exchange for a smaller gain in inaccuracy.

## 5.6 Surfeit of Areas

The concept of surfeit can be extended to research areas, to quantitative measure the amount of extra effort we are using to describe the topics of the area.

**Definition 5.6.1** Let  $A \subset \mathcal{T}$  be an area with known subset  $\hat{A} = \{t_1, t_2, \dots, t_n\}$ , and let  $d_{\hat{A}}$  be a description. We define the *surfeit of the description*  $d_{\hat{A}}$  as:

$$\sigma(d_{\hat{A}}) = 1 - \frac{K(\langle t_1, t_2, \dots, t_n \rangle)}{l(d_{\hat{A}})}$$

As it was the case of the concept of redundancy, in general we do not know the complexity of the area  $K(\hat{A})$ , and so, in practice, it must be approximated by the complexity of the descriptions themselves  $K(\hat{d}_{\hat{A}})$ . However, in the particular case of areas, we could have also problems with the quantity  $\hat{d}_{\hat{A}}$ , since it requires to study the conditional descriptions of the topics included in the area.

**Definition 5.6.2** Let  $A \subset \mathcal{T}$  be an area with known subset  $\hat{A} = \{t_1, t_2, \dots, t_n\}$ , and let  $d_{\hat{A}}$  be a description. We define the *weak redundancy of the description*  $d_{\hat{A}}$  as:

$$\rho(d_{\hat{A}}) = 1 - \frac{K(d_{\hat{A}})}{l(d_{\hat{A}})}$$

## References

The concept of redundancy has been also investigated in the context of information theory, since we are interested on using codes with low redundancy (see for example [Abr63]).



## 6. Nescience

*There are known knowns. These are things we know that we know. There are known unknowns. That is to say, there are things that we know we don't know. But there are also unknown unknowns. There are things we don't know we don't know.*

Donald Rumsfeld

Following the foundational Chapter ??, which introduced the core concepts of entity, representation, and description, and the subsequent Chapters 3, 4, and 5, which developed the new metrics of miscoding, inaccuracy, and surfeit, we are now prepared to delve into the central concept of nescience in this chapter, focusing on its fundamental properties.

Unlike Shannon entropy or Kolmogorov complexity, which measure information, nescience quantifies the absence of information, that is, what remains unknown. The theory of nescience characterizes our ignorance about a research entity through the three previously introduced metrics: miscoding, inaccuracy, and surfeit. Miscoding assesses how accurately an entity is represented as a string of symbols; inaccuracy measures how well our best available model describes this representation; and surfeit evaluates the descriptive efficiency of the model, as reflected in its length or symbol count.

These three metrics are inherently interdependent and often in conflict: reducing one can lead to an increase in another. The central challenge, therefore, is to develop a method for simultaneously minimizing all three. This requirement reflects, in our view, the fundamental nature of scientific inquiry as a multi-objective optimization problem.

One of the most significant outcomes of our definition of nescience, grounded in the metrics of miscoding, inaccuracy, and surfeit, is its capacity to partition the domain of research topics into two distinct areas. The first is the known unknown, which includes topics we are aware we do not fully understand, yet can recognize and acknowledge our incomplete knowledge. The second is the unknown unknown, comprising topics that have not yet been discovered or conceptualized. A key application of the theory of nescience is its use as a methodological framework for identifying and exploring what lies within the unknown unknown.

Another noteworthy implication of the nescience framework is the counterintuitive idea that, for certain topics, continued research may be counterproductive. In these cases, additional investigation may actually increase our ignorance rather than reduce it. This occurs when we reach a critical threshold beyond which our descriptions become more inaccurate, overly complex, or based on flawed representations, preventing any real progress toward understanding.

## 6.1 Nescience

Intuitively, our understanding of an entity should be based in the quality of the model we use to describe it, specifically, its ability to explain why things happen. Within the theory of nescience, we propose a quantitative measure of our ignorance concerning a research entity, based on three components: the miscoding of a string-based representation of the entity, and the inaccuracy and surfeit of the model describing that representation. Miscoding captures how accurately the representation encodes the entity, inaccuracy reflects how well the model describes the representation, and surfeit quantifies the extent of unnecessary effort embedded in the model. We argue that the goal of science should be to minimize all three quantities: miscoding, inaccuracy, and surfeit. Unfortunately, these metrics are inherently conflicting, reducing one may lead to an increase in one or both of the others.

According to the theory of nescience, science can be viewed as a multi-objective optimization problem<sup>1</sup> (see Section F.6):

---

<sup>1</sup>Technically speaking, science is a deterministic, discrete, nonlinear, nonconvex, nondifferentiable multiobjective optimization problem with a single decision maker.

### The Science Problem

$$\begin{aligned} & \text{minimize} && \{\mu(r), \iota(d, r), \sigma(d, r)\} \\ & \text{subject to} && (r, d) \in \mathcal{B}^* \times \mathcal{D} \end{aligned}$$

A *scientific method*, further discussed in Chapter G, refers to any algorithm or computable procedure capable of solving, or closely approximating a solution to, the above minimization problem. This includes a broad class of techniques and methodologies aimed at systematically reducing the values of miscoding, inaccuracy, and surfeit. In doing so, scientific methods contribute to improving the accuracy and conciseness of our representations and models, thereby advancing our understanding of the world.

The feasible region is the Cartesian product  $\mathcal{B}^* \times \mathcal{D}$ , where  $\mathcal{B}^*$  denotes the set of finite binary strings and  $\mathcal{D}$  the set of descriptions. The decision vectors are pairs  $(r, d)$ , referred to as *hypotheses*, consisting of a representation and a description. The objective functions to be minimized are miscoding, surfeit, and inaccuracy. The objective space is the subset  $\mathbf{Z} \subset \mathbb{R}^3$ , whose elements are the objective vectors.

In our formulation of science and the scientific method, we deliberately exclude the set  $\mathcal{E}$  of entities. Requiring direct knowledge of an entity  $e \in \mathcal{E}$  would render the scientific problem ill-posed for most research areas. Science, at its core, is a matter of manipulating strings of symbols. From a practical standpoint, it is about discovering strings that have meaningful interpretations in the real world and can be used to solve concrete problems. From a more theoretical perspective, the aim of science can be seen as the attempt to understand the workings of an unknown abstract oracle.

If the set  $\mathcal{R}_e$  of representations for a particular entity  $e$  is known, or approximately known, we can restrict the science problem to:

$$\begin{aligned} & \text{minimize} && \{\mu(r), \iota(d, r), \sigma(d, r)\} \\ & \text{subject to} && (r, d) \in \mathcal{R}_e \times \mathcal{D} \end{aligned}$$

Within the theory of nescience, our primary focus lies in the decision space  $\mathcal{B}^* \times \mathcal{D}$ , the space of representations and descriptions, rather than in the objective space  $\mathbf{Z} \subset \mathbb{R}^3$  of metric values. In the following definitions, we revisit key concepts from multiobjective optimization (see Section F.6) as they apply specifically to the science problem.

### Pareto Optimality

If the representation and description currently used to characterize an entity are not perfect, our goal is to find an alternative representation or description

that reduces at least one of the metrics miscoding, inaccuracy, or surfeit without increasing the value of any of the others.

**Definition 6.1.1** We say that a hypothesis  $(r, d) \in \mathcal{B}^* \times \mathcal{D}$  *dominates* another hypothesis  $(r', d') \in \mathcal{B}^* \times \mathcal{D}$  if it improves at least one of the metrics miscoding, inaccuracy, or surfeit without worsening either of the other two.

For example, we might identify a new representation that encodes the entity more accurately without degrading the quality of its description. Alternatively, we could find a new description that improves accuracy without increasing surfeit, or a more concise description that preserves accuracy.

■ **Example 6.1** Consider an experiment in which we collect a set of observations  $r$  and apply a mathematical model  $f_1$  from a model family  $\mathcal{M}_1$ , resulting in an inaccuracy of  $i_1$ . Later, we fit a second model  $f_2$  from a different model family  $\mathcal{M}_2$ , which is smaller in size (i.e., has lower surfeit) but yields the same inaccuracy  $i_1$ . In this case, the hypothesis  $B = (r, f_2)$  dominates the hypothesis  $A = (r, f_1)$ , even though it is not better in terms of inaccuracy alone. ■

For most entities, there does not exist a single solution that simultaneously minimizes all three metrics. Instead, we encounter a set of Pareto optimal solutions that define an optimal frontier.

**Definition 6.1.2** We say that a hypothesis  $(r, d) \in \mathcal{B}^* \times \mathcal{D}$  is *Pareto optimal* if there does not exist another hypothesis  $(r', d') \in \mathcal{B}^* \times \mathcal{D}$  such that  $(r', d')$  dominates  $(r, d)$ . The set of Pareto optimal solutions, denoted by  $\mathbf{P}_{\mathcal{B}^* \times \mathcal{D}}$ , is called the *Pareto frontier*.

In the realm of scientific research, the concept of the Pareto frontier, as defined by the set of Pareto optimal solutions  $\mathbf{P}_{\mathcal{B}^* \times \mathcal{D}}$ , plays a crucial role. It delineates the boundary of optimal trade-offs among the conflicting metrics of miscoding, inaccuracy, and surfeit, such that none can be improved without worsening at least one of the others. This frontier represents the spectrum of best-achievable balances, guiding researchers to identify models and representations that offer the most scientifically rigorous understanding of their subject matter (see Section 6.3).

However, in certain situations or specific applications, it may be reasonable to adopt a solution that is not Pareto optimal. For instance, one might choose to prioritize a particular metric due to its relevance or importance to the research objectives, accepting less favorable values for the remaining metrics as a necessary trade-off (see Section 6.2).

Building on the concept of Pareto optimality, where a solution is consid-

ered optimal if no other solution improves one objective without worsening another, we introduce the notion of weak Pareto optimality. A hypothesis is said to be weakly Pareto optimal if there is no other hypothesis that improves all objectives simultaneously. This concept is broader than Pareto optimality, as it includes solutions that may not be the best in any single objective but are not strictly outperformed in every dimension.

**Definition 6.1.3** We say that a hypothesis  $(r, d) \in \mathcal{B}^* \times \mathcal{D}$  *weakly dominates* another hypothesis  $(r', d') \in \mathcal{B}^* \times \mathcal{D}$  if it improves all three metrics miscoding, inaccuracy, and surfeit simultaneously. That is, if  $\mu(r') < \mu(r)$ ,  $\iota(d', r') < \iota(d, r)$ , and  $\sigma(d', r') < \sigma(d, r)$ .

A hypothesis is *weakly Pareto optimal* if there does not exist another hypothesis that improves all three metrics: miscoding, inaccuracy, and surfeit.

**Definition 6.1.4** We say that a hypothesis  $(r, d) \in \mathcal{B}^* \times \mathcal{D}$  is *weakly Pareto optimal* if there does not exist another hypothesis  $(r', d') \in \mathcal{B}^* \times \mathcal{D}$  such that  $(r', d')$  weakly dominates  $(r, d)$ . The set of weakly Pareto optimal solutions, denoted by  $\mathbf{P}_{\mathcal{B}^* \times \mathcal{D}}$ , is called the *weakly Pareto frontier*.

If a hypothesis is weakly Pareto optimal, it means there is no other hypothesis that improves all three metrics simultaneously. However, it is still possible to find a hypothesis that improves one of the metrics without worsening the others, that is, a hypothesis that is Pareto optimal. Thus, the Pareto frontier is a subset of the weakly Pareto frontier. In the theory of nescience, we focus primarily on the set of Pareto optimal solutions rather than the set of weakly Pareto optimal ones.

■ **Example 6.2** Based on the assumptions of Example 6.1, hypothesis *A* could still be weakly Pareto optimal, but it cannot be Pareto optimal, since it is dominated by hypothesis *B*. However, it is not weakly dominated by hypothesis *B*. ■

The concepts of Pareto and weakly Pareto optimality can also be particularized to the case in which the set  $\mathcal{R}_e$  of representations of a particular entity *e* is known.

## Range of Solutions

As discussed in Section F.6.1, an objective vector that achieves the minimum possible value for all objective functions is termed the ideal objective vector. For the science problem, this ideal vector is represented by the origin  $(0, 0, 0)$ , symbolizing the complete elimination of miscoding, inaccuracy, and surfeit.

**Proposition 6.1.1** The ideal objective vector for the science problem is the origin  $\$(0, 0, 0)$ .

*Proof.* Proposition 3.1.1 established that miscoding is greater than or equal to zero, and Proposition 3.1.2 showed that it can be equal to zero. Likewise, Proposition 4.5.1 demonstrated that inaccuracy is non-negative, while Proposition 4.1.2 confirmed that a value of zero is attainable. Finally, Proposition 5.2.1 stated that surfeit is at least zero, and Proposition 5.1.2 verified that zero surfeit is achievable. ■

A hypothesis  $(r, d) \in \mathcal{B}^* \times \mathcal{D}$  is said to be ideal if it exhibits zero miscoding, zero inaccuracy, and zero surfeit. This implies that the representation  $r$  is valid, the model  $d$  produces  $r$  as output, and there exists no shorter model  $d'$  that also achieves zero inaccuracy. Intuitively, a hypothesis  $(r, d)$  is ideal if there exists an entity  $e \in \mathcal{E}$  such that  $r$  perfectly encodes  $e$ , and  $d$  is both an accurate and minimal model of  $r$ .

Ideal hypotheses embody the notion of perfect knowledge within the theory of nescience. Unfortunately, in most practical applications, reaching the ideal objective vector is not feasible due to the inherently conflicting nature of the metrics: miscoding, inaccuracy, and surfeit.

■ **Example 6.3** A research topic for which it is impossible to reach the ideal objective vector is weather prediction. In this case, the entity under study is the atmosphere over a geographical region. The representation of this entity (typically a set of meteorological measurements such as temperature, pressure, and humidity) is inherently flawed due to the limited spatial and temporal resolution of sensors, noise in the data, and incomplete coverage, particularly over oceans and remote areas. As a result, miscoding is strictly greater than zero. Furthermore, even the most sophisticated atmospheric models, which numerically approximate the physical laws governing weather dynamics, cannot produce perfectly accurate forecasts due to the chaotic nature of the system, the need for simplifying assumptions, and errors in initial conditions, ensuring that inaccuracy also remains greater than zero. Finally, these models are large, complex, and often include redundant components or overly general submodules, making them far from minimal in size; thus, surfeit is also non-zero. ■

The upper bound of the Pareto optimal set is given by the nadir objective vector. In the theory of nescience, the nadir vector is the point  $(1, 1, 1)$ , corresponding to maximum miscoding, maximum inaccuracy, and maximum surfeit.

**Proposition 6.1.2** The nadir objective vector of the science problem is the vector  $(1, 1, 1)$ .

*Proof.* Proposition 3.1.1 demonstrated that miscoding is greater than or equal to zero, and Proposition 3.1.2 showed that it can be equal to zero. Similarly,

Proposition 4.5.1 established that inaccuracy is greater than or equal to zero, and Proposition 4.1.2 indicated that a zero value is attainable. Finally, Proposition 5.2.1 showed that surfeit is greater than or equal to zero, and Proposition 5.1.2 confirmed that it can also reach zero. Therefore, since all three metrics are bounded between 0 and 1, the upper bound of the objective region is  $(1, 1, 1)$ . ■

The nadir vector represents a state of complete ignorance: a hypothesis  $(r, d)$  in which the representation  $r$  contains no meaningful information about the entity  $e$  under study, the description  $d$  generates a string entirely unrelated to  $r$ , and the description is of maximal length. This extreme point illustrates the worst-case scenario in terms of scientific knowledge: maximum miscoding, total inaccuracy, and maximum unnecessary complexity.

■ **Example 6.4** Consider the case of studying the physical law governing the motion of a pendulum. Suppose we define a hypothesis  $(r, d)$ , where the representation  $r$  is a random binary string encoding information entirely unrelated to the pendulum, such as the binary representation of a shuffled deck of cards. The description  $d$  is a program that outputs an unrelated string, for example, one billion digits of  $\pi$ . In this scenario, miscoding is maximal because the representation bears no connection to the entity being studied, inaccuracy is maximal because the description produces a string entirely different from the representation, and surfeit is also maximal since the description is very long compared to the length of  $r$ . This hypothesis  $(r, d)$  reaches the nadir objective vector  $(1, 1, 1)$ , reflecting a state of complete ignorance about the entity. ■

## Trade-offs

In Section 4.4, we analyzed the trade-off between inaccuracy and miscoding:

$$\Delta_{\iota\mu}(\mathbf{x}_1, \mathbf{x}_2) = \frac{\iota(d_2, r_2) - \iota(d_1, r_1)}{\mu(r_2) - \mu(r_1)}$$

where  $\mathbf{x}_1 = (r_1, d_1)$  and  $\mathbf{x}_2 = (r_2, d_2)$  are two hypotheses.

This ratio quantifies the rate at which inaccuracy changes relative to miscoding when transitioning between two hypotheses. A positive value of  $\Delta_{\iota\mu}$  indicates that both metrics vary in the same direction, either improving or deteriorating together, whereas a negative value reflects a trade-off between them.

In Section 5.5, we similarly studied the trade-off between surfeit and inaccuracy:

$$\Delta_{\sigma\iota}(\mathbf{x}_1, \mathbf{x}_2) = \frac{\sigma(d_2, r) - \sigma(d_1, r)}{\iota(d_2, r) - \iota(d_1, r)}$$

This ratio captures how surfeit changes with respect to inaccuracy when the representation remains fixed and only the description changes. Again, a positive value of  $\Delta_{\sigma_l}$  indicates that both metrics are moving in the same direction, either improving or deteriorating together, whereas a negative value reflects a trade-off between them.

These trade-off ratios provide a local, quantitative tool for evaluating whether a change in hypothesis moves us toward Pareto optimality or away from it, and how it relates to the extreme points represented by the ideal and nadir objective vectors.

In this section we introduce a unified framework that provides a global trade-offs.

**Definition 6.1.5** Let  $\mathbf{x}_1 = (r_1, d_1)$  and  $\mathbf{x}_2 = (r_2, d_2)$  be two hypotheses. We define the *nescience trade-off vector* between  $\mathbf{x}_1$  and  $\mathbf{x}_2$  as:

$$\Delta_{\text{nescience}}(\mathbf{x}_1, \mathbf{x}_2) = \left( \frac{\iota(d_2, r_2) - \iota(d_1, r_1)}{\mu(r_2) - \mu(r_1)}, \frac{\sigma(d_2, r_2) - \sigma(d_1, r_1)}{\iota(d_2, r_2) - \iota(d_1, r_1)} \right)$$

provided that  $\mu(r_2) \neq \mu(r_1)$  and  $\iota(d_2, r_2) \neq \iota(d_1, r_1)$ .

This vector  $\Delta_{\text{nescience}}(\mathbf{x}_1, \mathbf{x}_2)$  describes the rate of change of inaccuracy relative to miscoding, and the rate of change of surfeit relative to inaccuracy.

**Definition 6.1.6** Given two hypotheses  $\mathbf{x}_1 = (r_1, d_1)$  and  $\mathbf{x}_2 = (r_2, d_2)$ , we define the *unified trade-off magnitude*, denoted  $\Theta(\mathbf{x}_1, \mathbf{x}_2)$ , as:

$$\Theta(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\left( \frac{\iota(d_2, r_2) - \iota(d_1, r_1)}{\mu(r_2) - \mu(r_1)} \right)^2 + \left( \frac{\sigma(d_2, r_2) - \sigma(d_1, r_1)}{\iota(d_2, r_2) - \iota(d_1, r_1)} \right)^2}$$

whenever both denominators are non-zero.

A small  $\Theta$  value suggests an efficient trade-off: significant improvement in one metric with minor cost in others. A large  $\Theta$  indicates a steep or unbalanced trade-off path in the nescience objective space. This formulation complements the concepts of Pareto dominance and optimality by quantifying how sharply a transition between hypotheses navigates the trade-offs among conflicting objectives.

## 6.2 Minimizing Nescience

From a mathematical perspective, any solution within the Pareto optimal set is considered a valid answer to the Science Problem. In fact, the problem is formally solved once all Pareto optimal solutions have been identified. However, in scientific practice, this is often not sufficient. Researchers

usually seek a single, most appropriate solution that best aligns with the priorities or goals of the investigation.

In multi-objective optimization, a decision maker (see Section F.6) is an entity (either a human agent, a set of predefined criteria, or an algorithm) responsible for selecting one solution from the set of Pareto optimal hypotheses. The decision maker incorporates external preferences, priorities, or domain-specific constraints to guide the selection process. Its role is to introduce a preference relation that induces an ordering over the Pareto set, thereby allowing for the identification of the most suitable hypothesis according to the specific goals or values of the scientific inquiry.

To formalize this within the theory of nescience, we introduce the notion of a nescience decision maker, defined as a scalar-valued function that evaluates and ranks hypotheses based on their levels of miscoding, inaccuracy, and surfeit. This function reflects the relative preference for each hypothesis.

**Definition 6.2.1** Let  $(r, d) \in \mathcal{B}^* \times \mathcal{D}$  be a hypothesis. A decision maker for the science problem is a multivariate function  $V : \mathbb{R}^3 \rightarrow \mathbb{R}$ , that assigns to each triplet  $(\mu(r), \iota(d, r), \sigma(d, r))$  the real value  $V(\mu(r), \iota(d, r), \sigma(d, r))$ .

The function  $V$  is often referred to as a *value function* or *utility function*, and it encodes the preferences or priorities of the scientist or research community. Depending on the application,  $V$  may treat the three metrics equally, emphasize one over the others, or apply a more sophisticated transformation to reflect factors such as risk tolerance, interpretability, or domain-specific considerations.

This formalism enables the selection of a single hypothesis from the Pareto frontier, effectively transforming the multi-objective problem into a single-objective one guided by scientific judgment.

## 6.2.1 Global Criterion

The global criterion method (see Section F.6.3) is an approach to solving multi-objective optimization problems by minimizing the distance between a reference point and the feasible region in the objective space. The reference point is typically chosen to be the ideal vector, which in our case corresponds to the origin  $(0, 0, 0)$ .

Different distance metrics can be used in this framework. For example, the global criterion based on the origin and the Euclidean distance leads to the following minimization problem:

$$\begin{aligned} \text{minimize} \quad & \sqrt{\mu(r)^2 + \iota(d, r)^2 + \sigma(d, r)^2} \\ \text{subject to} \quad & (r, d) \in \mathcal{B}^* \times \mathcal{D} \end{aligned}$$

As shown in Proposition F.6.1, the solutions obtained through the global criterion method are guaranteed to be Pareto optimal. However, it is important to note that this method does not consider all Pareto optimal solutions. In particular, optimal points that lie far from the reference point are excluded.

■ **Example 6.5** Consider the following three Pareto optimal solutions:

- $A = (0.1, 0.8, 0.9)$
- $B = (0.8, 0.1, 0.9)$
- $C = (0.5, 0.5, 0.5)$

Although all three solutions are Pareto optimal, a global criterion based on Euclidean distance may select only  $C$ , as it is closest to the origin. However,  $A$  and  $B$  represent equally valid trade-offs. In particular, solution  $B$  achieves very low inaccuracy, which, in certain contexts, might make it the preferred option. ■

When working with a global criterion for optimization problems, it is customary to normalize the range of the objective values to prevent objectives with larger scales from disproportionately influencing the result. However, in the case of the theory of nescience, normalization is unnecessary. This is because the metrics we use (miscoding, inaccuracy, and surfeit) are already defined within the normalized range  $[0, 1]$ . Consequently, all objectives contribute equally to the evaluation of candidate solutions under the global criterion.

Moreover, another important advantage of the theory of nescience is that all three metrics are inherently commensurable, meaning they are measured using the same units: lengths of computer programs. This shared unit of measurement allows for a coherent and meaningful comparison across the metrics, reinforcing the theoretical consistency and interpretability of the optimization framework.

The following are other metrics that could be used as global criterion to minimize nescience:

- Arithmetic mean:  $\frac{\mu(r) + \iota(d, r) + \sigma(d, r)}{3}$
- Geometric mean:  $(\mu(r) \times \iota(d, r) \times \sigma(d, r))^{1/3}$
- Product:  $\mu(r) \times \iota(d, r) \times \sigma(d, r)$
- Addition:  $\mu(r) + \iota(d, r) + \sigma(d, r)$
- Harmonic mean:  $\frac{3}{\mu(r)^{-1} + \iota(d, r)^{-1} + \sigma(d, r)^{-1}}$

Not all of these metrics qualify as distances, as the geometric mean, product, and harmonic mean do not satisfy the triangle inequality. Moreover, some are not even defined when one of the components is zero, for example, the harmonic mean becomes undefined in such cases. The following paragraphs describe the advantages and disadvantages of each of these metrics.

The Euclidean distance to the ideal vector  $(0, 0, 0)$  offers a geometrically intuitive measure of closeness to perfect knowledge. It treats the three components symmetrically and accounts for their joint magnitude. However, it can bias the solution toward more centrally located points in the objective space and may exclude valid Pareto optimal solutions that lie further from the origin but with highly desirable properties in one metric.

The arithmetic mean provides a simple and interpretable way to aggregate the three metrics, treating each equally. It is widely used due to its mathematical convenience and stability. Nevertheless, it allows compensations (poor performance in one metric can be offset by better performance in others) making it unsuitable when balance among all metrics is crucial.

The geometric mean emphasizes balance by being more sensitive to high values than the arithmetic mean. It discourages solutions that perform poorly in any one metric, making it effective for encouraging uniformly low nescience. However, it is less intuitive and returns zero if any component is zero, which might unfairly suggest perfect knowledge even when other metrics are poor.

The product magnifies the penalty for imbalance: a single large value dominates the result, and a single zero collapses the result to zero. It strongly favors evenly low values across all metrics. While this strictness can be advantageous, it is overly sensitive to small changes, especially near zero, and may mask informative differences in otherwise competitive hypotheses.

The addition is mathematically equivalent to the arithmetic mean (up to scaling) and offers the same interpretability and ease of computation. It is particularly useful when metrics are already normalized, as in the theory of nescience. However, like the mean, it permits trade-offs that might be scientifically undesirable when balance across all three aspects is necessary.

The harmonic mean severely penalizes large values, making it ideal for enforcing balance and avoiding outliers. It is particularly well-suited when the worst-performing metric should drive the overall score. However, it is undefined when any component is zero, and it is more difficult to interpret than other averages.

### 6.2.2 Weighting Method

In the weighting method for solving multi-objective optimization problems, the idea is to associate a weighting coefficient to each objective function and minimize the weighted sum of the objectives. The weighting coefficients  $w_i$  are real numbers such that  $w_i \geq 0$  for all  $i = 1, \dots, k$ . We also require that the weights are normalized, meaning  $\sum_{i=1}^k w_i = 1$ .

According to the weighting method, the science problem is modified into

the following problem:

$$\begin{aligned} & \text{minimize} && w_\mu \mu(r) + w_\iota \iota(d, r) + w_\sigma \sigma(d, r) \\ & \text{subject to} && \mathbf{x} \in \mathbf{S} \end{aligned}$$

where  $w_\mu + w_\iota + w_\sigma = 1$ .

A weighting coefficient of zero is not meaningful, as it would imply that one of the objective functions has no significance whatsoever, an assumption that contradicts the principles of the theory of nescience.

As shown in Proposition F.6.2, the solutions obtained through the weighted method are guaranteed to be Pareto optimal, although this approach does not capture all Pareto optimal solutions. In standard multi-objective optimization, it is customary to normalize the objective functions to prevent those with larger numerical ranges from disproportionately influencing the outcome. In the case of the theory of nescience, as previously discussed in the context of the global criterion, such normalization is unnecessary because the metrics we employ (miscoding, inaccuracy, and surfeit) are intrinsically defined within the normalized interval  $[0, 1]$ .

If the weighting method is applied as an a priori approach, a natural question arises: what do the weighting coefficients actually represent within the context of the theory of nescience? These coefficients are often said to reflect the relative importance of the three objectives (misCoding, inaccuracy, and surfeit). However, the meaning of importance in this context is not well-defined and may lead to ambiguity. Rather than interpreting the weights as measures of absolute importance, it is more accurate to view them as expressing the rate at which the decision maker is willing to trade off one metric against another. That is, the weighting coefficients indicate how much increase in one metric the decision maker is willing to tolerate in exchange for a decrease in another, thereby quantifying their tolerance for imbalances among the three dimensions of nescience.

■ **Example 6.6** Suppose a scientist is studying the structure of a newly discovered biological molecule. The current best hypothesis  $(r, d)$  has a misCoding of  $\mu(r) = 0.2$ , an inaccuracy of  $\iota(d, r) = 0.3$ , and a surfeit of  $\sigma(d, r) = 0.6$ . The scientist is considering the use of the weighting method to guide the selection of the next hypothesis.

If the scientist assigns weights  $w_\mu = 0.1$ ,  $w_\iota = 0.7$ , and  $w_\sigma = 0.2$ , this does not necessarily mean that inaccuracy is more important in some absolute sense. Rather, it reflects the scientist's current willingness to tolerate some misCoding or extra descriptive complexity (surfeit) in exchange for a more accurate model. For instance, the scientist might be aiming for predictive precision in experimental outcomes, even if the underlying representation is suboptimal or the model is longer than ideal.

This weighted preference would steer the optimization process toward hypotheses that reduce inaccuracy, even if they slightly increase miscoding or surfeit. In this way, the weighting coefficients quantify the scientist's subjective trade-offs in light of their practical research goals. ■

Employing the weighting method as an a priori method presumes that the decision maker's underlying value function is or can be approximated by a linear function. It must be noted that altering the weighting vectors linearly does not have to mean that the values of the objective functions also change linearly. It is difficult to control the direction of the solutions by the weighting coefficients. Because  $\mu, \iota, \sigma \in [0, 1]$  but interact non-linearly, the drawbacks of the naive weighting scheme are amplified.

### 6.3 Perfect Knowledge

According to the theory of nescience, the fundamental aim of scientific research is to systematically reduce our ignorance concerning the topics under investigation. This ignorance is quantitatively expressed through the notion of nescience, which captures the combined deficiencies in our current understanding. Scientific progress, within this framework, is evaluated by the degree to which a hypothesis, defined as a pair consisting of a representation and a description, reduces the three interdependent components of miscoding, inaccuracy and surfeit. Perfect knowledge, in this context, is said to be achieved when further reductions in nescience are no longer possible. At this point, the nescience associated with the topic reaches its minimum value, signifying that our understanding is both complete and optimally efficient, and no further scientific improvement can be made.

We must distinguish between two notions of perfect knowledge. The first, ideal perfect knowledge, is a theoretical construct that represents a state in which all three components of nescience are exactly zero. This would require a representation that perfectly encodes the entity, a description that exactly reproduces the representation, and no shorter or simpler description being possible. However, this ideal is generally unattainable in practice due to limitations in representation, modeling, and the nature of the entities under investigation. In contrast, Pareto perfect knowledge, reflects the practical achievable implementation of the concept. It corresponds to hypotheses that lie on the Pareto frontier, where no further reduction in one component of nescience can be made without increasing at least one of the others. Thus, Pareto perfect knowledge represents the best feasible approximation to the ideal, balancing trade-offs among miscoding, inaccuracy, and surfeit.

Next, we formally define the concept of (theoretical) ideal perfect knowledge, which occurs when a hypothesis achieves zero miscoding, zero inaccuracy,

racy, and zero surfeit.

**Definition 6.3.1** Let  $(r, d) \in \mathcal{B}^* \times \mathcal{D}$  be a hypothesis. We say that we have attained *ideal perfect knowledge* for the hypothesis  $(r, d)$  if  $\mu(r) = 0$ ,  $\iota(d, r) = 0$ , and  $\sigma(d, r) = 0$ .

While perfect knowledge can be achieved with respect to a given representation and its description, there is no mathematical or logical procedure to determine with certainty which entity this perfect knowledge pertains to. The mapping between representations and entities is mediated by an oracle, and since oracles are unknown, the identification of the underlying entity remains epistemologically inaccessible. Consequently, although we can verify that a hypothesis satisfies all the formal conditions for perfect knowledge, we cannot be certain of the true nature of the entity being perfectly known.

A consequence of our definition is that perfect knowledge implies randomness, in the sense of incompressible descriptions.

**Proposition 6.3.1** Let  $(r, d) \in \mathcal{B}^* \times \mathcal{D}$  be a hypothesis that yields ideal perfect knowledge. Then  $K(d) = l(d)$ .

*Proof.* Apply Proposition 5.1.2. ■

The common intuition is that a random string conveys no meaningful information, as randomness is often associated with noise or disorder. However, within the framework of the theory of nescience, a random description refers to a description that encodes the maximum amount of information in the smallest possible space—that is, one that contains no redundant elements. Such descriptions are incompressible and thus optimal in terms of descriptive efficiency.

The converse, however, does not generally hold, as the following counterexample demonstrates.

■ **Example 6.7** Aristotelian physics provides an inaccurate description of the physical world, as it makes predictions that are inconsistent with empirical observations, for instance, the claim that planets orbit the Earth. Suppose we take a description of Aristotelian physics and compress it using a standard compression algorithm. The resulting file would be a random description in the sense of having zero redundancy (i.e., it is incompressible). However, despite this lack of redundancy, our nescience would not be zero. That is, our knowledge would not be perfect, since the inaccuracy of the description is not zero. ■

Ideal perfect knowledge is unique in value, but not in representation or description, meaning that although there may be many different hypotheses (i.e., pairs of representations and descriptions) that all yield zero nescience,

they are all equivalent in their informational content, they correspond to the same minimal level of uncertainty or ignorance (i.e., zero). However, these hypotheses can differ in their syntactic form: the representation of the entity or the syntax of the description may vary.

When continued research into a particular topic no longer yields improvements in any of these dimensions—i.e., when no further reduction in miscoding, inaccuracy, or surfeit can be achieved without worsening one of the others—we propose to say that a pareto perfect knowledge has been attained. This does not imply absolute or universal truth, but rather that we have reached the limit of what can be improved given our current symbolic framework. In such a state, the hypothesis lies on the Pareto frontier and corresponds to a point of local optimality, where no further refinement is scientifically justified within the constraints of the theory. The concept of perfect knowledge thus provides a formal stopping criterion for inquiry, signaling that the topic has been fully explored with respect to the goals of scientific understanding as defined by the theory of nescience.

**Definition 6.3.2** Let  $(r, d) \in \mathcal{B}^* \times \mathcal{D}$  be a hypothesis. We say that we have attained *Pareto perfect knowledge* if  $(r, d)$  is Pareto optimal.

When continued research into a particular topic no longer leads to improvement in any of the three components of nescience without worsening at least one of the others, we propose to say that Pareto perfect knowledge has been attained. This state does not imply an absolute or universal truth, but rather marks the boundary of what can be improved within the current symbolic and methodological framework. At this point, the hypothesis lies on the Pareto frontier, corresponding to a condition of local optimality where no further refinement is scientifically justified according to the principles of the theory of nescience. The concept of Pareto perfect knowledge therefore serves as a formal stopping criterion for scientific inquiry, indicating that the topic has been thoroughly explored with respect to the epistemic objectives defined by the theory.

**Definition 6.3.3** Let  $(r, d) \in \mathcal{B}^* \times \mathcal{D}$  be a hypothesis. We say that we have attained *Pareto perfect knowledge* if  $(r, d)$  is Pareto optimal.

It is important to emphasize that reaching Pareto perfect knowledge does not necessarily mean that our total ignorance, as measured by nescience, is zero. However, any attempt to reduce one of the components of nescience would unavoidably increase at least one of the others. In such cases, further scientific inquiry, understood as the pursuit of reduced overall nescience, ceases to be meaningful. Nevertheless, in certain practical contexts, we might still choose to prioritize one dimension over the others. For example, in high-

stakes engineering applications, it may be preferable to reduce inaccuracy even at the expense of greater miscoding or increased surfeit. Such trade-offs are driven by external goals or constraints, rather than the internal logic of the theory of nescience.

## 6.4 Current Best Hypothesis

Scientific discovery is inherently iterative. As we explore a topic, we accumulate a set of hypotheses, each composed of a representation and a description, aimed at approximating the underlying entity with increasing accuracy and efficiency. Within this evolving set of candidate hypotheses, it is both natural and practically useful to identify a preferred hypothesis. We refer to this distinguished hypothesis as the current best hypothesis. It represents the best approximation to perfect knowledge that we have found so far and serves as a concrete benchmark against which scientific progress can be measured.

**Definition 6.4.1** Let  $e \in \mathcal{E}$  be an entity, and let  $\hat{\mathcal{H}}_e$  denote the collection of known hypotheses about  $e$ . We designate a distinguished element of  $\hat{\mathcal{H}}_e$  as our *current best hypothesis*.

Our current best hypothesis should be minimal with respect to at least one of the three components (miscoding, inaccuracy, or surfeit) within the set  $\hat{\mathcal{H}}_e$ . The selection of a particular metric as the basis for minimization may lead to different hypotheses being identified as the current best. Ideally, the current best hypothesis would be minimal with respect to two, or even all three components simultaneously; however, this is not always attainable given the current set of known hypotheses for every entity.

■ **Example 6.8** Suppose we are studying a physical phenomenon such as planetary motion. We consider three competing hypotheses: (1) a geocentric representation with a descriptive model based on epicycles, (2) a heliocentric representation with elliptical orbits, and (3) a relativistic representation with spacetime curvature. Our current best hypothesis would be the third one, as it achieves the lowest nescience given the representations and descriptions we have investigated so far. ■

In some scientific research scenarios, it is common to fix a particular representation, such as the result of an experiment or a dataset composed of measured data, and search only for the description that best models that representation. This motivates the definition of a related but more restricted notion: the best description for a given representation.

**Definition 6.4.2** Let  $r \in \mathcal{R}$  be a representation, and let  $\hat{\mathcal{D}}_r$  denote the collection of known descriptions of  $r$ . We designate a distinguished element of  $\hat{\mathcal{D}}_r$  as our *current best description*.

As in the case of the current best hypothesis, the current best description should be minimal with respect to at least one of the two components, inaccuracy and surfeit, within the set  $\hat{\mathcal{D}}_r$ . For most representations  $r$ , it is generally not possible to minimize both components simultaneously.

While this approach simplifies the search space and is often motivated by convenience or domain conventions, the theory of nescience discourages fixing the representation in advance. This restriction introduces a significant epistemological risk: one may fall into a local minimum of the nescience landscape, where improvements to the description cannot reduce ignorance further, even though a different representation might unlock a much lower nescience value when paired with an appropriate description. A more robust scientific methodology should simultaneously explore alternative representations and descriptions to maximize the chance of converging to a global minimum of nescience.

Excellent. Here's a revised version of your paragraph that includes a practical example more aligned with your preference—where selecting a different subset of attributes, possibly enhanced with external data, leads to a better model:

■ **Example 6.9** Consider a machine learning task to predict customer churn in a telecommunications company. A standard approach might involve using all available customer attributes, such as service usage metrics, demographic information, and past complaint history, as a fixed representation. A model is then trained to minimize prediction error on this data. Suppose that after exhaustive hyperparameter tuning and model selection, the best model still shows significant error. This may indicate that the current representation is suboptimal. A more insightful representation might focus on a carefully chosen subset of attributes, perhaps augmented with external data such as regional economic indicators or customer sentiment extracted from support call transcripts. ■

## 6.5 Unknown Unknown

In Chapter 1, we introduced the concept of the unknown unknown region, an area encompassing problems for which we not only lack solutions but are entirely unaware of their existence. Within the framework of the theory of nescience, one of the central objectives is to develop a systematic procedure for identifying and exploring potential research entities hidden within this

region (refer to Chapter 7 for a detailed procedure on how to approach this task). In this section, we provide a formal characterization of the unknown unknown region and examine its properties.

To uncover what lies hidden within the unknown unknown region, we must first establish the boundary of what is already known. We define this boundary not merely by the existence of hypotheses about a topic, but by the quality of our current knowledge.

**Definition 6.5.1** Let  $e \in \mathcal{E}$  be an entity, and let  $\hat{\mathcal{H}}_e$  denote the collection of known hypotheses about  $e$ . If there exists a hypothesis  $h \in \hat{\mathcal{H}}_e$  that is Pareto optimal we say that  $e$  belongs to the set *known knowns* (denoted by  $\mathcal{E}_{KK}$ ).

This definition identifies as known knowns those entities for which our current best hypotheses are not only available but have achieved a local optimum in the tradeoff between miscoding, inaccuracy, and surfeit. In practice, however, determining the exact set of known known entities is far from straightforward. Assessing whether a given hypothesis lies on the true Pareto frontier is, in general, undecidable. Consequently, our classification of entities as known knowns must be regarded as provisional and subject to revision as our knowledge evolves.

In addition to identifying what we already understand perfectly, we must also recognize the known unknowns, those entities or topics we acknowledge as scientifically relevant but for which our current understanding remains incomplete. These are areas where our ignorance is recognized, and active efforts are being made to reduce it through research.

**Definition 6.5.2** Let  $e \in \mathcal{E}$  be an entity, and let  $\hat{\mathcal{H}}_e$  be the collection of known hypotheses about  $e$ . If  $\hat{\mathcal{H}}_e \neq \emptyset$  and no hypothesis  $h \in \hat{\mathcal{H}}_e$  is Pareto optimal, we say that  $e$  belongs to the set *known unknown* (denoted by  $\mathcal{E}_{KU}$ ).

Known unknowns are entities for which we possess hypotheses, but none of them lie on the Pareto frontier. That is, each known hypothesis can be improved in at least one component of nescience without increasing the others. This implies that our current models are suboptimal and that further scientific progress is achievable. As before, identifying the set of known unknowns in practice is challenging. It requires not only recognizing which entities have been explicitly studied, but also verifying that hypotheses have been formally documented, typically through publications, datasets, or other structured records.

The unknown unknowns correspond to entities for which no hypothesis has ever been proposed. These are aspects of reality that lie entirely beyond

the scope of our current scientific awareness. We are not only ignorant of their true nature, but also unaware of their very existence.

**Definition 6.5.3** Let  $e \in \mathcal{E}$  be an entity, and let  $\hat{\mathcal{H}}_e$  be the collection of known hypotheses about  $e$ . If  $\hat{\mathcal{H}}_e = \emptyset$ , we say that  $e$  belongs to the set *unknown unknown* (denoted by  $\mathcal{E}_{UU}$ ).

In the theory of nescience, all entities for which no hypotheses are currently known are considered part of the realm of the unknown unknown. This includes both entities that may become knowable in the future (given improved representations or descriptions), and those that might remain permanently inaccessible (see Section G.2). The theory of nescience does not distinguish between knowable and unknowable unknowns, as such a distinction is not operationally meaningful within a framework grounded in symbolic representations.

Every knowable entity belongs to exactly one of three epistemic categories: known knowns, known unknowns, or unknown unknowns. The following proposition formalizes this trichotomy.

**Proposition 6.5.1** Let  $\mathcal{E}$  be the set of all entities. Then:

$$\mathcal{E} = \mathcal{E}_{KK} \cup \mathcal{E}_{KU} \cup \mathcal{E}_{UU} \quad \text{and} \quad \mathcal{E}_{KK} \cap \mathcal{E}_{KU} = \mathcal{E}_{KK} \cap \mathcal{E}_{UU} = \mathcal{E}_{KU} \cap \mathcal{E}_{UU} = \emptyset$$

*Proof.* Let  $e \in \mathcal{E}$  be arbitrary. By definition, the hypothesis set  $\hat{\mathcal{H}}_e$  associated with  $e$  is either empty or non-empty. If  $\hat{\mathcal{H}}_e = \emptyset$ , then, by definition,  $e \in \mathcal{E}_{UU}$ . If  $\hat{\mathcal{H}}_e \neq \emptyset$ , then two cases are possible: i) if there exists  $h \in \hat{\mathcal{H}}_e$  such that  $h$  is Pareto optimal, then  $e \in \mathcal{E}_{KK}$ ; ii) if all  $h \in \hat{\mathcal{H}}_e$  are not Pareto optimal, then  $e \in \mathcal{E}_{KU}$ . Therefore, every  $e \in \mathcal{E}$  belongs to at least one of the three subsets.

To show that these subsets are pairwise disjoint, assume by contradiction that an entity  $e$  belongs to both  $\mathcal{E}_{KK}$  and  $\mathcal{E}_{KU}$ . Then  $\hat{\mathcal{H}}_e \neq \emptyset$ , and both,  $\exists h \in \hat{\mathcal{H}}_e$  that is Pareto optimal (by membership in  $\mathcal{E}_{KK}$ ), and  $\forall h \in \hat{\mathcal{H}}_e$ ,  $h$  is not Pareto optimal (by membership in  $\mathcal{E}_{KU}$ ), which is a contradiction.

Similar contradictions arise if  $e$  belongs to both  $\mathcal{E}_{KU}$  and  $\mathcal{E}_{UU}$  or both  $\mathcal{E}_{KK}$  and  $\mathcal{E}_{UU}$ , since  $\mathcal{E}_{UU}$  requires  $\hat{\mathcal{H}}_e = \emptyset$ . ■

We are concerned not only with what we know or do not know, but also with the structural boundaries that separate these domains. The boundary that separates the set of entities for which at least one hypothesis is currently known (i.e., the known) from those for which no hypotheses exist (i.e., the unknown unknown) is referred to as the knowledge frontier.

Before doing so, we must introduce the concept of an anti-Pareto optimal hypothesis, which is one where no component of nescience can be made worse without improving at least one of the others. This notion mirrors the definition of Pareto optimality, but in reverse.

**Definition 6.5.4** Let  $\mathcal{H}_e$  be the collection of hypotheses about an entity  $e \in \mathcal{E}$ . We say that  $(r, d) \in \mathcal{H}_e$  is *anti-Pareto optimal* for  $e$  if, for all  $(r', d') \in \mathcal{H}_e \setminus (r, d)$ , the following holds: if

$$\mu(r') > \mu(r), \quad \iota(d', r') \geq \iota(d, r), \quad \text{and} \quad \sigma(d', r') \geq \sigma(d, r),$$

then at least one of the following is strictly true:

$$\iota(d', r') < \iota(d, r), \quad \text{or} \quad \sigma(d', r') < \sigma(d, r).$$

The condition must also hold symmetrically for permutations involving  $\iota$  and  $\sigma$ .

This definition captures the idea that the hypothesis lies at a point of locally maximal nescience, any attempt to increase ignorance along one dimension (e.g., adding more miscoding) will necessarily force a reduction in ignorance along at least one of the remaining dimensions (e.g., increased accuracy or reduced redundancy).

**Definition 6.5.5** Let  $\mathcal{E}_K$  be the set of known entities. The *knowledge frontier* is defined as the set:

$$\mathcal{F} = \{e \in \mathcal{E}_K \mid \text{the current best hypothesis is anti-Pareto optimal}\}.$$

This definition situates the knowledge frontier at the outermost edge of our current understanding. These are the entities for which: (i) at least one hypothesis is known; (ii) the best-known hypothesis is anti-Pareto optimal, meaning that any further degradation in one dimension of nescience necessarily leads to an improvement in at least one of the others; and (iii) no meaningful scientific degradation is possible without paradoxically improving the hypothesis. This configuration marks the conceptual boundary between the barely known and the unexplored.

A new research entity is any entity that lies beyond the current knowledge frontier. That is, it is an entity for which no hypothesis has yet been formulated, recorded, or studied. These entities exist within the realm of the unknown unknown, not only do we lack understanding of them, but we are not even aware of their existence. Identifying such entities represents the initial and most foundational step in expanding scientific knowledge, since they mark the points at which our current understanding fails to even formulate a question.

**Definition 6.5.6** Let  $\mathcal{E}$  denote the set of all knowable entities, and let  $\hat{\mathcal{H}}_e$  be the collection of known hypotheses about an entity  $e \in \mathcal{E}$ . We say that  $e$  is a *new research entity* if  $\hat{\mathcal{H}}_e = \emptyset$ .

New research entities are, by definition, members of the unknown unknown region of the knowledge space. They have not been explicitly represented or described, and no attempt has yet been made to understand them through a scientific hypothesis. The identification and formalization of such entities is central to the methodology proposed by the theory of nescience: only by expanding the set of known entities can we ensure the continuous reduction of global ignorance.

A new research entity refer to those entities for which no hypothesis has yet been proposed. By definition, such entities fall into the class of unknown unknowns. The following proposition formalizes this relationship.

**Proposition 6.5.2** Let  $\mathcal{E}_{NRE} \subseteq \mathcal{E}$  be the set of all *new research entities*, defined as entities  $e \in \mathcal{E}$  such that  $\hat{\mathcal{H}}_e = \emptyset$  and  $e$  has not yet been discovered or studied. Let  $\mathcal{E}_{UU} \subseteq \mathcal{E}$  be the set of all *unknown unknowns*, defined by  $\mathcal{E}_{UU} = \{e \in \mathcal{E} \mid \hat{\mathcal{H}}_e = \emptyset\}$ . Then,  $\mathcal{E}_{NRE} \subseteq \mathcal{E}_{UU}$ .

*Proof.* Let  $e \in \mathcal{E}_{NRE}$  be an arbitrary new research entity. By the definition of new research entity, we have  $\hat{\mathcal{H}}_e = \emptyset$ . But this condition is precisely the definition of membership in  $\mathcal{E}_{UU}$ . Therefore,  $e \in \mathcal{E}_{UU}$ . Since every element of  $\mathcal{E}_{NRE}$  is also in  $\mathcal{E}_{UU}$ , it follows that:  $\mathcal{E}_{NRE} \subseteq \mathcal{E}_{UU}$ . ■

This result captures the foundational idea that research innovation arises from exploring the unknown unknown region. By definition, any new scientific entity must originate from an epistemically uncharted area, one where no hypotheses currently exist.

## 6.6 Science vs. Pseudoscience

In the theory of nescience, scientific progress is measured by the systematic reduction of ignorance. As we propose and validate hypotheses about previously unstudied entities, those entities transition from the unknown unknown region to either the known known or known unknown regions. From this perspective, the set of unknown entities should decrease over time.

**Proposition 6.6.1** Let  $\mathcal{E}_{UU}^{(t)}$  denote the set of unknown unknown entities at time  $t$  and  $\mathcal{E}_{UU}^{(t+1)}$  at a posterior time. Then,  $\mathcal{E}_{UU}^{(t+1)} \subseteq \mathcal{E}_{UU}^{(t)}$  for all  $t \in \mathbb{N}$ .

*Proof.* At each time step  $t$ , let the process of scientific discovery introduce new hypotheses about entities in  $\mathcal{E}$ , thereby moving those entities from the unknown unknown region into the known known region (either known unknowns or known known). ■

In addition to the reduction in the cardinality of the unknown unknown region, we can also observe a progressive reduction in the nescience of

individual entities as more refined hypotheses are introduced. Recall that the nescience of an entity is defined based on its current best hypothesis.

**Proposition 6.6.2** Let  $e \in \mathcal{E}$  be an entity such that  $\hat{\mathcal{H}}_e^{(t)}$  and  $\hat{\mathcal{H}}_e^{(t+1)}$  are the collections of known hypotheses about  $e$  at times  $t$  and  $t+1$ , respectively. Then, if a new hypothesis  $h' \in \hat{\mathcal{H}}_e^{(t+1)} \setminus \hat{\mathcal{H}}_e^{(t)}$  is such that  $v(h') < v(h)$  for all  $h \in \hat{\mathcal{H}}_e^{(t)}$ , the nescience of  $e$  satisfies:

$$v^{(t+1)}(e) < v^{(t)}(e),$$

where  $v^{(t)}(e)$  denotes the nescience of the current best hypothesis at time  $t$ .

*Proof.* By definition, we have that  $v^{(t)}(e) = \min_{h \in \hat{\mathcal{H}}_e^{(t)}} v(h)$  and  $v^{(t+1)}(e) = \min_{h \in \hat{\mathcal{H}}_e^{(t+1)}} v(h)$ . If a new hypothesis  $h' \in \hat{\mathcal{H}}_e^{(t+1)}$  is added such that  $v(h') < v^{(t)}(e)$ , then  $v^{(t+1)}(e) = v(h') < v^{(t)}(e)$ . ■

This result reflects the iterative nature of scientific inquiry: as we formulate new hypotheses or improve existing ones, the combined errors associated with our representations and descriptions tend to diminish. Over time, and under continued research, the nescience of most entities should asymptotically approach its minimum, potentially reaching the Pareto frontier, or in rare cases, even ideal perfect knowledge.

This characteristic trend of decreasing nescience offers a powerful criterion for distinguishing scientific disciplines from pseudoscientific ones. In genuine scientific fields, the average nescience of topics demonstrably declines over time as knowledge improves. In contrast, pseudoscientific domains typically exhibit stagnation in nescience: either the number of proposed hypotheses remains static or their quality fails to improve, leading to no measurable reduction in ignorance. Thus, the presence or absence of a decreasing trend in nescience provides a formal, quantitative indicator for demarcating science from pseudoscience.

## 6.7 Nescience of Areas

As discussed in Section 2.8, entities can be grouped into research areas, defined as subsets  $\mathcal{A} \subset \mathcal{E}$  of the overall entity space. The concept of a research area is meaningful when the entities in  $\mathcal{A}$  share a common epistemic domain or possess a unifying property, such as belonging to the same scientific field or addressing a related phenomenon. To quantify how much we do not know about a research area  $\mathcal{A}$ , we must work with the subset of entities within  $\mathcal{A}$  for which at least one hypothesis is currently known  $\hat{\mathcal{A}} \subseteq \mathcal{A}$ . A representation

of the area  $\mathcal{A}$  is then a symbolic encoding  $R_{\hat{\mathcal{A}}}$  of the known entities in  $\hat{\mathcal{A}}$ , suitable for further descriptive analysis. Given this representation, we define a description of the area  $\mathcal{A}$  as a string  $d_{\hat{\mathcal{A}}}$  that models or summarizes the structure or content of  $R_{\hat{\mathcal{A}}}$ .

The concept of nescience can be extended from individual entities to entire research areas, allowing us to quantitatively assess how well a given area is understood. This extension relies on generalizing the core components of nescience to sets of entities. We use the notions of combined miscoding of the known subset (see Section 3.5), combined inaccuracy (see Section 4.5), and combined surfeit (see Section 5.6) to define a composite measure of nescience at the area level.

In practice, due to the difficulty of precisely identifying the entire known subset  $\hat{\mathcal{A}} \subseteq \mathcal{A}$ , it is often convenient to use approximations. One practical and interpretable approximation is the notion of average nescience, which provides a summary of the epistemic quality of the known entities within the area.

**Definition 6.7.1** Let  $\mathcal{A} \subset \mathcal{E}$  be an area, and  $\hat{\mathcal{A}}$  be the subset of known entities. The *average nescience of the area* is defined as:

$$\bar{v}(\mathcal{A}) = \frac{1}{|\hat{\mathcal{A}}|} \sum_{e \in \hat{\mathcal{A}}} v(e),$$

where  $v(e)$  denotes the nescience of the current best hypothesis about entity  $e$  given a particular decision maker.

For instance, when dealing with research topics, a research area may correspond to a broad disciplinary domain, such as biology, mathematics, or physics, each comprising a collection of entities that fall under that domain. In this context, the area of biology would include all research topics formally classified within the biological sciences. This classification allows us to compute and compare the nescience of distinct knowledge domains by aggregating the nescience of the individual topics they encompass (see Section 9.3). Such comparisons provide insight into which areas of human knowledge remain most poorly understood and where future research efforts might be most impactful.

## References

Here are some selected references that are particularly relevant to the concept nescience, along with brief explanations of their relevance.

[LV13] is a foundational text in algorithmic information theory, central to defining concepts such as description length, randomness, and information

content. These are directly used in our theory of nescience to formalize inaccuracy and surfeit. [Cha69] provides key ideas on the relationship between program length and algorithmic simplicity. This work contributes to the formal underpinnings of inaccuracy and model complexity in the theory of nescience. [CT12] A comprehensive treatment of classical information theory, including entropy, mutual information, and coding theorems. These concepts are foundational for measuring redundancy and assessing the informativeness of representations in our framework. [Grü07] is directly relevant to the concept of surfeit in nescience. The MDL principle formalizes model selection based on description length, which parallels the idea of seeking non-redundant descriptions. [Sol64] Solomonoff's work on algorithmic probability underlies the notion of model uncertainty and prediction, both of which are implicitly addressed in our theory's pursuit of minimal ignorance. [Kol65] lays the groundwork for the use of Kolmogorov complexity as a measure of information, randomness, and compressibility, all essential in quantifying nescience. [Tur36] provides the foundational model of computation used throughout the theory of nescience. The theory's reliance on representations and oracle machines assumes the basic framework established by Turing. [Cal02] explores algorithmic complexity and randomness in depth, offering a broader perspective that connects well with our treatment of knowledge, representation, and incompleteness in the theory of nescience. [Cha95] illustrates limits of formal descriptions and links to incompleteness phenomena. This is relevant for understanding the boundaries of compressibility and the epistemological challenges in defining knowledge. [Mie12] is crucial for our use of Pareto and anti-Pareto frontiers in describing tradeoffs among miscoding, inaccuracy, and surfeit. Provides the theoretical tools for reasoning about optimality in multi-dimensional knowledge spaces. [Pop14] while not technical, this work provides a philosophical background on how scientific knowledge evolves through falsification, a process aligned with your model's notion of reducing nescience via hypothesis revision. [Sup02] addresses the problem of how scientific structures relate to the entities they describe, an issue directly relevant to your notion of representations, miscoding, and scientific models.



## 7. Interesting Questions

*It is not the answer that enlightens,  
but the question.*  
Eugène Ionesco

In this chapter, we introduce a set of metrics for classifying research topics according to their potential to generate interesting problems, along with a methodology for the assisted discovery of new research questions. The objective is to propose new directions, novel research ideas, that contribute to reducing nescience, that is, to diminishing the extent of the scientific unknown. The proposed methodology supports both the identification of new applications for existing tools to address open problems (the known unknown) and the discovery of entirely new and previously unexplored research directions (the unknown unknown). While the methodology is applicable to both intradisciplinary and interdisciplinary topics, the most impactful results typically arise in the latter case. In Chapter 10, we demonstrate the methodology in practice and propose several new questions and research topics.

We have already examined three dimensions for classifying research topics: miscoding (Chapter 3), inaccuracy (Chapter 4), and surfeit (Chapter 5). These metrics allow us to quantitatively assess our level of understanding

of a topic, a broader concept we refer to as nescience. In this section, we introduce two additional metrics for characterizing topics: relevance (Section ??) and applicability (Section 7.2). Relevance measures the impact a topic has on people's lives and complements the existing dimensions used to quantify nescience. Applicability quantifies how frequently a topic has been applied in other domains and helps identify new uses for existing technologies.

What is proposed in this chapter is an algebraic approach to the assisted discovery of potentially interesting research questions, grounded in the theory of nescience. On the one hand, it aims to support researchers in their day-to-day work. The methodology can be used to uncover novel tools that may be applied to a given problem, or to identify new problems where existing tools could be effectively used. In its more advanced form, the methodology facilitates the exploration of the unknown unknown, that is, research areas that have not yet been conceptualized, described, or even imagined.

On the other hand, because the methodology is based on well-defined mathematical principles, it lends itself to automation. This opens the door for artificial intelligence systems to move beyond their current limitations in autonomously generate truly novel research directions. By formalizing the process of question discovery, the methodology enables AI to propose interesting and previously unexplored research questions, and even to discover entirely new topics of scientific inquiry.

## 7.1 Integrative Depth

In addition of the metrics that compose nescience, further metrics are required in order to characterize the role that different research topics play within the scientific knowledge. Such metrics are useful not only for classifying individual topics, but also for identifying promising directions for inquiry (Section 7.5), and guiding the combination of topics in the discovery of new research questions (Section 7.6).

A new metric of *integrative depth* is introduced to capture the multiplicity of nescience-reducing relation among topics. Intuitively, some research topics admit a substantial reduction of nescience by conditioning on a small number of other topics, whereas others admit reductions along many distinct directions. In the latter case, there exist multiple background knowledge bases such that conditioning on each of them yields a decrease in the nescience of the topic, even though none of these backgrounds is individually necessary.

**Definition 7.1.1** Given the nescience dependency graph  $NDG = (\mathcal{T}, E_N)$  we define the *integrative depth* of a research topic  $t \in \mathcal{T}$  as its indegree in  $NDG$ , that is,

$$ID(t) = \text{indeg}(t)$$

Recall that a directed edge from B to A in the nescience dependency graph indicates that conditioning on the knowledge background associated with B reduces the nescience of A. This does not imply that such background knowledge is necessary to understand A, nor that it is the only background capable of reducing its nescience.

Integrative depth measures the extent to which understanding a research topic requires the integration of knowledge originating from other topics. A topic with low integrative depth can be substantially understood in relative isolation, whereas a topic with high integrative depth might require to understand many other independent sources of knowledge.

In practice, topics with high integrative depth often benefit from combining insights drawn from several contributing areas (see Section XXX), although the existence of multiple nescience-reducing directions does not, by itself, imply that such combination is necessary.

■ **Example 7.1** Consider the research topic of cancer. Conditioning on knowledge from genetics reduces nescience by clarifying the role of mutations and oncogenes, while conditioning on cell biology reduces nescience by explaining mechanisms of uncontrolled proliferation. Knowledge from immunology further reduces nescience by accounting for immune surveillance and evasion, and insights from tissue mechanics and the tumor microenvironment contribute by explaining how physical and structural factors influence tumor growth. Each of these background topics independently yields a reduction in the nescience associated with cancer, even though none of them is individually necessary or sufficient to achieve a comprehensive understanding. The high integrative depth of this topic therefore reflects the existence of multiple distinct directions along which ignorance can be reduced, rather than a single required sequence of prerequisites. ■

The integrative depth metric inherits several basic properties directly from its graph-theoretic definition: *i*) for any research topic  $t \in \mathcal{T}$ , the integrative depth  $D(t)$  is a non-negative integer; *ii*) a research topic  $t$  has integrative depth  $D(t) = 0$  if and only if there is no other topic whose representations reduce the nescience of  $t$ .

Topics with zero integrative depth are autonomous in the sense that their nescience cannot be reduced by conditioning on other topics alone.

Integrative depth is independent of the current level of nescience of a

topic. A topic may have high integrative depth and low nescience, corresponding to a well-understood domain that nonetheless requires the integration of many different areas. Conversely, a topic may have high integrative depth and high nescience, indicating a complex, unresolved problem whose understanding depends on progress across multiple topics.

### Weighted integrative depth

In some cases, it is not sufficient to know how many topics contribute to reducing the nescience of a given topic; it is also important to quantify *how much* each contributing topic reduces nescience. For this purpose, we consider the weighted version of the nescience dependency graph.

**Definition 7.1.2** Given the weighted nescience dependency graph  $WNDG = (\mathcal{T}, E_N, w)$  we define the *weighted integrative depth* of a research topic  $t \in \mathcal{T}$  as

$$D_w(t) = \sum_{A:(A,t) \in E_N} w(A, t) = \sum_{A:(A,t) \in E_N} (N(t | A) - N(t)).$$

The weighted integrative depth of a topic measures the total reduction in nescience obtained by integrating all topics that contribute to its understanding.

While integrative depth counts the number of distinct topics whose representations reduce the nescience of a given topic, weighted integrative depth captures the overall magnitude of that reduction. A topic may have a high integrative depth but a modest weighted integrative depth if each contributing topic yields only a small reduction in nescience. Conversely, a topic with moderate integrative depth may exhibit a large weighted integrative depth if a few incoming topics provide substantial explanatory power.

Weighted integrative depth therefore quantifies the *strength of integration* required to understand a topic, complementing the purely structural information provided by integrative depth.

■ **Example 7.2** Consider again the entity corresponding to *climate change*. Its integrative depth is high, since many topics contribute to reducing its nescience. However, the magnitude of these contributions may vary substantially. Conditioning on atmospheric physics may yield a large reduction in nescience, while conditioning on a narrowly scoped economic model may yield a smaller one. The weighted integrative depth aggregates these heterogeneous contributions into a single measure of total epistemic gain. ■

The weighted integrative depth satisfies several immediate properties: *i)* For any research topic  $t \in \mathcal{T}$ , the weighted integrative depth  $D_w(t)$  is a non-negative real number. **Add another property derived from graph theory.**

Introduce this proposition

**Proposition 7.1.1** A research topic  $t$  has weighted integrative depth  $D_w(t) = 0$  if and only if it has integrative depth  $D(t) = 0$ .

*Proof.* If  $D(t) = 0$ , there are no incoming edges, and the defining sum for  $D_w(t)$  is empty, yielding  $D_w(t) = 0$ . Conversely, if  $D_w(t) = 0$ , there can be no incoming edges with positive weight, and hence  $D(t) = 0$ . ■

Integrative depth and weighted integrative depth capture complementary aspects of epistemic integration. Integrative depth reflects the number of distinct sources that must be combined to reduce the nescience of a topic, while weighted integrative depth reflects the total magnitude of the reduction achieved by those sources. Together, these metrics distinguish between topics that depend on many weakly informative contributors and those that depend on a smaller number of highly informative ones.

In later chapters, weighted integrative depth will be used to refine the classification of research topics and to guide the construction of composite research directions that balance epistemic feasibility with the potential for substantial nescience reduction.

### Normalized weighted integrative depth

In many practical situations, we are not interested in the full set of research topics  $\mathcal{T}$ , but rather in a restricted collection  $S \subseteq \mathcal{T}$  corresponding, for instance, to a discipline, a research program, or the scope of a particular organization. In this case, it is natural to evaluate integrative metrics relative to the weighted nescience dependency graph restricted to  $S$ .

Assume that the restriction of the weighted nescience dependency graph to a subset  $S$  has been defined elsewhere in the book. We denote the resulting restricted weighted graph by

$$G_{N,S}^w = (S, E_{N,S}, w),$$

where  $E_{N,S} \subseteq S \times S$  is the set of directed edges between topics in  $S$ , and each edge  $(A, t) \in E_{N,S}$  satisfies  $w(A, t) = N(t | A) - N(t) > 0$ .

**Definition 7.1.3** Let  $S \subseteq \mathcal{T}$  and let  $G_{N,S}^w = (S, E_{N,S}, w)$  be the restricted weighted nescience dependency graph. The *weighted integrative depth of  $t$  restricted to  $S$*  is defined as

$$D_{w,S}(t) = \sum_{A: (A, t) \in E_{N,S}} w(A, t), \quad t \in S.$$

This quantity measures the total reduction in nescience of  $t$  obtained by integrating only those topics that lie within  $S$ .

Since the absolute scale of  $D_{w,S}(t)$  depends on the size and composition of  $S$ , it is often useful to normalize the weighted integrative depth within  $S$  so that values lie in the unit interval and are comparable across subsets.

**Definition 7.1.4** Let  $S \subseteq \mathcal{T}$  be finite. The *min–max normalized weighted integrative depth on  $S$*  is defined for each  $t \in S$  by

$$\overline{D}_{w,S}(t) = \frac{D_{w,S}(t) - \min_{u \in S} D_{w,S}(u)}{\max_{u \in S} D_{w,S}(u) - \min_{u \in S} D_{w,S}(u)}.$$

In the degenerate case where  $\max_{u \in S} D_{w,S}(u) = \min_{u \in S} D_{w,S}(u)$ , all normalized values are set to 0.

**Proposition 7.1.2** Let  $S \subseteq \mathcal{T}$  be finite. For any topic  $t \in S$ , the restricted weighted integrative depth satisfies  $D_{w,S}(t) \geq 0$ .

*Proof.* By construction, each edge in  $E_{N,S}$  has strictly positive weight  $w(A, t) > 0$ . Therefore,  $D_{w,S}(t)$  is a sum of non-negative terms, and hence non-negative. ■

**Proposition 7.1.3** Let  $S \subseteq \mathcal{T}$  be finite. For any topic  $t \in S$ , the normalized weighted integrative depth satisfies

$$0 \leq \overline{D}_{w,S}(t) \leq 1.$$

Moreover,  $\overline{D}_{w,S}(t) = 0$  for at least one topic in  $S$ , and  $\overline{D}_{w,S}(t) = 1$  for at least one topic in  $S$ , whenever the denominator is non-zero.

*Proof.* If the denominator is non-zero, then  $\min_{u \in S} D_{w,S}(u) \leq D_{w,S}(t) \leq \max_{u \in S} D_{w,S}(u)$ , and the stated bounds follow immediately. The extreme values are attained by any minimizer and maximizer of  $D_{w,S}$  on  $S$ . ■

**Proposition 7.1.4** Let  $S \subseteq \mathcal{T}$  be finite and let  $a > 0$  and  $b \in \mathbb{R}$ . Define a rescaled depth

$$D'_{w,S}(t) = aD_{w,S}(t) + b, \quad t \in S.$$

If  $\max_{u \in S} D_{w,S}(u) \neq \min_{u \in S} D_{w,S}(u)$ , then the min–max normalized values are invariant under this transformation:

$$\overline{D}'_{w,S}(t) = \overline{D}_{w,S}(t) \quad \text{for all } t \in S.$$

*Proof.* Substituting  $D'_{w,S}(t) = aD_{w,S}(t) + b$  into the min–max normalization formula shows that the factor  $a$  cancels and the shift  $b$  subtracts out in numerator and denominator. ■

**Proposition 7.1.5** Let  $S \subseteq \mathcal{T}$  be finite and let  $t \in S$ . If new incoming edges pointing to  $t$  are added to the restricted weighted graph  $G_{N,S}^w$  (or if the weight of an existing incoming edge is increased), then  $D_{w,S}(t)$  can only increase. Consequently, if the minimum and maximum of  $D_{w,S}$  over  $S$  remain unchanged, then  $\bar{D}_{w,S}(t)$  can only increase.

*Proof.* Each new incoming edge contributes an additional positive term to the sum defining  $D_{w,S}(t)$ , and increasing an existing incoming weight increases that sum. If  $\min_{u \in S} D_{w,S}(u)$  and  $\max_{u \in S} D_{w,S}(u)$  remain fixed, then the normalization is an increasing affine transformation of  $D_{w,S}(t)$ , and the same monotonicity carries over to  $\bar{D}_{w,S}(t)$ . ■

The restricted quantity  $D_{w,S}(t)$  measures the degree to which a topic  $t$  can be clarified by integrating knowledge *available within* the subset  $S$ . This is particularly important when  $S$  represents the effective scope of a research team or organization: a topic may have large global weighted integrative depth, but small restricted weighted integrative depth if most of its supporting topics lie outside the subset.

The normalized score  $\bar{D}_{w,S}(t)$  provides a dimensionless ranking of topics within  $S$ , enabling comparisons across different subsets and supporting portfolio-level analyses in later chapters.

## 7.2 Applicability

In addition to nescience and integrative depth, it is useful to characterize the extent to which a research topic can serve as a tool for reducing ignorance in other topics. Some topics play a primarily enabling role: conditioning on them yields reductions in nescience across a broad range of other topics. The metric of *applicability* is introduced to capture this structural property.

**Definition 7.2.1** Let  $NDG = (\mathcal{T}, E_N)$  be the nescience dependency graph. We define the *applicability* of a research topic  $t \in \mathcal{T}$  as its out-degree in  $NDG$ , that is,

$$A(t) = \text{outdeg}(t)$$

Applicability measures the breadth of the nescience-reducing influence of a research topic. A topic with low applicability contributes to the reduction of nescience in few other topics, whereas a topic with high applicability admits many distinct directions along which it can be used to reduce ignorance elsewhere.

■ **Example 7.3** As an example of a topic with high applicability, consider *linear algebra*. Conditioning on linear algebra reduces the nescience of

a wide range of topics, including numerical analysis, machine learning, quantum mechanics, signal processing, and control theory. In the nescience dependency graph, this corresponds to many outgoing arcs, and hence a high applicability.

By contrast, a narrowly scoped descriptive topic, such as the *maximum recorded jump distance of a specific flea species*, may reduce the nescience of few or no other topics. Such a topic therefore exhibits low applicability. ■

Applicability inherits several immediate properties from its graph-theoretic definition: *i*) for any research topic  $t \in \mathcal{T}$ , the applicability  $A(t)$  is a non-negative integer; *ii*) A research topic  $t$  has applicability  $A(t) = 0$  if and only if conditioning on  $t$  does not reduce the nescience of any other topic.

### Weighted applicability

**Definition 7.2.2** Let  $G_N^w = (\mathcal{T}, E_N, w)$  be the weighted nescience dependency graph. The *weighted applicability* of a research topic  $t \in \mathcal{T}$  is defined as

$$A_w(t) = \sum_{u:(t,u) \in E_N} w(t,u).$$

Weighted applicability measures not only how many topics benefit from conditioning on  $t$ , but also the total magnitude of the nescience reductions it provides.

### Normalized applicability

To facilitate comparison across topics, applicability values can be normalized using min-max normalization.

**Definition 7.2.3** Let  $\mathcal{T}$  be finite. The *min–max normalized applicability* of a topic  $t \in \mathcal{T}$  is defined as

$$\bar{A}(t) = \frac{A(t) - \min_{u \in \mathcal{T}} A(u)}{\max_{u \in \mathcal{T}} A(u) - \min_{u \in \mathcal{T}} A(u)}.$$

In the degenerate case where all topics have the same applicability, all normalized values are set to 0.

An analogous normalization can be applied to weighted applicability by replacing  $A(t)$  with  $A_w(t)$ .

Applicability and integrative depth capture complementary structural aspects of research topics within the nescience dependency graph. Integrative depth characterizes the plurality of directions along which the nescience of a topic can be reduced, whereas applicability characterizes the plurality of directions along which a topic can reduce the nescience of others.

In later chapters, these metrics will be used jointly to distinguish between research topics that primarily function as tools and those that primarily function as problems, and to guide the construction of research portfolios that balance enabling power with epistemic challenge.

### 7.3 Maturity

Given an unknown topic  $A$ , a natural strategy for reducing its nescience is to rely on background topics  $B$  such that conditioning on  $B$  yields a reduction in nescience, that is,

$$N(A | B) < N(A).$$

However, not all background topics satisfying this condition are equally suitable to be used as tools. In particular, the usefulness of a background topic depends not only on how much it reduces the nescience of  $A$ , but also on how well understood the background topic itself is.

To capture this trade-off, we adopt the following guiding principle: a background topic  $B$  is considered preferable for reducing the nescience of  $A$  when the combined unresolved nescience associated with using  $B$  is smaller than the original nescience of  $A$ . Formally, this corresponds to requiring that

$$N(A | B) + N(B) < N(A).$$

This condition expresses that the reduction in nescience obtained by conditioning on  $B$  is large enough to compensate for the remaining nescience of  $B$  itself.

This principle is particularly relevant in situations where errors propagate strongly, commitments are irreversible, or independent means of verification are unavailable. In such contexts, relying on background knowledge that is itself poorly understood may introduce additional uncertainty that outweighs the apparent reduction in nescience of the primary topic.

At the same time, the condition above does not exclude the use of poorly understood background topics in all circumstances. In particular, reliance on such topics may be justified when no better-understood alternative is known to yield a reduction in the nescience of the target topic, or when errors are inexpensive and results can be independently verified. In these situations, even a background topic with high nescience may be used as a provisional tool, provided that it contributes to lowering the nescience of the primary problem.

■ **Example 7.4** Large language models illustrate this situation. Conditioning on such models often produces substantial reductions in the nescience

of a wide range of tasks, including text generation, code synthesis, and exploratory problem solving. Although these models are themselves only partially understood, they are typically employed in settings where alternative well-understood tools are unavailable, errors are inexpensive, and outputs can be checked or corrected by external means. Under these conditions, reliance on such models can be epistemically productive despite their limited maturity. ■

To formalize the notion of how well a topic is understood, we introduce the concept of *maturity*. Maturity is defined as a simple reparameterization of nescience, emphasizing its inverse interpretation.

**Definition 7.3.1** Let  $t \in \mathcal{T}$  be a topic, and let  $N(t)$  denote its nescience. The *maturity* of  $t$ , denoted  $M(t)$ , is defined as

$$M(t) = 1 - N(t).$$

A higher maturity value indicates that a topic is better understood and therefore more suitable to be used as a background tool for reducing the nescience of other topics. Conversely, topics with low maturity should be used with caution, since their own unresolved nescience contributes to the total uncertainty associated with their application.

■ **Example 7.5** Linear regression provides an example of a highly mature research topic. Its theoretical assumptions, limitations, and behavior are well characterized, resulting in a low nescience and a correspondingly high maturity. When used as background knowledge, linear regression typically satisfies the condition  $N(A | B) + N(B) < N(A)$  for a wide range of applied problems, making it a reliable tool in settings where stability and interpretability are important. ■

Maturity is an intrinsic property of a research topic, reflecting the extent to which its representations are complete and reliable. It is independent of the topic's applicability or integrative depth, and should therefore not be interpreted as a measure of usefulness or importance. Rather, maturity provides a measure of epistemic stability: mature topics tend to behave predictably when used as background knowledge, whereas immature topics introduce additional uncertainty that must be managed explicitly.

In practice, when comparing maturity values across the topics of a given research area, it is often convenient to work with a normalized scale.

**Definition 7.3.2** Given a finite set of topics  $\mathcal{T}$ , the *min–max normalized*

maturity of a topic  $t \in \mathcal{T}$ , denoted  $\tilde{M}(t)$ , is defined as

$$\tilde{M}(t) = \frac{M(t) - \min_{t' \in \mathcal{T}} M(t')}{\max_{t' \in \mathcal{T}} M(t') - \min_{t' \in \mathcal{T}} M(t')},$$

where  $M(t)$  is the maturity of  $t$  as in Definition 7.3.1.

In the degenerate case where all topics have the same maturity, all normalized values are set to 0. Normalized maturity values will be used in later chapters to guide the selection of background knowledge and to balance nescience reduction against the reliability of the tools employed.

## 7.4 Interestingness

In many scientific and practical contexts it is necessary to compare research topics and to decide which ones are more promising to pursue, either as tools for solving other problems or as problems worthy of investigation in their own right. In this section, we introduce the notion of *interestingness* as a decision-oriented concept derived from the metrics previously introduced.

The theory of nescience naturally induces several dimensions along which topics may be evaluated, but it does not prescribe a unique way of aggregating them into a single scalar score. As a result, interestingness should be understood as arising from a multi-objective optimization problem (see Section F.6), together with a choice of decision rule.

A research topic may be considered interesting as a tool when it can be reliably used to reduce the nescience of other topics. Two theory-derived quantities are relevant in this context: maturity, which measures how well the topic itself is understood and applicability, which measures how widely the topic has been successfully used to reduce nescience in other topics.

**Definition 7.4.1** We define the *interestingness as a tool* of a topic  $t$  as the multi-objective optimization problem:

$$\text{maximize } (M(t), A(t)),$$

where  $M(t)$  denotes the maturity of the topic  $t$  and  $A(t)$  its applicability.

This formulation induces a partial order on topics: a topic  $t_1$  dominates another topic  $t_2$  if  $M(t_1) \geq M(t_2)$  and  $A(t_1) \geq A(t_2)$ , with at least one inequality strict. The set of non-dominated topics forms the pareto frontier of interesting tools.

A research topic may also be interesting as a problem, independently of its usefulness as a tool. In this case, two different theory-internal quantities play a central role: nescience, which measures the extent to which the topic

is not yet understood, and integrative depth, which measures the number of distinct background topics whose knowledge reduces the nescience of the topic. Together, these quantities characterize both the magnitude of unresolved ignorance and the structural richness of the ways in which that ignorance can be reduced.

**Definition 7.4.2** We define the *interestingness as a problem* of a topic  $t$  as the multi-objective optimization problem

$$\text{maximize } (N(t), D(t)),$$

where  $N(t)$  is the nescience of topic  $t$  and  $D(t)$  its integrative depth.

As in the case of tools, this formulation induces a partial order. Topics with high nescience but low integrative depth correspond to forms of ignorance that admit few known avenues for progress, whereas topics with both high nescience and high integrative depth admit many distinct nescience-reducing directions and are structurally richer as research problems.

In practical applications it is often necessary to produce a total ordering of topics. This requires the introduction of a decision maker, that is, a scalarization of the underlying multi-objective problem.

One possible choice is a balanced *Euclidean scalarization*. For example, the interestingness of a topic as a tool may be defined as

$$IT(t) = \frac{\sqrt{\tilde{M}(t)^2 + \tilde{A}(t)^2}}{\sqrt{2}},$$

where  $\tilde{M}(t)$  and  $\tilde{A}(t)$  are normalized versions of maturity and applicability. Analogously, interestingness as a problem may be defined as

$$IP(t) = \frac{\sqrt{\tilde{N}(t)^2 + \tilde{D}(t)^2}}{\sqrt{2}},$$

where  $\tilde{N}(t)$  and  $\tilde{D}(t)$  denote normalized nescience and integrative depth.

These metrics correspond to a particular decision method that favors balanced topics over extreme ones. Other decision makers may be more appropriate in different contexts. For instance, a *minimum-based* decision rule,  $\min\{\tilde{M}(t), \tilde{A}(t)\}$ , emphasizes robustness by penalizing weaknesses in any dimension, meanwhile a *weighted sum* allows explicit prioritization of maturity versus applicability, or nescience versus integrative depth, according to external constraints or strategic goals.

Such decision makers do not alter the underlying theory, but provide practical means of selecting and ranking topics for specific purposes.

## 7.5 Interesting Questions

In the theory of nescience we distinguish between known unknowns, that is, problems whose existence is recognized but whose solutions are not known, and unknown unknowns, corresponding to problems that have not yet been identified. In this section we focus on the former, and introduce a formal notion of *interesting questions* aimed at supporting the systematic exploration of known unknowns.

A central observation underlying this approach is that new research questions often arise when existing knowledge is brought to bear on unresolved problems. In the theory of nescience, such situations are naturally captured by the *nescience dependency graph*, which encodes when knowledge of one topic reduces the nescience of another.

**Definition 7.5.1** Let  $G_N = (\mathcal{T}, E_N)$  denote the nescience dependency graph. We define *interesting question* as an ordered pair of topics  $Q_{t \rightarrow p} = (t, p)$  such that  $(t, p) \in E_N$ .

In other words, a question exists only when the theory predicts that the topic  $t$  can be meaningfully used to reduce the nescience of the topic  $p$ . The directed nature of the dependency graph makes the asymmetry between tools and problems explicit: in general,  $Q_{t \rightarrow p}$  and  $Q_{p \rightarrow t}$  represent distinct questions, and only one of them may be well defined.

Operationally, the question  $Q_{t \rightarrow p}$  can be interpreted as asking whether, and how, the knowledge associated with topic  $t$  can be used to make progress on the unresolved aspects of topic  $p$ .

The nescience dependency graph not only specifies the existence of a question, but also provides a quantitative measure of its potential impact.

**Definition 7.5.2** Let  $Q_{t \rightarrow p}$  be a question. The *pair-specific nescience reduction* associated with  $Q_{t \rightarrow p}$  is defined as

$$\Delta N(p; t) = N(p) - N(p \mid t),$$

which is strictly positive by definition of the dependency graph.

The quantity  $\Delta N(p; t)$  measures how much of the unresolved ignorance about  $p$  can be reduced by assuming the knowledge background associated with  $t$ . Unlike topic-level metrics, this quantity is intrinsically relational and depends on the specific pairing of tool and problem.

The interestingness of a question cannot, in general, be reduced to a single scalar quantity without introducing additional assumptions. Instead, the theory of nescience naturally induces a multi-objective evaluation of questions based on theory-derived quantities.

Given a question  $Q_{t \rightarrow p}$ , the following dimensions are relevant:

- $\Delta N(p; t)$ , the pair-specific nescience reduction;
- $M(t)$ , the maturity of the tool topic  $t$ ;
- $A(t)$ , the applicability of the tool topic  $t$ ;
- $N(p)$ , the nescience of the problem topic  $p$ ;
- $D(p)$ , the integrative depth of the problem topic  $p$ .

Together, these quantities define the objective vector

$$\Phi(t, p) = (\Delta N(p; t), M(t), A(t), N(p), D(p)).$$

**Definition 7.5.3** A question  $Q_{t \rightarrow p}$  is said to be *at least as interesting* as another question  $Q_{t' \rightarrow p'}$  if

$$\Phi(t, p) \geq \Phi(t', p')$$

componentwise, with at least one inequality being strict. The set of *interesting questions* is defined as the Pareto frontier of questions under this partial order.

This definition does not impose a total ordering on questions, and naturally allows for the existence of incomparable questions corresponding to different trade-offs between nescience reduction, reliability of tools, and structural richness of problems.

In practical applications, such as research planning, portfolio construction, or automated recommendation systems, it is often necessary to rank questions or to select a subset of them. This requires the introduction of a *decision maker*, understood as a scalarization of the underlying multi-objective problem.

Several decision makers are particularly natural in the present framework:

- **Risk-averse decision maker.** Select questions that maximize  $\Delta N(p; t)$  subject to a lower bound on tool maturity,  $M(t) \geq \tau$ , for some threshold  $\tau$ .
- **Exploratory decision maker.** Maximize  $\Delta N(p; t) \cdot N(p)$ , favoring questions that yield large reductions on highly unresolved problems.
- **Scarcity-aware decision maker.** Favor questions associated with problems of low integrative depth  $D(p)$ , reflecting situations where few alternative nescience-reducing paths are available.
- **Balanced scalarization.** Combine normalized versions of the components of  $\Phi(t, p)$  using a weighted or Euclidean aggregation, yielding a total ordering suitable for visualization or large-scale comparison.

The choice of decision maker depends on contextual factors such as acceptable risk, available resources, and strategic goals. Importantly, these choices

do not alter the underlying theory, but merely provide different ways of navigating the space of interesting questions it defines.

In summary, interesting questions are defined in the theory of nescience as graph-grounded objects corresponding to nescience-reducing relationships between topics. Their interestingness emerges from a multi-objective structure inherent in the theory, while scalar rankings arise only at the level of decision making and application.

■ **Example 7.6** We illustrate the proposed framework with a simplified example drawn from cancer research. Consider the following topics in the set  $\mathcal{T}$ :

- $p = \text{Cancer progression}$ , understood as the set of mechanisms by which cancer develops, invades tissue, and metastasizes.
- $t_1 = \text{Genomics}$ , including gene expression profiling and mutation analysis.
- $t_2 = \text{Statistical learning}$ , encompassing classification, regression, and model selection techniques.
- $t_3 = \text{Cell signaling pathways}$ , describing biochemical interaction networks inside cells.
- $t_4 = \text{Immunotherapy}$ , covering immune-based treatment strategies.

Assume that empirical and theoretical work has established the following nescience-reducing relationships:

$$N(p \mid t_i) < N(p) \quad \text{for } i \in \{1, 2, 3, 4\}.$$

These relationships define arcs  $(t_i, p)$  in the nescience dependency graph  $G_N$ . Each arc corresponds to a question  $Q_{t_i \rightarrow p}$  asking whether the knowledge associated with  $t_i$  can be used to reduce ignorance about cancer progression.

Suppose further that the estimated pair-specific nescience reductions satisfy

$$\Delta N(p; t_1) > \Delta N(p; t_2) > \Delta N(p; t_3) > \Delta N(p; t_4),$$

reflecting the fact that genomic data currently explains a larger fraction of observed variability in cancer progression than the other topics.

At the same time, the maturity values of the tool topics may differ substantially. For instance, statistical learning may have high maturity due to well-established theory and methods, whereas immunotherapy may exhibit lower maturity due to incomplete understanding of immune response dynamics. Integrative depth  $D(p)$  is high, since multiple independent background topics reduce the nescience of  $p$ .

Within this framework, the set of questions

$$\{Q_{t_1 \rightarrow p}, Q_{t_2 \rightarrow p}, Q_{t_3 \rightarrow p}, Q_{t_4 \rightarrow p}\}$$

forms the candidate space of graph-grounded questions. Their interestingness is evaluated by comparing the corresponding objective vectors

$$\Phi(t_i, p) = (\Delta N(p; t_i), M(t_i), A(t_i), N(p), D(p)).$$

Different decision makers may select different questions from this set. A risk-averse decision maker may favor  $Q_{t_2 \rightarrow p}$  due to the high maturity of statistical learning, even if its nescience reduction is not maximal. An exploratory decision maker may prioritize  $Q_{t_1 \rightarrow p}$ , which yields the largest reduction in nescience. A scarcity-aware decision maker would note that  $D(p)$  is high, indicating that cancer progression admits many independent lines of inquiry, and may therefore prioritize questions that explore underrepresented connections, such as  $Q_{t_4 \rightarrow p}$ . ■

This example illustrates how the theory of nescience supports the systematic generation and evaluation of research questions. Questions are not arbitrary combinations of topics, but graph-grounded objects whose interestingness emerges from the interplay between pair-specific nescience reduction, the maturity of available tools, and the structural richness of the underlying problem.

## 7.6 New Topics

The region of *unknown unknowns* plays a central role in scientific progress, as it contains those research topics that have not yet been conceptualized or identified. One of the main objectives of the theory of nescience is to provide principled tools for exploring this region and for anticipating the emergence of future research topics. In this section we introduce a formal framework for the identification of *new topics*, understood as topics that lie beyond the current knowledge frontier.

A guiding intuition, illustrated in Figure 1.8, is that the probability of crossing the knowledge frontier through the combination of existing topics is higher when the topics involved are themselves poorly understood. Well-understood topics are typically embedded in dense and highly explored neighborhoods of the representation space, so their combinations tend to remain within the domain of known knowledge. In contrast, topics with high nescience are associated with sparser and less structured neighborhoods, making combinations involving them more likely to reach unexplored regions.

### 7.6.1 Candidate Topic Combinations

Let  $\mathcal{T}$  denote the set of existing research topics, and let  $N(t)$  denote the nescience of topic  $t$ , with  $\tilde{N}(t)$  its normalized version. Let  $D(t)$  denote the

(normalized) integrative depth of  $t$ , as introduced in Section 7.1.

**Definition 7.6.1** Given two distinct topics  $t_1, t_2 \in \mathcal{T}$ , a *candidate topic combination* is the unordered pair

$$C_{\{t_1, t_2\}} = \{t_1, t_2\}.$$

A candidate topic combination does not, by itself, constitute a new topic. Rather, it represents a potential source from which a new topic may emerge, depending on whether the combination leads beyond the knowledge frontier.

### 7.6.2 Multi-Objective Characterization of New Topics

The emergence of a new topic from a candidate combination cannot be adequately captured by a single scalar quantity. Instead, it is naturally described as a multi-objective problem involving theory-internal quantities.

Given a candidate combination  $C_{\{t_1, t_2\}}$ , we associate the following objective vector:

$$\Psi(t_1, t_2) = (\tilde{N}(t_1), \tilde{N}(t_2), D(t_1), D(t_2)).$$

Each component has a clear interpretation:

- $\tilde{N}(t_1), \tilde{N}(t_2)$  measure how close the topics are to the knowledge frontier;
- $D(t_1), D(t_2)$  measure the structural richness of the topics, in terms of the number of distinct ways background knowledge can reduce their nescience.

High values of  $\tilde{N}$  indicate proximity to the frontier, while high values of  $D$  indicate that the topic participates in many independent nescience-reducing relations. Combinations in which both topics exhibit high nescience and substantial integrative depth are therefore structurally well positioned to give rise to new topics beyond the frontier.

**Definition 7.6.2** A candidate combination  $C_{\{t_1, t_2\}}$  is said to have *high potential as a new topic* if it lies on the Pareto frontier of the set

$$\{\Psi(t_i, t_j) : t_i, t_j \in \mathcal{T}, i \neq j\}$$

under the componentwise partial order.

This definition avoids imposing an arbitrary total ordering on candidate combinations and reflects the fact that different combinations may represent different trade-offs between proximity to the frontier and structural richness.

### 7.6.3 Decision Makers and Practical Selection

In practical applications, such as research planning or automated topic discovery systems, it is often necessary to select or rank candidate combinations. This requires introducing a *decision maker*, understood as a scalarization of the multi-objective problem above.

Examples of decision makers include:

- **Frontier-seeking decision maker**, which prioritizes combinations maximizing  $\tilde{N}(t_1) + \tilde{N}(t_2)$ ;
- **Structure-aware decision maker**, which prioritizes combinations with high  $D(t_1) + D(t_2)$ , favoring structurally rich topics;
- **Balanced decision maker**, which applies a weighted aggregation of the components of  $\Psi(t_1, t_2)$ .

The choice of decision maker depends on contextual factors and does not alter the underlying theoretical framework.

### 7.6.4 Intradisciplinary and Interdisciplinary Combinations

**Definition 7.6.3** Let  $\mathcal{A} \subseteq \mathcal{T}$  be a research area. A candidate combination  $C_{\{t_1, t_2\}}$  is said to be *intradisciplinary* if  $t_1, t_2 \in \mathcal{A}$ ; otherwise, it is *interdisciplinary*.

Interdisciplinary combinations often correspond to candidate combinations whose components lie in distant regions of the nescience dependency graph. Such combinations are therefore more likely to explore underconnected regions of the representation space and to contribute to the expansion of the knowledge frontier, although this is not guaranteed in general.

## References

The following works provide theoretical and philosophical foundations for the concepts of interestingness, maturity, and the combination of topics as tools and problems.

[Cha13] An accessible introduction to the philosophy of science, addressing how scientific questions are formulated, evaluated, and justified.

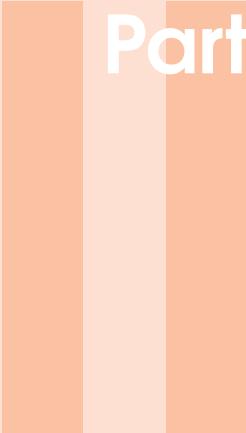
[PM19] Introduces the principles of causal reasoning, crucial for determining whether applying one topic as a tool can effectively address another as a problem.

[Pop14] Discusses falsifiability, novelty, and the importance of bold conjectures—foundational ideas for the “new and original” criterion.

[Shm+10] Clarifies the distinction between explanatory and predictive goals, helping to differentiate between topics valuable as problems versus tools.

[Van80] Explores the aims of science, model construction, and empirical adequacy, offering a philosophical context for defining “interesting” research.





# Part 2: Applications

8	Machine Learning .....	177
9	Analysis of Science .....	223
10	The Discovery of the Unknown .....	257





## 8. Machine Learning

*There are no difficult problems,  
only lack of imagination.*

Antonio García

We have seen that the most challenging problems to which we can apply the results of the theory of nescience arise when the set of entities  $\mathcal{E}$  under study is composed of abstract elements. The difficulty with abstract entities is that there is no straightforward way to encode them as finite strings of symbols from which they can be effectively reconstructed. Without such an encoding, it becomes impossible to measure their redundancy, accuracy, or any other property required by the theory of nescience.

In practice, a common approach to overcome this limitation is to design an experiment and collect observable outcomes, as is typically done in physics or other empirical sciences. Another possible approach is to collect a set of indirect measurements that serve as observable proxies for the abstract entities, for example, by recording the behavior of users in an online social network to study social interaction patterns.

This chapter is devoted to applying the concept of minimum nescience to the field of machine learning. We assume that the entities under study have been encoded as a dataset  $\mathbb{X}$  consisting of  $n$  training vectors, each with  $p$

predictors (or features), and an associated response variable  $\mathbf{y}$  (see Section F.2).

We will begin by presenting practical approximations for the concepts of miscoding, inaccuracy, and surfeit when the entities are represented as datasets. We will then show how these quantities can be combined into the single measure of nescience. These approximations will allow us to introduce the minimum nescience principle, a method designed to automate the process of selecting optimal models in machine learning (a form of automated machine learning, or AutoML).

In addition to introducing these approximations, we will demonstrate how to apply them to solve practical machine learning problems. The examples provided will be based on the `nescience`<sup>1</sup> library, an open-source Python library that implements the ideas presented in this chapter.

## 8.1 Nescience Python Library

The `nescience` library is an open-source Python package that implements the ideas presented in this book, specifically their application to the field of machine learning. The library follows the API design and conventions of the widely used `scikit-learn` toolkit, which allows it to be seamlessly combined with the methods and workflows provided by that package.

The `nescience` library can be installed with the `pip` utility:

```
pip install nescience
```

On the companion website for the library, the reader will find a collection of `jupyter-lab` notebooks that illustrate how the library works. Each subsection of this chapter is accompanied by a notebook containing implementations of all the examples discussed, so that the reader can reproduce the results and experiment with them interactively. Additional information about the `nescience` library, including a complete API reference, is also available on the library's website.

## 8.2 A Note About Compression

As is customary, the Kolmogorov complexity  $K(s)$  of a string  $s$  will be approximated by the length of the compressed version of that string using a standard compressor. In this way, we can estimate the normalized compression distance as follows:

$$E_Z(\mathbf{x}_j, \mathbf{y}) = \frac{\max\{\hat{K}_Z(\mathbf{x}_j \mid \mathbf{y}), \hat{K}_Z(\mathbf{y} \mid \mathbf{x}_j)\}}{\max\{\hat{K}_Z(\mathbf{x}_j), \hat{K}_Z(\mathbf{y})\}},$$

<sup>1</sup><https://github.com/rleiva/nescience>

where  $\hat{K}_Z(s)$  denotes the length of the compressed version of the string  $s$  using the compressor  $Z$ .

In the particular case of having a vector  $\mathbf{x} = \{x_1, \dots, x_n\}$  of measurements, the string  $s$  to be compressed will be the concatenation of the encoded values  $s = \langle x_1, \dots, x_n \rangle$ . In practice, it is usually easier to compute the joint encoding of two vectors than their conditional encoding. For this reason, we prefer the following equivalent definition of normalized compression distance:

$$E_Z(\mathbf{x}_j, \mathbf{y}) = \frac{\hat{K}_Z(\mathbf{x}_j, \mathbf{y}) - \min\{\hat{K}_Z(\mathbf{x}_j), \hat{K}_Z(\mathbf{y})\}}{\max\{\hat{K}_Z(\mathbf{x}_j), \hat{K}_Z(\mathbf{y})\}}.$$

As the compression technique, we will use a code  $C$  of minimal expected length, given the relative frequencies of the observed values (see Section D.3). If  $\mathbf{x}$  is a qualitative vector (either a feature or the target variable) taking values from a set of labels  $\mathcal{G} = \{g_1, \dots, g_\ell\}$ , that is,  $\mathbf{x} \in \mathcal{G}^n$ , the quantity  $\hat{K}_C(\mathbf{x})$  can be computed as:

$$\hat{K}_C(\mathbf{x}) = - \sum_{i=1}^{\ell} \log_2 \left( \frac{\sum_{j=1}^n I(x_j = g_i)}{n} \right).$$

If  $\mathbf{x}$  is based on a continuous random variable, we cannot directly calculate the probability of each  $x_j$  since, in general, the underlying probability distribution of  $\mathbf{x}$  is unknown, and moreover  $P(x_j) = 0$  for all  $j$ . To approximate  $K(\mathbf{x})$  using a minimal-length code  $C$ , we must first discretize the vector  $\mathbf{x}$  into a finite set of intervals.

A discretization algorithm maps a (potentially huge) set of numeric values to a reduced set of discrete categories, inevitably losing some information. The choice of discretization algorithm can have a significant impact on the practical computation of nescience. Ideally, we want a discretization method that balances bias and variance: it should produce enough intervals to capture relevant variation (low bias), while ensuring enough observations per interval to obtain reliable frequency estimates (low variance).

Common techniques include *equal-width discretization*, *equal-frequency discretization*, and *fixed-frequency discretization*. However, these methods require selecting a hyperparameter (the number of intervals), which makes them unsuitable for fully automated model selection.

In the nescience library, we use a proportional discretization approach (see Section ??), where both the number of intervals  $m$  and the number of observations per interval  $s$  are proportional to the total number of observations  $n$ . In particular, the library sets  $s = m = \sqrt{n}$ , which grows adaptively with the dataset size without introducing external hyperparameters.

Using this discretization procedure, we can approximate the Kolmogorov complexity of a vector  $\mathbf{x}$  by:

$$\hat{K}_C(\mathbf{x}) = - \sum_{i=1}^m \log_2 \left( \frac{\sum_{j=1}^n I(x_j \in D_i)}{n} \right),$$

where  $D_i$  is the  $i$ -th interval defined by the endpoints  $(i-1, i)$ .

This quantity  $\hat{K}_C$  can be generalized to an arbitrary number of  $m$  vectors  $\hat{K}_C(\mathbf{x}_1, \dots, \mathbf{x}_m)$  composed of  $n$  samples each, by considering the joint encoded vector

$$\langle \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m \rangle = \{ \langle x_{11}, x_{12}, \dots, x_{1m} \rangle, \dots, \langle x_{n1}, x_{n2}, \dots, x_{nm} \rangle \}.$$

**Proposition 8.2.1** The normalized compression distance of two vectors  $\mathbf{x}$  and  $\mathbf{y}$ , when computed using a compressor based on optimal codes, is equivalent to the normalized mutual information between these two vectors, that is:

$$NCD_C(\mathbf{x}, \mathbf{y}) = 1 - \frac{I(\mathbf{x}; \mathbf{y})}{\max\{H(\mathbf{x}), H(\mathbf{y})\}}.$$

*Proof.* We must prove that

$$\begin{aligned} NCD(\mathbf{x}, \mathbf{y}) &= \frac{C(\mathbf{x}, \mathbf{y}) - \min\{C(\mathbf{x}), C(\mathbf{y})\}}{\max\{C(\mathbf{x}), C(\mathbf{y})\}} \\ &= \frac{nH(\mathbf{x}, \mathbf{y}) - \min\{nH(\mathbf{x}), nH(\mathbf{y})\}}{\max\{nH(\mathbf{x}), nH(\mathbf{y})\}} \\ &= \frac{H(\mathbf{x}, \mathbf{y}) - \min\{H(\mathbf{x}), H(\mathbf{y})\}}{\max\{H(\mathbf{x}), H(\mathbf{y})\}}. \end{aligned}$$

Consider two cases.

**Case 1:**  $H(\mathbf{x}) > H(\mathbf{y})$ .

$$\begin{aligned} NCD(\mathbf{x}, \mathbf{y}) &= \frac{H(\mathbf{x}, \mathbf{y}) - H(\mathbf{y})}{H(\mathbf{x})} = \frac{H(\mathbf{y}) + H(\mathbf{x} | \mathbf{y}) - H(\mathbf{y})}{H(\mathbf{x})} = \frac{H(\mathbf{x} | \mathbf{y})}{H(\mathbf{x})} \\ &= \frac{H(\mathbf{x}) - I(\mathbf{x}; \mathbf{y})}{H(\mathbf{x})} = 1 - \frac{I(\mathbf{x}; \mathbf{y})}{H(\mathbf{x})}. \end{aligned}$$

**Case 2:**  $H(\mathbf{y}) > H(\mathbf{x})$ .

$$\begin{aligned} NCD(\mathbf{x}, \mathbf{y}) &= \frac{H(\mathbf{x}, \mathbf{y}) - H(\mathbf{x})}{H(\mathbf{y})} = \frac{H(\mathbf{x}) + H(\mathbf{y} | \mathbf{x}) - H(\mathbf{x})}{H(\mathbf{y})} = \frac{H(\mathbf{y} | \mathbf{x})}{H(\mathbf{y})} \\ &= \frac{H(\mathbf{y}) - I(\mathbf{x}; \mathbf{y})}{H(\mathbf{y})} = 1 - \frac{I(\mathbf{x}; \mathbf{y})}{H(\mathbf{y})}. \end{aligned}$$

Combining both cases, we obtain:

$$NCD_C(\mathbf{x}, \mathbf{y}) = 1 - \frac{I(\mathbf{x}; \mathbf{y})}{\max\{H(\mathbf{x}), H(\mathbf{y})\}}.$$

■

### 8.3 Miscoding

In Section 3.1.1 we introduced the concept of miscoding as a quantitative measure of how well a string based encoding  $r \in \mathcal{R}$  represents a research entity from  $\mathcal{E}$ . The miscoding of a representation  $r$  was defined as:

$$\mu(r) = \min_{s \in \mathcal{R}_{\mathcal{E}}} \frac{\max\{K(s | r), K(r | s)\}}{\max\{K(s), K(r)\}}$$

and we saw that this quantity cannot be computed in practice in the general case: first, because it would require computations from an abstract oracle machine; second, because it relies on the uncomputable Kolmogorov complexity; and third, because it does not directly take into account the particular entity  $e$  we are interested in.

In this section, we will show how this concept can be adapted in practice to estimate the error made when using a dataset  $\mathbf{X}$  as a representation of a response variable  $\mathbf{y}$  (see Section F.2). Our goal is twofold: on one hand, we are interested in measuring the overall quality of the dataset  $\mathbf{X}$  as a predictor of the variable  $\mathbf{y}$ ; on the other, we want to identify those individual features  $\mathbf{x}_j$  of  $\mathbf{X}$  that have the highest predictive power for  $\mathbf{y}$ . This is precisely the problem addressed by discriminative models (see Section ??), in which we want to estimate the conditional distribution  $P(\mathbf{y} | \mathbf{X})$ .

Given a training dataset  $\mathbf{X}$ , we can approximate the miscoding of a feature  $\mathbf{x}_j$  for the target variable  $\mathbf{y}$  by computing the normalized information distance between  $\mathbf{x}_j$  and  $\mathbf{y}$  (see Section E.5):

$$E(\mathbf{x}_j, \mathbf{y}) = \frac{\max\{K(\mathbf{x}_j | \mathbf{y}), K(\mathbf{y} | \mathbf{x}_j)\}}{\max\{K(\mathbf{x}_j), K(\mathbf{y})\}}$$

The Kolmogorov complexity  $K(\mathbf{v})$  of a vector  $\mathbf{v}$  will be approximated by the length of the compressed version of that vector  $\hat{K}_C(\mathbf{v})$  using as compressor a minimal length code  $C$  (see Section 8.2).

**Definition 8.3.1** Let  $\mathbf{y}$  be a response variable,  $\mathbf{X}$  a dataset composed by  $p$  features, and  $\mathbf{x}_j$  the  $j$ -th feature. We define the regular feature miscoding

of  $\mathbf{x}_j$  as a representation of  $\mathbf{y}$ , denoted by  $\hat{\mu}(\mathbf{x}_j, \mathbf{y})$ , as:

$$\hat{\mu}(\mathbf{x}_j, \mathbf{y}) = \frac{\hat{K}_C(\mathbf{x}_j, \mathbf{y}) - \min\{\hat{K}_C(\mathbf{x}_j), \hat{K}_C(\mathbf{y})\}}{\max\{\hat{K}_C(\mathbf{x}_j), \hat{K}_C(\mathbf{y})\}}$$

Intuitively, the quantity  $\hat{\mu}(\mathbf{x}_j, \mathbf{y})$  measures, in relative terms, the additional description length required to fully encode  $\mathbf{y}$  given knowledge of  $\mathbf{x}_j$ , and vice versa. The lower this value, the better  $\mathbf{x}_j$  serves as a predictor for  $\mathbf{y}$ .

■ **Example 8.1** Let  $\mathbf{y}$  be a target variable composed of 1,000 random samples from a normal distribution  $N(3, 1)$  with mean  $\mu = 3$  and standard deviation  $\sigma = 1$ ; let  $\mathbf{x}_1$  be a predictor feature approximately equal to  $\mathbf{y}$  with some added noise, that is  $\mathbf{x}_1 = \mathbf{y} + N(3, 1)/10$ ; and let  $\mathbf{x}_2$  be a second predictor consisting of random samples from an exponential distribution with rate  $\lambda = 1$ .

```
from scipy.stats import norm, expon

y = norm.rvs(loc=3, scale=1, size=10000)
x1 = y + norm.rvs(loc=3, scale=1, size=10000) / 10
x2 = expon.rvs(size=10000)
```

We can use the Nescience library to compute the miscoding of the features  $\mathbf{x}_1$  and  $\mathbf{x}_2$  when they encode the target variable  $\mathbf{y}$ .

```
from fastautoml.miscoding import Miscoding
import numpy as np

X = np.column_stack((x1, x2))

miscoding = Miscoding()
miscoding.fit(X, y)
miscoding.miscoding_features(mode="regular")
```

The output of the library would be something similar to the following<sup>2</sup>:

```
array([0.27445364, 0.9934222])
```

As it was expected the miscoding of  $\hat{\mu}(\mathbf{x}_1, \mathbf{y})$  is much smaller than the miscoding of  $\hat{\mu}(\mathbf{x}_2, \mathbf{y})$ . In this case, we should prefer  $\mathbf{x}_1$  over  $\mathbf{x}_2$  as a predictor of  $\mathbf{y}$ .

Sometimes, we will use the normalized version of the complements of the individual miscodings, that is:

$$\frac{1 - \hat{\mu}(\mathbf{x}_i, \mathbf{y})}{\sum_{j=1}^p (1 - \hat{\mu}(\mathbf{x}_j, \mathbf{y}))},$$

<sup>2</sup>Since we are generating a list of 1,000 random samples, the reader could get a slightly different result when running this example.

instead of the raw values  $\hat{\mu}(\mathbf{x}_i, \mathbf{y})$ . This alternative, which we call the *adjusted feature miscoding*, is useful because it allows easier comparison with other feature selection techniques, and it has a visually intuitive interpretation as relative importance weights.

■ **Example 8.2** In this example we generate a synthetic dataset where the target variable  $\mathbf{y}$  is a collection of normally distributed clusters of points, and the training set  $\mathbf{X}$  is composed of both relevant and irrelevant predictors. In particular, we generate 1,000 samples with 20 features describing 10 clusters; only 4 of the features are relevant for prediction, and the remaining 16 are just random noise.

Figure 8.1 shows a two-dimensional projection of this dataset along the hyperplane composed of features 8 and 10.

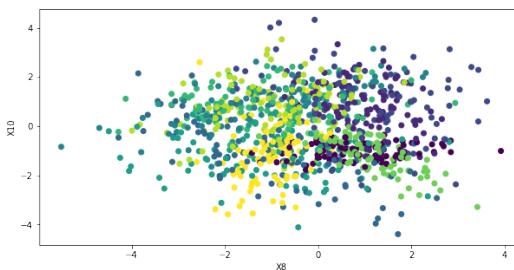


Figure 8.1: Gaussian Blob Cluster.

```
from fastautoml.miscoding import Miscoding
from sklearn.datasets.samples_generator import make_classification

X, y = make_classification(n_samples=1000, n_features=20, n_informative=4,
                           n_redundant=0, n_classes=10, n_clusters_per_class=1, flip_y=0)

miscoding = Miscoding()
miscoding.fit(X, y)
msd = miscoding.miscoding_features(mode='adjusted')
```

We use the adjusted version of miscoding for easier comparison with other feature selection techniques. If we plot the results (see Figure 8.2) we can see that the library has successfully identified the four relevant predictors ( $\mathbf{x}_3, \mathbf{x}_8, \mathbf{x}_{10}$  and  $\mathbf{x}_{16}$ ). Since we are using the adjusted version, higher values are better; note that the actual values should be interpreted only in relative terms.

We can compare miscoding with correlation, a common technique used in machine learning to identify relevant features. Figure 8.3 shows the correlation between the individual features of  $\mathbf{X}$  and the target variable  $\mathbf{y}$ . As we can observe, correlation fails to properly identify one of the relevant

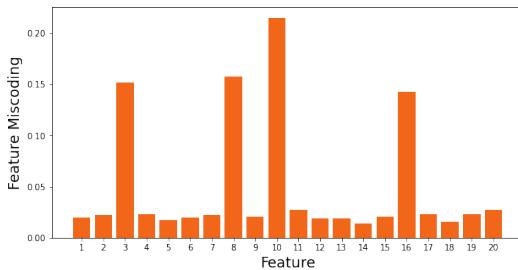


Figure 8.2: Miscoding of a Synthetic Dataset.

features ( $\mathbf{x}_3$ ).

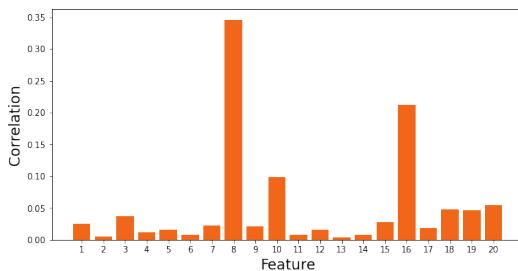


Figure 8.3: Correlation of a Synthetic Dataset.

Feature miscoding allows us to identify the most relevant features of a training dataset  $\mathbf{X}$ , but it cannot be used directly to compute the miscoding of the dataset itself. If we start with a miscoding of 1 (complete ignorance) and subtract the miscodings of the individual features, we may end up with a negative miscoding, which is not allowed by our theory. If we use the adjusted version, the dataset miscoding would be 0 for all datasets, which goes against the intuition that not all datasets  $\mathbf{X}$  represent a target variable  $\mathbf{y}$  equally well. According to the theory of nescience, we expect non-relevant features to *add*, rather than subtract, to the global miscoding of the dataset.

To address this, we introduce the concept of partial miscoding of a feature, defined as the difference between the adjusted and the normalized miscodings.

**Definition 8.3.2** Let  $\mathbf{y}$  be a target variable,  $\mathbf{X}$  a dataset composed by  $p$  features, and  $\mathbf{x}_j$  the  $j$ -th feature. We define the partial miscoding of  $\mathbf{x}_j$

as a representation of  $\mathbf{y}$ , denoted by  $\tilde{\mu}(\mathbf{x}_j, \mathbf{y})$ , as:

$$\tilde{\mu}(\mathbf{x}_i, \mathbf{y}) = \frac{1 - \hat{\mu}(\mathbf{x}_i, \mathbf{y})}{\sum_{j=1}^p 1 - \hat{\mu}(\mathbf{x}_j, \mathbf{y})} - \frac{\hat{\mu}(\mathbf{x}_i, \mathbf{y})}{\sum_{j=1}^p \hat{\mu}(\mathbf{x}_j, \mathbf{y})}$$

A positive partial miscoding indicates that the feature contributes to describing the target variable; a negative value indicates that the feature is not relevant (or is counterproductive) in representing  $\mathbf{y}$ .

■ **Example 8.3** We reuse the synthetic dataset of Example 8.2, but we increase the number of relevant features from 4 to 14. Then, we compute the list of partial miscodings.

```
from fastautoml.miscoding import Miscoding
from sklearn.datasets.samples_generator import make_classification

X, y = make_classification(n_samples=1000, n_features=20, n_informative=14,
                           n_redundant=0, n_classes=10, n_clusters_per_class=1, flip_y=0)

miscoding = Miscoding()
miscoding.fit(X, y)
msd = miscoding.miscoding_features(mode="partial")
```

As shown in Figure 8.4, the library not only correctly identifies the relevant features, but also assigns negative contributions to non-relevant features, thereby increasing the global miscoding when they are included.

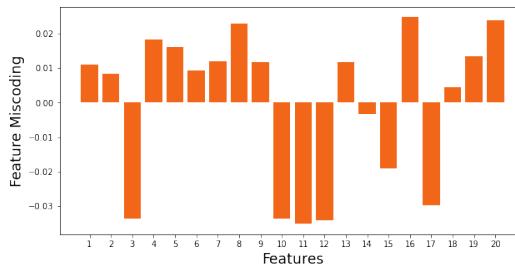


Figure 8.4: Partial Feature Miscoding.

Given the definition of partial feature miscoding, we can now define the miscoding of a target variable given a subset of predictors in a way that is closer to the original notion of miscoding in the theory of nescience.

■ **Definition 8.3.3** Let  $\mathbf{y}$  be a target variable,  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_p\}$  a dataset composed by  $p$  features, and  $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$  a subset of features, that is,  $\{\mathbf{z}_1, \dots, \mathbf{z}_k\} \subseteq \{\mathbf{x}_1, \dots, \mathbf{x}_p\}$ . We define the miscoding of  $\mathbf{Z}$  as a represen-

tation of  $\mathbf{y}$ , denoted by  $\hat{\mu}(\mathbf{Z}, \mathbf{y})$ , as:

$$\hat{\mu}(\mathbf{Z}, \mathbf{y}) = \sum_{i=1}^k \tilde{\mu}(\mathbf{z}_i, \mathbf{y})$$

■ **Example 8.4** Using the dataset and partial feature miscodings computed in Example 8.3, Figure 8.5 shows the evolution of the miscoding of the training subset  $\mathbf{Z}$  as we add more features to the study.

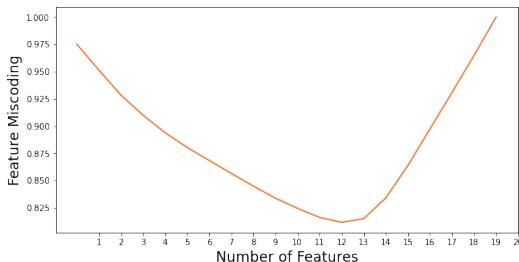


Figure 8.5: Accumulated Partial Feature Miscoding.

In the next example we compare the performance of a classifier when using the full dataset and a reduced version containing only those features identified as relevant (i.e., with positive partial miscoding).

■ **Example 8.5** We train a neural network on the `scikit-learn` digits dataset to classify handwritten digits. The evaluation criterion is the classifier's accuracy on a held-out test set. The neural network is trained and evaluated first on all features, and then on a reduced dataset containing only those features with positive partial miscoding.

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.datasets import load_digits
from fastautoml.fastautoml import Miscoding

data = load_digits()
X_raw = data.data
y_raw = data.target

mCoding = Miscoding()
mCoding.fit(X_raw, y_raw)
mscd = mCoding.miscoding_features(miscoding='partial')
X_red = X_raw[:, np.where(mscd > 0)[0]]
y_red = y_raw

X_raw_train, X_raw_test, y_raw_train, y_raw_test = train_test_split(X_raw,
```

```
    y_raw, test_size=.3)
X_red_train, X_red_test, y_red_train, y_red_test = train_test_split(X_red,
    y_red, test_size=.3)

clf = MLPClassifier(alpha=1, max_iter=1000)

clf.fit(X_raw_train, y_raw_train)
score_raw = clf.score(X_raw_test, y_raw_test)

clf.fit(X_red_train, y_red_train)
score_red = clf.score(X_red_test, y_red_test)

reduction = 1 - X_red_train.shape[1] / X_raw_train.shape[1]

print("Score raw:", score_raw, " Score Miscoding:", score_red,
      " Reduction:", reduction)

Score raw: 0.9833333333333333  Score Miscoding: 0.9814814814814815  Data Reduction: 0.47
```

Running the above code typically shows that the classifier's accuracy is essentially the same for both datasets (about 98% on the test set). However, the reduced dataset used for training (selected via positive partial miscoding) is about 47% smaller than the original dataset. This size reduction can substantially decrease training time. Smaller datasets are also beneficial when training ensembles (e.g., random forests), where many models must be fitted. ■

Intuitively, as Example 8.4 suggests, we should prefer the subset  $\mathbf{Z} \subseteq \mathbf{X}$  composed of those features whose partial miscodings are greater than zero. However, as we will see in the following sections, this might not always be the case. Feature selection is only one of the criteria used when searching for an optimal model for an entity represented by a dataset; other components of nescience, such as inaccuracy or surfeit, may suggest a different subset of predictors. The overall optimization criterion is the nescience itself. A sensible approach to using partial miscoding is to incrementally add to the model those features with the highest (positive) partial miscoding until all positive features are included or an external optimality criterion is reached.

In the case of a generative model (see Section ??), i.e., a machine learning algorithm designed to estimate the joint probability  $P(\mathbf{X}, \mathbf{y})$ , we can use miscoding to assess how individual features relate to each other, by computing  $\hat{\mu}(\mathbf{x}_i, \mathbf{x}_j)$  for each pair  $i, j \leq p$ . The result is a *miscoding matrix* (see Example 8.6).

■ **Example 8.6** The Boston dataset included in the `scikit-learn` library contains a collection of variables that (potentially) explain the price of houses in the Boston area. In this example, instead of identifying the factors that contribute most to price, we study the interdependence among factors using

a miscoding matrix.

```
from fastautoml.fastautoml import Miscoding
from sklearn.datasets import load_boston

data = load_boston()

miscoding = Miscoding(X_type="numeric", y_type="numeric")
miscoding.fit(data.data, data.target)
mscd_matrix = miscoding.features_matrix(mode='regular')
```

Figure 8.6 shows a heatmap of the miscoding matrix computed over the features. Darker values represent lower miscoding (note we use the regular version here). In particular, the diagonal entries are equal to zero.

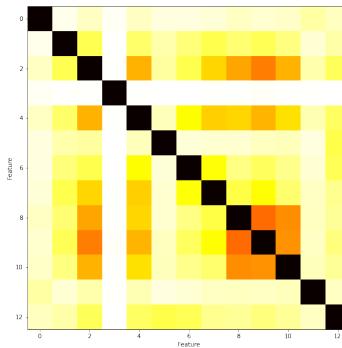


Figure 8.6: Regular Miscoding Matrix.

The minimum value of 0.52 is obtained for the pair (8, 9), corresponding to the features "index of accessibility to radial highways" and "full-value property-tax rate per \$10,000". These features are good candidates to evaluate jointly in a predictive model, since they contain non-redundant information. The maximum value of 0.99 is achieved for the pair (3, 12), corresponding to "Charles River dummy variable" and "% lower status of the population". These features contain almost the same information; including both in a model adds little new information while increasing model complexity and the risk of overfitting.

## 8.4 Inaccuracy

In Section 4.1 we defined the inaccuracy of a description  $d \in \mathcal{D}$  for a representation  $r \in \mathcal{R}$  as the normalized information distance between the representation  $r$  and the string  $\Gamma(d)$  printed out by a universal Turing machine

when given the description as input:

$$\iota(d, r) = \frac{\max\{K(r \mid \Gamma(d)), K(\Gamma(d) \mid r)\}}{\max\{K(r), K(\Gamma(d))\}}$$

Inaccuracy, being based on Kolmogorov complexity, is not computable in the general case and must be approximated in practice. In this section we show how this concept can be estimated for a model trained on a dataset. The approach parallels the one used for miscoding (see Section ??).

For discriminative models, which estimate  $P(\mathbf{y} \mid \mathbb{X})$ , the natural proxy compares predicted targets  $\hat{\mathbf{y}}$  with true targets  $\mathbf{y}$  via normalized compression distance.

**Definition 8.4.1** Let  $\mathbb{X}$  be a dataset,  $\mathbf{y}$  a response variable,  $m$  a model, and  $\hat{\mathbf{y}} = m(\mathbb{X})$  the predicted values by  $m$  given  $\mathbb{X}$ . We define the *inaccuracy* of the model  $m$  for the target values  $\mathbf{y}$ , denoted by  $\hat{\iota}(\hat{\mathbf{y}}, \mathbf{y})$ , as:

$$\hat{\iota}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\hat{K}_C(\hat{\mathbf{y}}, \mathbf{y}) - \min\{\hat{K}_C(\hat{\mathbf{y}}), \hat{K}_C(\mathbf{y})\}}{\max\{\hat{K}_C(\hat{\mathbf{y}}), \hat{K}_C(\mathbf{y})\}}$$

Intuitively,  $\hat{\iota}(\hat{\mathbf{y}}, \mathbf{y})$  measures how far the predictions are from the true values. The lower this quantity, the better  $m$  is as a predictor of  $\mathbf{y}$ . Unlike standard accuracy metrics, this measure accounts not only for how difficult it is to reconstruct  $\mathbf{y}$  from  $\hat{\mathbf{y}}$ , but also for how much additional information  $\hat{\mathbf{y}}$  contains that is unrelated to  $\mathbf{y}$ .

For generative models, which aim to capture the joint distribution  $P(\mathbb{X}, \mathbf{y})$ , we compare the observed joint sample with a synthetic joint sample generated by the model.

**Definition 8.4.2** Let  $m$  be a generative model for  $(\mathbb{X}, \mathbf{y})$ . Let  $(\hat{\mathbb{X}}, \hat{\mathbf{y}})$  be a synthetic sample of the same size as  $(\mathbb{X}, \mathbf{y})$  generated by  $m$  (after applying the same encoding/discretization scheme). The *generative inaccuracy* of  $m$  for  $(\mathbb{X}, \mathbf{y})$  is:

$$\hat{\iota}_g(m; (\mathbb{X}, \mathbf{y})) = \frac{\hat{K}_C(\langle \mathbb{X}, \mathbf{y} \rangle, \langle \hat{\mathbb{X}}, \hat{\mathbf{y}} \rangle) - \min\{\hat{K}_C(\langle \mathbb{X}, \mathbf{y} \rangle), \hat{K}_C(\langle \hat{\mathbb{X}}, \hat{\mathbf{y}} \rangle)\}}{\max\{\hat{K}_C(\langle \mathbb{X}, \mathbf{y} \rangle), \hat{K}_C(\langle \hat{\mathbb{X}}, \hat{\mathbf{y}} \rangle)\}}$$

When a model provides explicit probabilities, one may also define  $\hat{K}$  using the model code length  $-\sum \log_2 \hat{p}_m(\cdot)$ ; we keep the compression-based proxy here for consistency with the rest of the chapter.

■ **Example 8.7** According to the minimum nescience principle, inaccuracy is the normalized compression distance between the actual targets  $\mathbf{y}$  and the predicted targets  $\hat{\mathbf{y}}$ . We compare our inaccuracy metric with a classical score (we use  $1 - \text{accuracy}$ ) on the MNIST digits dataset.

```

from fastautoml.fastautoml import Inaccuracy
from sklearn.datasets import load_digits

X, y = load_digits(return_X_y=True)

inacc = Inaccuracy()
inacc.fit(X, y)

```

We train a decision-tree classifier with maximum depth  $i \in \{1, \dots, 20\}$ :

```

from sklearn.tree import DecisionTreeClassifier

scores      = list()
inaccuracies = list()

for i in range(20):

    tree = DecisionTreeClassifier(max_depth=i, random_state=42)
    tree.fit(X, y)

    scores.append(1 - tree.score(X, y))
    inaccuracies.append(inacc.inaccuracy_model(tree))

```

We compare 1 – accuracy and inaccuracy. As shown in Figure 8.7, both behave similarly, with inaccuracy typically taking larger values due to its stronger emphasis on incorrectly predicted values.

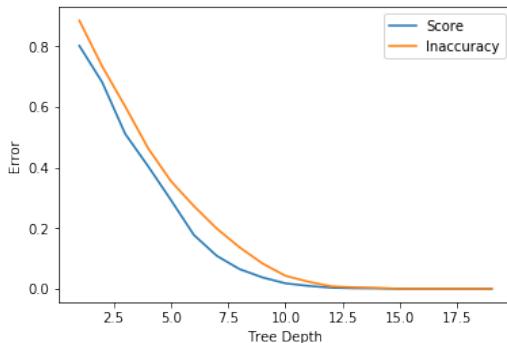


Figure 8.7: Inaccuracy vs. Score of Decision Trees

In Example 8.7 we observed that deeper trees yield smaller training error. Of course, larger depth increases the risk of overfitting. However, within the nescience framework, overfitting is handled by *surfeit* (see Section ??), so inaccuracy need not penalize complexity directly.

We can interpret inaccuracy as the effort, measured in description length, required to fix a model’s predictions. Consequently, a model that makes one hundred *identical* errors is easier to correct than a model that makes one

hundred *distinct* errors (see Example 8.8).

■ **Example 8.8** We again use a decision-tree classifier, now with the hyperparameter `min_samples_leaf=5` (a common regularization to reduce overfitting).

```
tree = DecisionTreeClassifier(min_samples_leaf=5)
tree.fit(X, y)
```

The inaccuracy of this new trained model is 0.17, and its score 0.08. Next we will artificially introduce one hundred errors in the dataset, simulating the case that the tree is not able to model correctly these data points. In this particular case all the errors are exactly the same.

```
X2 = X.copy()
y2 = y.copy()
for i in range(100):
    X2 = np.append(X2, [X[0]], axis=0)
    y2 = np.append(y2, (y[0]+1) % 10)
```

The inaccuracy of the decision tree, given this new dataset, has increased<sup>3</sup> from 0.17 to 0.21.

```
inacc.fit(X2, y2)
inacc.inaccuracy_predictions(pred)
```

Score has also increased, in this case from 0.08 to 0.13.

```
1 - tree.score(X2, y2)
```

Finally, we are going to repeat exactly the same experiment, but this time instead of adding one hundred times the same error, adding one hundred different errors.

```
X3 = X.copy()
y3 = y.copy()
for i in range(100):
    index = np.random.randint(X.shape[0])
    X3 = np.append(X3, [X[index]], axis=0)
    y3 = np.append(y3, (y[index]+1) % 10)
```

In this last case the inaccuracy of the model has increased up to 0.25, meanwhile score remained the same. ■

In line with Example 8.8, consider an extreme binary case: a model that always flips the label (predicts False when the truth is True, and vice versa).

<sup>3</sup>Note that we had to `fit()` again the class `Inaccuracy` in order to use the new dataset. Normally this is not the way we use this class; instead what we should do is to fit once a dataset, and then compute the inaccuracy of different models. We are doing here in this way to demonstrate an interesting property of the concept of inaccuracy.

Classical accuracy deems this the worst possible model; our inaccuracy, however, is low because the predictions can be perfectly "fixed" by a simple transformation. In some applications (e.g., a hedge fund trading strategy), such a model would be extremely valuable.

In highly imbalanced datasets, standard accuracy can be misleading: a high score may simply reflect correct classification of the majority class while failing on the minority class. Practitioners address this with specialized metrics. With our inaccuracy metric, as Example 8.9 shows, a model that fails to classify the minority class is penalized even if that class has few samples.

■ **Example 8.9** We create a synthetic dataset with two classes, using the `make_classification` utility of scikit-learn, where one has 95% of the samples and the other 5%:

```
from sklearn.datasets import make_classification

depth = list()
score = list()
inacc = list()

inaccuracy = Inaccuracy()

for i in np.arange(1, 100):

    X, y = make_classification(n_samples=1000, n_features=2,
                               n_informative=2, n_redundant=0,
                               class_sep=2, flip_y=0, weights=[0.95,0.05])

    inaccuracy.fit(X, y)

    tree = DecisionTreeClassifier(min_samples_leaf=i)
    tree.fit(X, y)

    depth.append(i)
    score.append(1 - tree.score(X, y))
    inacc.append(inaccuracy.inaccuracy_model(tree))
```

We vary the minimum number of samples per leaf  $i \in \{1, \dots, 100\}$ . In Figure 8.8, for large  $i$  the accuracy-based error (i.e.,  $1 - \text{accuracy}$ ) may stay near 0.05, yet the inaccuracy remains high because the model effectively ignores the minority class.

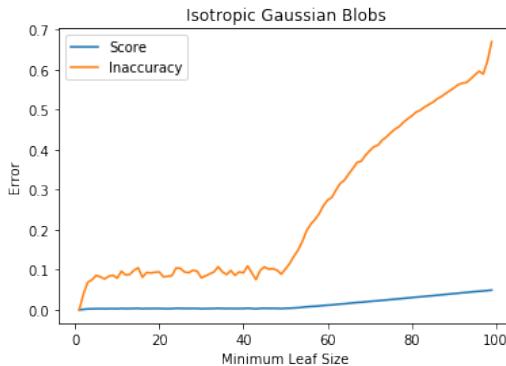


Figure 8.8: Inaccuracy of Decision Tree.

## 8.5 Surfeit

In Section 5.2 we defined the surfeit of the model  $m \in \mathcal{M}$  for a representation  $r \in \mathcal{R}$  as:

$$\sigma(m, r) = 1 - \frac{K(r)}{l(m)}$$

Since the length  $K(r)$  of the shortest possible description of  $r$  (its Kolmogorov complexity) is in general unknown, we must approximate this concept in practice. In the setting of a training dataset  $\mathbb{X}$  and a target variable  $\mathbf{y}$ , we approximate the surfeit of a model  $m$  for the representation  $\mathbf{y}$  by

$$\hat{\sigma}(m, \mathbf{y}) = 1 - \frac{\hat{K}_C(\mathbf{y})}{l(m)},$$

where  $\hat{K}_C(\mathbf{y})$  is the length of the compressed version of the vector  $\mathbf{y}$  using a minimal-length code  $C$  determined by the empirical relative frequencies of the values observed in  $\mathbf{y}$  (see Section ??).

**Definition 8.5.1** Let  $\mathbf{y}$  be a response variable, and  $\mathbb{X}$  a dataset composed of  $p$  features and  $n$  samples. We define the surfeit of the model  $m \in \mathcal{M}$  as a representation of  $\mathbf{y}$ , denoted by  $\hat{\sigma}(m, \mathbf{y})$ , as

$$\hat{\sigma}(m, \mathbf{y}) = 1 - \frac{\hat{K}_C(\mathbf{y})}{l(m)}.$$

This definition requires a method to encode models as strings of symbols so that their length  $l(m)$  can be computed. Ideally, one would encode models as Turing machines and agree on a universal Turing machine to interpret them. However, this would make it very difficult to add new models to

the nescience library. Instead, we use a restricted subset of the Python language as the model-encoding scheme: only a limited set of constructions is allowed, and third-party libraries are disallowed.

Surfeit can help avoid overfitted models. The higher the surfeit of a model, the greater the risk that the model is an overfit to the training dataset, as Example 8.10 illustrates.

■ **Example 8.10** We generate 900 samples from a sinusoidal curve and fit  $n$ -degree polynomials with  $n \in \{1, \dots, 15\}$ .

```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures

from Nescience.Nescience import Surfeit
from Nescience.Nescience import Inaccuracy

n_samples = 900
degrees = np.arange(1, 15)

X = np.sort(np.random.rand(n_samples) * 3)
y = np.cos(1.5 * np.pi * X)

linacc = list()
lsurfeit = list()

for i in degrees:

    poly = PolynomialFeatures(degree=i, include_bias=False)
    newX = poly.fit_transform(X[:, np.newaxis])

    linear_regression = LinearRegression()
    linear_regression.fit(newX, y)

    inacc.append(inaccuracy)
    inaccuracy = inaccuracy_model(linear_regression)

    sft.fit(newX, y)
    surfeit = sft.surfeit_model(linear_regression)

    linacc.append(inaccuracy)
    lsurfeit.append(surfeit)
```

In Figure 8.9 we see that, as expected, higher-degree polynomials yield lower training error, but at the cost of higher surfeit. The preferred model is one achieving low inaccuracy *and* low surfeit. ■

Another advantage of surfeit is that it allows us to compare models from different families. For example, if two models have the same accuracy, should we prefer a decision tree to a neural network, or a naive Bayes classifier to a support vector machine? The next example shows how surfeit can guide such choices.

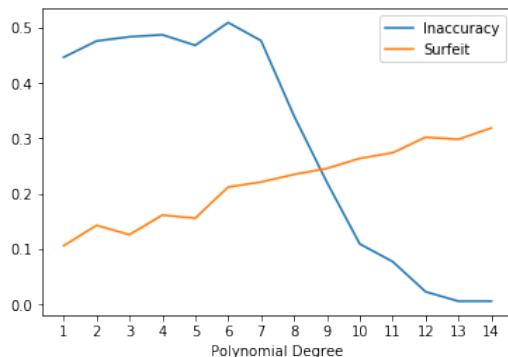


Figure 8.9: Surfeit vs Inaccuracy

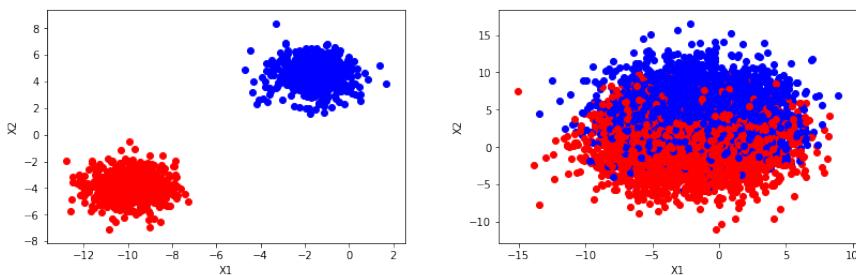


Table 8.1: Isotropic Gaussian blobs.

■ **Example 8.11** We compare a decision tree with a neural network on synthetic data consisting of two isotropic Gaussian blobs. First, with standard deviation 1 and two dimensions, the two clusters are easily separable (Table 8.1, left).

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from Nescience.Nescience import Surfeit
from Nescience.Nescience import Inaccuracy
from sklearn.datasets.samples_generator import make_blobs

X, y = make_blobs(n_samples=1000, centers=2, n_features=2, cluster_std=1)

tree = DecisionTreeClassifier()
tree.fit(X, y)
tree.score(X, y)

nn = MLPClassifier()
nn.fit(X, y)
nn.score(X, y)
```

```

sft = Surfeit()
sft.fit(X, y)

sft.surfeit_model(tree)
sft.surfeit_model(nn)

```

If we run the code, both models typically achieve accuracy 1. However, the decision tree often has substantially lower surfeit than the neural network (e.g., 0.25 vs. 0.73 in one run), so we would prefer the decision tree in this setting.

If we repeat the experiment with standard deviation 3 and higher dimensionality (Table 8.1, right), the situation can change:

```

X, y = make_blobs(n_samples=10000, centers=2, n_features=8, cluster_std=3)

```

In this second case, both models can again achieve very high accuracy, but the surfeit of the decision tree typically increases markedly (e.g., to  $\approx 0.82$ ), while the neural network's surfeit may remain similar (e.g.,  $\approx 0.76$ ). For this dataset, we would then prefer the neural network. ■

In Example 8.11 we assumed equal accuracy. When accuracies differ, we must appeal to the overall concept of *nescience* to decide between models, jointly considering inaccuracy and surfeit (and, where relevant, miscoding).

## 8.6 Nescience

In Chapter 6 we defined the concept of nescience as the solution to a nonlinear multiobjective optimization problem, in which we minimize the miscoding, inaccuracy, and surfeit of representations and models. The solution to this problem is, in general, not unique: we can often find multiple pairs (representation, model) such that none of the three quantities can be improved without degrading at least one of the others (Pareto optimality). However, in practice we expect a machine-learning library to provide a single solution when training a model on a dataset. To provide a unique solution, we resort to a utility function that selects one element from the Pareto-optimal set. The `nescience` library offers several utility functions; the default is the arithmetic mean of the three metrics.



The `nescience` library implements the following utility functions to approximate the concept of nescience, i.e., to compute  $\hat{v}(\mathbb{Z}, m, \mathbf{y})$ :

- Euclidean distance:  $(\hat{\mu}(\mathbb{Z}, \mathbf{y})^2 + \hat{\iota}(\hat{\mathbf{y}}, \mathbf{y})^2 + \hat{\sigma}(m, \mathbf{y})^2)^{1/2}$
- Arithmetic mean:  $\frac{\hat{\mu}(\mathbb{Z}, \mathbf{y}) + \hat{\iota}(\hat{\mathbf{y}}, \mathbf{y}) + \hat{\sigma}(m, \mathbf{y})}{3}$

- Geometric mean:  $(\hat{\mu}(\mathbb{Z}, \mathbf{y}) \times \hat{\iota}(\hat{\mathbf{y}}, \mathbf{y}) \times \hat{\sigma}(m, \mathbf{y}))^{1/3}$
- Product:  $\hat{\mu}(\mathbb{Z}, \mathbf{y}) \times \hat{\iota}(\hat{\mathbf{y}}, \mathbf{y}) \times \hat{\sigma}(m, \mathbf{y})$
- Addition:  $\hat{\mu}(\mathbb{Z}, \mathbf{y}) + \hat{\iota}(\hat{\mathbf{y}}, \mathbf{y}) + \hat{\sigma}(m, \mathbf{y})$
- Weighted mean:  $w_\mu \hat{\mu}(\mathbb{Z}, \mathbf{y}) + w_\iota \hat{\iota}(\hat{\mathbf{y}}, \mathbf{y}) + w_\sigma \hat{\sigma}(m, \mathbf{y})$
- Harmonic mean:  $\frac{3}{\hat{\mu}(\mathbb{Z}, \mathbf{y})^{-1} + \hat{\iota}(\hat{\mathbf{y}}, \mathbf{y})^{-1} + \hat{\sigma}(m, \mathbf{y})^{-1}}$

Euclidean distance and addition can produce nescience values greater than one, which conflicts with the intended  $[0, 1]$  range. The geometric mean, product, and harmonic mean have the issue that the resulting nescience is zero (or undefined, in the harmonic case) whenever any of the three components is zero. The weighted mean introduces three additional hyperparameters  $(w_\mu, w_\iota, w_\sigma)$  that must be chosen (often subject to  $w_\mu + w_\iota + w_\sigma = 1$ ). It remains an open question which utility function is most appropriate in general; by default we use the arithmetic mean.

Example 8.12 shows how to use the `nescience` library to compute the nescience of a dataset and a model.

■ **Example 8.12** We compute the nescience of a decision-tree classifier on the digits dataset (MNIST handwritten digits) from `sklearn`.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.datasets import load_digits
from Nescience.Nescience import Nescience

data = load_digits()

tree = DecisionTreeClassifier()
tree.fit(data.data, data.target)
tree.score(data.data, data.target)
[ ] 1

nescience = Nescience()
nescience.fit(data.data, data.target)

nescience.nescience(tree)
[ ] 0.5895603819965907
```

The training accuracy of the decision tree is 1.0, meaning all samples are correctly classified on the training set. This is symptomatic of overfitting. In practice we would use a train/test split or cross-validation. Nonetheless, the nescience value of about 0.59 flags that something is wrong with the model-data pairing. ■

As Example 8.12 illustrates, a key advantage of nescience is that it can evaluate model quality without requiring computationally expensive procedures such as cross-validation or holding out a test set. Another advantage is that nescience allows comparisons across different model families, as shown next.

■ **Example 8.13** We compare two model families on the Breast Cancer Wisconsin (Diagnostic) dataset from sklearn: a decision tree and a neural network.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.datasets import load_breast_cancer
from Nescience.Nescience import Nescience

data = load_breast_cancer()
X = data.data
y = data.target

tree = DecisionTreeClassifier(max_depth=3)
tree.fit(X, y)
tree.score(X, y)
[ ] 0.9789103690685413

nescience = Nescience()
nescience.fit(X, y)
nescience.nescience(tree)
[ ] 0.5945936419010083

nn = MLPClassifier()
nn.fit(X, y)
nn.score(X, y)
[ ] 0.9261862917398945

nescience.nescience(nn)
[ ] 0.7860523786210711
```

In one run, both models achieve high and broadly similar training accuracy. In this case, not only does the decision tree achieve the better score, but its nescience is also much lower than that of the multilayer perceptron; hence we should prefer the former to the latter.

Nescience can also be used to optimize hyperparameters within a (parametric) family of models. A practical advantage is that we can adopt a greedy search for a hyperparameter that is monotone with accuracy: select the value at which nescience stops decreasing and starts to increase, since this is the point at which we are no longer learning anything new from the dataset (see Example 8.14).

■ **Example 8.14** We revisit a decision tree on the breast cancer dataset. We train 10 trees with `max_depth` from 1 to 10. The deeper the tree, the higher the training accuracy, but also the higher the risk of overfitting. For each tree we compute the model's nescience and compare it to a cross-validation score. Figure 8.10 shows typical behavior: both nescience and the cross-validation error decrease as depth increases, up to an inflection point after which they increase, indicating overfitting. The nescience library suggests a maximum

depth of 7, whereas cross-validation indicates an optimum near 6.

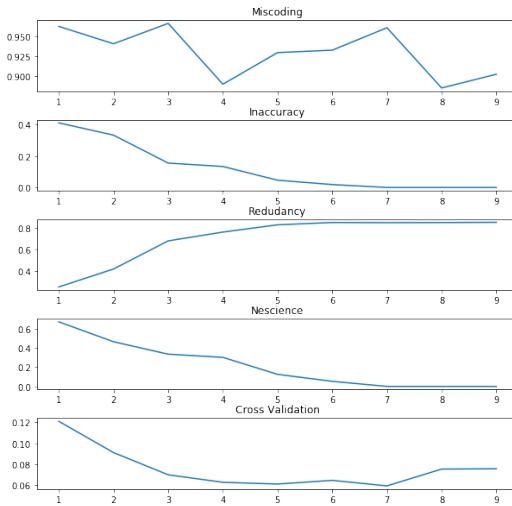


Figure 8.10: Evolution of Nescience with Tree Depth.

It is instructive to examine the three components in Figure 8.10. As expected, deeper trees reduce inaccuracy and increase surfeit. By contrast, miscoding can evolve nonmonotonically because each candidate tree may use a different subset of features at internal nodes. It would be desirable to have a tree-building algorithm that accounts for miscoding when selecting split features; such an algorithm is described in Section 8.11.

Finally, we consider hyperparameter searches where a greedy approach is not applicable—for example, when searching over multiple (often conflicting) hyperparameters. Grid search can be computationally expensive due to the number of combinations, and even more so if each candidate requires cross-validation. Since nescience can flag overfitting without cross-validation, it can substantially speed up hyperparameter search. Example 8.15 shows how to integrate nescience into `GridSearchCV`.

■ **Example 8.15** In this example we are going to see how we can use the `nescience` library to find the optimal hyperparameters for a model using a grid search. In particular, we are going to select the best hyperparameters for a multilayer perceptron classifier, including the number of hidden layers, and the size of those layers (what it is called Neural Architecture Search). The procedure will be demonstrated using the `digits` dataset.

```
from Nescience.Nescience import Nescience
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
```

First of all we have to provide a custom loss function based on the concept of nescience to be integrated with the search procedure. The next code shows how to implement such a function.

```
def my_custom_loss_func(estimator, X, y):
    nsc = Nescience()
    nsc.fit(X, y)
    nescience = nsc.nescience(estimator)

    # scikit-learn expect that higher numbers are better
    score = -nescience

    return score
```

Second, we have to define the grid of hyperparameters over which we are going to do the search. The larger the grid, the better the result, but also, the more computer time is required to evaluate all possible combinations.

```
parameters = {'solver': ['lbfgs'],
              'max_iter': [1000, 1500, 2000],
              'alpha': 10.0 ** -np.arange(1, 10, 3),
              'hidden_layer_sizes': [(60,), (100,), (60, 60,), (100, 100,),
                                     (60, 60, 60,), (100, 100, 100,)]}
```

Next code show how to do a classical grid search using the score of the models. The search will be evaluated using a train/test split of the dataset.

```
clf_std = GridSearchCV(estimator=MLPClassifier(), param_grid=parameters,
                       cv=3, iid=True, n_jobs=-1)
clf_std.fit(X_train, y_train)
clf_std.best_params_

[] {'alpha': 0.1,
[]  'hidden_layer_sizes': (100,),
[]  'max_iter': 1000,
[]  'solver': 'lbfgs'}

y_true, y_pred = y_test, clf_std.predict(X_test)
print(classification_report(y_true, y_pred))

[] precision    recall  f1-score   support
[] avg / total       0.98       0.97       0.97       540
```

Next code show how to perform exactly the same search, but using the concept of nescience instead of the metric score.

```
clf_nsc = GridSearchCV(estimator=MLPClassifier(), param_grid=parameters,
                       cv=3, scoring=my_custom_loss_func, iid=True)
clf_nsc.fit(X_train, y_train)
clf_nsc.best_params_

{'alpha': 0.1,
 'hidden_layer_sizes': (60,),
 'max_iter': 1500,
 'solver': 'lbfgs'}

y_true, y_pred = y_test, clf_nsc.predict(X_test)
print(classification_report(y_true, y_pred))

avg / total      precision      recall      f1-score      support
              0.98        0.98        0.98        540
```

In a typical run, both searches yield strong generalization. Notably, the nescience-based search selects a *smaller* model (e.g., a single hidden layer with 60 neurons instead of 100) and compensates with a higher `max_iter`. This reflects the role of `surfeit`: nescience tends to select the smallest model that attains high accuracy without overfitting the training data.

■

## 8.7 Auto Classification

The nescience library also includes a module for automated machine learning (for both classification and regression problems). The AutoML module returns the single model, chosen from a collection of model families, that yields the smallest nescience. For each family, the class performs a greedy search over the hyperparameters required by that family, using nescience as the objective.

The next example shows how to apply the AutoML tools.

■ **Example 8.16** In this example we use the nescience library to find the model that best describes the `digits` dataset.

```
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split

from Nescience.Nescience import AutoClassifier

(X, y) = load_digits(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)

model = AutoClassifier()
model.fit(X_train, y_train)

model.score(X_test, y_test)
[] 0.9622222222222222
```

■

If we evaluate `type(model.model)`, we will see that the library has selected a linear support vector machine as the best model for this dataset.

A key difference between the `nescience` library and other AutoML libraries is that it returns a *single* model as the best candidate, rather than an ensemble of models. In this way, the data scientist can directly reuse the result of `AutoClassifier` and proceed with the subsequent analysis.

## 8.8 Auto Regression

The `AutoRegressor` class automatically selects the best model for a regression problem. In particular, it (i) computes an optimal subset of features, (ii) selects the most suitable family of models, and (iii) tunes the model's hyperparameters—using `nescience` as the objective.

■ **Example 8.17** In this example we apply our auto-regression class to estimating house prices, using the `boston` dataset included with `scikit-learn`.

```
from fastautoml.fastautoml import AutoRegressor
from sklearn.datasets import load_boston
from sklearn.model_selection import train_test_split

(X, y) = load_boston(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)

model = AutoRegressor()
model.fit(X_train, y_train)
AutoRegressor()

model.score(X_test, y_test)
0.8763987309111113
```

If we evaluate `type(model.model)`, we see that the library has selected a linear support vector machine or a decision tree (depending on the run and dataset version) as the best model for this dataset.

A key difference between the `nescience` library and other AutoML libraries is that it returns a single model as the best candidate, rather than an ensemble of models. In this way, the data scientist can directly reuse the result of `AutoRegressor` and proceed with the analysis.

## 8.9 Time Series

In this section we study the application of the concept of miscoding to a time series and a delayed version of itself, or to a delayed version of a second time series.

Automiscoding applies miscoding to a time series and a lagged version of itself, as a function of the lag. Automiscoding estimates to what *extent* past observations of the series can explain (or help forecast) future observations. In this sense, automiscoding has a similar objective to autocorrelation in classical time-series analysis (see Section ??).

**Definition 8.9.1** Let  $\{\mathbf{x}_t\}$  be a time series composed by  $n$  samples. We define lag  $k$  *regular automiscoding* of  $\{\mathbf{x}_t\}$  as  $\hat{\mu}(\mathbf{x}_{x_{k+1}, x_{k+2}, \dots, x_n}, \mathbf{x}_{x_0, x_1, \dots, x_{(n-k)}})$ . We define in the same way the concepts of *adjusted automiscoding* and *partial automiscoding*.

In contrast to autocorrelation, automiscoding is defined for all time series, including those with a trend. Moreover, automiscoding remains interpretable in the presence of trends and can reveal seasonal components without requiring prior decomposition.

■ **Example 8.18** We examine cycles in the number of passengers of a US airline. Figure 8.11 shows monthly passengers from 1949 to 1960 (AirPassengers dataset, see References below). There is a clear yearly cycle. We apply automiscoding to confirm this analytically.

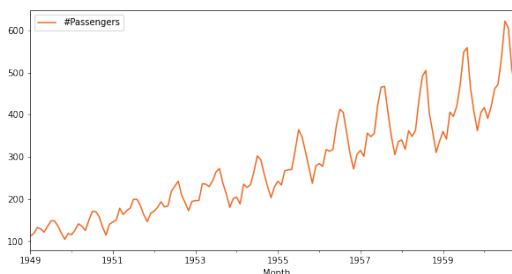


Figure 8.11: Air Passengers.

```
from nescience.timeseries import TimeSeries

data = pd.read_csv("data/AirPassengers.csv", index_col=["Month"], parse_dates=True)
X = np.array(data["#Passengers"]).reshape(-1, 1)

ts = TimeSeries(auto=False)
ts.fit(data)
mscd = ts.auto_miscode(max_lag=36)
```

As shown in Figure 8.12, the adjusted automiscoding exhibits a prominent peak every twelve months: the distance between the aligned sequences is minimized when the lag is a multiple of one year.

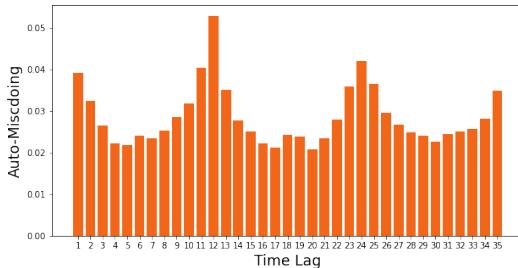


Figure 8.12: Auto-miscoding of Air Passengers.

Crossmisoding measures the relationship between a time series and a lagged version of a second time series. The objective is to detect whether the first series has temporal predictive power over the second.

**Definition 8.9.2** Let  $\{\mathbf{x}_t\}$  and  $\{\mathbf{y}_t\}$  be two time series composed by  $n$  samples each. We define lag  $k$  *regular crossmisoding* of  $\{\mathbf{x}_t\}$  and  $\{\mathbf{y}_t\}$  as  $\hat{\mu}(\mathbf{x}_{k+1}, \mathbf{x}_{k+2}, \dots, \mathbf{x}_n, \mathbf{y}_{x_0, x_1, \dots, x_{(n-k)}})$ . We define in the same way the concepts of *adjusted crossmisoding* and *partial crossmisoding*.

**■ Example 8.19** We investigate whether it is possible to predict household appliance energy consumption. The dataset (Appliances Energy Prediction; see References) contains temperature and humidity measurements from rooms every ten minutes, the appliance energy consumption, and weather variables from a nearby station.

For each feature we compute the lag in  $[1, 30]$  that maximizes crossmisoding with the target:

```
from fastautoml.fastautoml import Miscoding

X = pd.read_csv("../data/energydata_complete.csv", parse_dates=["date"], index_col="date")
y = X["Appliances"]
X = X.drop(["Appliances", "lights"], axis=1)

miscoding = Miscoding()
miscoding.fit(X, y)

best_lag = list()
for i in np.arange(X.shape[1]):
    msd = miscoding.cross_misoding(attribute1=i, min_lag=1, max_lag=30)
    best_lag.append(np.where(msd == np.max(msd))[0][0] + 1)
```

Figure 8.13 shows the results. In general, indoor measurements favor small lags, whereas weather variables (which influence indoor conditions with delay) favor larger lags.

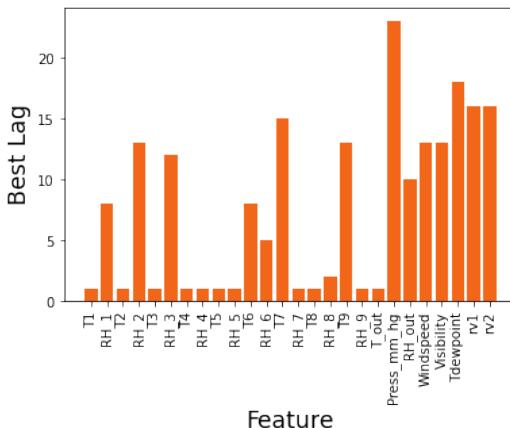


Figure 8.13: Cross Miscoding Lag



The approximation to miscoding introduced in this chapter estimates (i) the quality of individual features as predictors of a target and (ii) the quality of the training dataset as a whole. However, it does not explicitly account for redundancy among the features themselves. For instance, two features  $\mathbf{x}_i$  and  $\mathbf{x}_j$  may each have low miscoding with respect to  $y$  yet be largely redundant with one another. It remains an open question how to extend miscoding to incorporate feature redundancy in a way that stays close to the theoretical definition, is computationally efficient, and does not require prohibitively large samples.

## 8.10 Anomaly Detection

As we have seen in Section XXX, the main problem in anomaly detection is that we do not have a precise mathematical definition of what an anomaly is. In this book we propose to equate *abnormal* samples with *incompressible* samples, and to study the consequences. The essence is that learning is about finding regularities in a dataset, and data compression is about exploiting regularities. We have also seen that the best model minimizes the sum of the length of the model plus the length of the data given the model. This optimal model partitions the dataset into two disjoint subsets: a compressible part and an incompressible part. It is the latter that interests us here: being incompressible means the samples cannot be explained by the model, i.e., they are model-based anomalies under the best available model.

**Definition 8.10.1** Let  $X$  be a dataset composed by  $p$  features and  $n$  samples,  $y$  the target variable, and  $M$  a model such that the nescience  $N(\mathbf{X}, M)$

is minimal. Let  $\hat{y} = M(\mathbf{X})$  be the predictions made by the model  $M$  over the vectors of  $\mathbf{X}$ . We define the *anomaly subset* of  $\mathbf{X}$ , denoted by  $\mathcal{A}_M^{\mathbf{X}}$ , to the set of  $X$  such that  $y \neq \hat{y}$ .

The class `nescience.anomalies` allows us to identify the anomaly subset, i.e., the samples that do not match the regularity patterns found in the rest of the dataset. In Example 8.20 we apply this class to identify houses with abnormally low prices and to explain why they are cheaper.

■ **Example 8.20** We use the Boston House Prices dataset from `scikit-learn`. The dataset contains 13 predictive features (both numeric and categorical) describing different characteristics of the houses (number of rooms, age, etc.) and the target is the median value of owner-occupied homes. The dataset has 506 samples. We aim to identify houses whose prices are *abnormally low*, i.e., houses that, given their characteristics, should have a higher price.

```
from sklearn.datasets import load_boston
data = load_boston()
X = data.data
y = data.target
```

We first train a "knowledge model," i.e., the best model that explains the target variable given the predictors, without overfitting. By default, the `anomalies` class uses the AutoML capabilities provided by the `nescience` library.

```
from nescience.anomalies import Anomalies
model = Anomalies(X_type="mixed", y_type="numeric")
model.fit(X, y)
```

Finally, we select those samples for which the actual price is smaller than the price predicted by the model.

```
anomalies = model.get_anomalies("smaller")
X.shape, anomalies.shape
((506, 13), (25,))
```

Thus, there are 25 houses with abnormally low price.

The `anomalies` class also allows us to group the identified anomalies according to shared characteristics.

Let us examine which attributes best describe those abnormal houses.

```
model.get_classes(n_dims=1, an_type="smaller", filter_balancedness=True,
filter_redundancy=False, filter_repeatedAttrs=False)
```

Attribute1	Attribute2	Inertia	N	Class 0	N	Class 1	Ratio
2	None	154.953975	9	16		0.36	
4	None	0.090593	6	19		0.24	
5	None	5.002437	17	8		0.68	
7	None	20.352117	6	19		0.24	
8	None	77.611111	18	7		0.72	
9	None	41737.11111	7	18		0.28	
10	None	26.577436	13	12		0.52	
12	None	430.294828	18	7		0.72	

According to the inertia, the best attribute to classify the anomalies is attribute 4 (nitric oxides concentration). This attribute divides the abnormal houses into two clusters of sizes 6 and 19. Let us examine how price varies with this dimension.

```
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

lr = LinearRegression()
lr.fit(X[:,4].reshape(-1, 1), y)

plt.scatter(X[:,4].reshape(-1, 1), y)
plt.plot(X[:,4], lr.intercept_ + lr.coef_ * X[:,4], color="red")
plt.xlabel(data.feature_names[4])
plt.ylabel("Price")
```

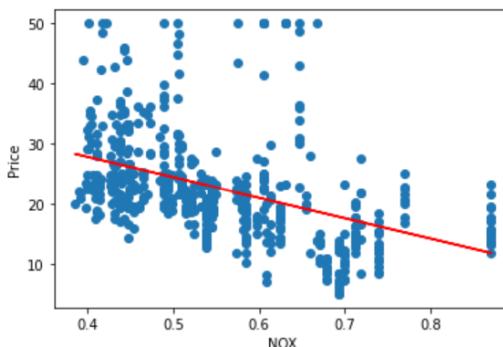


Figure 8.14: Price as a function of NOX.

The regression line suggests that house prices are lower in areas with higher nitric oxides concentration. Let us see how the anomalies split along this dimension.

```
class0, class1 = model.get_class_points(attribute1=4, attribute2=None, an_type="smaller")

plt.hist(class0)
plt.hist(class1)
plt.ylabel("Count")
plt.xlabel(data.feature_names[4])
```

```
plt.show()
```

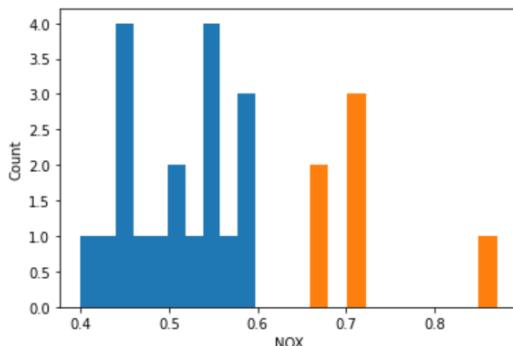


Figure 8.15: Histogram of anomalies NOX.

The analysis suggests that six of the houses have abnormally low prices because they are located in areas with high pollution levels. We can repeat the same analysis with the other attributes; however, their higher inertia indicates that class separation will be less evident. Note that there may be multiple reasons why a house's price is abnormally low.

## 8.11 Decision Trees

In the last sections we have seen how to use the concepts of miscoding, inaccuracy, surfeit and nescience to evaluate the quality of datasets and models, and to automatically select a family of models and search over its hyperparameters to find the best possible description of a topic. In particular, we have studied in detail the family of binary decision trees. The procedure used in the `fastautoml` library with trees was a mix between a classical approach (a CART algorithm combined with a cost-complexity pruning), and an evaluation of candidate trees using the minimum nescience principle. In this section we are going to see a new algorithm to derive optimal trees, both for classification and regression problems, that is entirely based on the theory of nescience. The new algorithm, by design, avoids the overfitting of the training dataset without losing accuracy, it does not require the optimization of hyperparameters, thus significantly reducing the training time, and it produces much smaller and more shallow trees than traditional algorithms, facilitating the interpretability of the results.

### 8.11.1 Algorithm Description

The following pseudocode shows the proposed algorithm to build a decision tree given a training dataset ( $\mathbf{X}, \mathbf{y}$ ). The procedure is based on a breadth first traversal of trees combined with a greedy approach. It requires a function called **BESTSPLIT()** that returns the best split of a given subset of the data into two subsets; and a second function, called **NESCIENCE()** that provides an estimation of the nescience of the current tree. The algorithm is based on two nested loops: the external **while** loop keeps a list of the candidate nodes to grow, whereas the internal **for** loop finds the best node to grow the tree. The latter operation requires to check all possible growing options and select the one that minimizes the nescience. The exit point of the algorithm is when there are no more branches to grow. We keep track of the best nescience achieved during the building process and return the associated tree.

```
def BUILD_TREE(data)

    nodesList <- list()
    tree <- BESTSPLIT(data)
    bestNescience <- NESCIENCE(tree)
    nodesList.append(tree)

    while not nodesList.empty()

        nescience <- bestNescience
        bestNode <- None
        childNode <- None

        for i <- 1, nodesList.length()

            node <- nodesList[i]

            node.child <- BESTSPLIT(node.ldata)
            tmp <- NESCIENCE(tree)
            if tmp < nescience
                nescience <- tmp
                bestNode <- i

            node.left <- None

            if nescience < bestNescience
                node <- nodeList[bestNode]
                bestNescience <- nescience
                nodesList.append(node.left)

            if not node.left.empty() and not node.right.empty()
                nodesList.remove(bestNode)

    return tree
```

The main difference of our algorithm from other decision tree building algorithms is in the way the tree is evaluated. Instead of using only accuracy

as most of the algorithms do, in addition, we take into account the complexity of the tree (surfeit) to avoid overfitting, and the quality of the subset of data used during the training process (misCoding).

## Nescience

The calculation of the nescience implemented in the algorithm is based on a Euclidean distance utility function (see Section 8.6), because that one was the one that produced the best results in the tests we have performed. For the computation of misCoding and inaccuracy, we use the same techniques that the one used in the `fastautoml` library, described in Section 8.3 and Section 8.4 respectively. For the implementation of surfeit, we use the same template to describe trees that was used in the `DecisionTreeClassifier` of the `AutoClassifier` class, and that was described in Section ???. The only difference is that we also allow equalities in the nodes (if `[attr] = [thresh]`), something not supported by the `DecisionTreeClassifier` algorithm of the `scikit-learn` library.

The generic problem of the instability of inaccuracy due to very short models, also applies to this algorithm (see Section 8.5), and the particular problem of the algorithms to build decision trees, in which the best local split might not be that one that minimizes the error (see Section ??) is also relevant in this case.

The concept of nescience is used in two different ways in our algorithm. For every iteration of the `for` loop we have to decide which one of the candidate branches of the tree we should develop. Recall that the order in which we develop the branches is important, since it might happen that one branch does not get developed because that would mean increase the surfeit without a sufficiently large decrease of the inaccuracy. The second place is at the end of the `while` loop, where we keep track of the nescience of the different building steps, to decide at the end of the algorithm which tree we return.

We treat regression problems as classification problems in which we discretize the continuous target variable  $y$  into  $n$  intervals given the number of samples, and using a uniform discretization (see Section ). Once the target variable has been discretized, we train a regular classification tree.

## Splitting Criteria

Given a subset  $\mathbf{Q} \subseteq \mathbf{X}$  we have to find a split for  $\mathbf{Q}$  such that the values of  $\mathbf{y}$  are grouped together. Recall that a split is a pair  $\theta = (j, w)$ , where  $1 \leq j \leq p$  is a feature index and  $w$  is the partition point (see Section ??). A split divides the set  $\mathbf{Q}$  into two disjoint subsets  $\mathbf{Q}_l$  and  $\mathbf{Q}_r = \mathbf{Q} \setminus \mathbf{Q}_l$ . In case of a continuous variable we have that  $\mathbf{Q}_l = \{\mathbf{x}_i \in \mathbf{Q} : x_{ij} \leq w\}$ , and if the feature is categorical

we define  $\mathbf{Q}_l = \{\mathbf{x}_i \in \mathbf{Q} : x_{ij} = w\}$ <sup>4</sup>.

In Section ?? we saw that a common splitting criteria used in practice is to minimize the weighted entropy  $\tilde{H}$  of the subsets  $\mathbf{Q}_l$  and  $\mathbf{Q}_r$ , that is, to find an split that it is minimal  $\theta^* = \arg \min_{\theta} \tilde{H}(\mathbf{Q}, \theta)$ . More explicitly, if  $\mathbf{y}$  is a target vector taking values from a set of  $k$  labels  $\mathcal{G} = \{g_1, \dots, g_k\}$  (either because is a categorical target or a continuous target that has been discretized into  $k$  intervals), and denoting the subsets of  $\mathbf{y}$  as  $\mathbf{y}^l = \{y_i : \mathbf{x}_i \in \mathbf{Q}_l\}$  and  $\mathbf{y}^r = \{y_i : \mathbf{x}_i \in \mathbf{Q}_r\}$ , and  $n_l$  and  $n_r$  are the number of elements of  $\mathbf{y}^l$  and  $\mathbf{y}^r$  respectively, we have that

$$\begin{aligned} \tilde{H}(\mathbf{Q}, \theta) &= \frac{n_l}{n} \left( - \sum_{i=1}^k \frac{\sum_{j=1}^{n_l} I(y_j^l = g_i)}{n_l} \log_2 \frac{\sum_{j=1}^{n_l} I(y_j^l = g_i)}{n_l} \right) \\ &\quad + \frac{n_r}{n} \left( - \sum_{i=1}^k \frac{\sum_{j=1}^{n_r} I(y_j^r = g_i)}{n_r} \log_2 \frac{\sum_{j=1}^{n_r} I(y_j^r = g_i)}{n_r} \right) \end{aligned} \quad (8.1)$$

In our nescience based decision tree algorithm, the splitting criteria is to minimize the total length of encoding the subsets  $\mathbf{Q}_l$  and  $\mathbf{Q}_r$  using optimal codes. We have to find the optimal split  $\theta^* = \arg \min_{\theta} \hat{K}_C(\mathbf{Q} | \theta) = \arg \min_{\theta} \{\hat{K}_{C_l}(\mathbf{Q}_l) + \hat{K}_{C_r}(\mathbf{Q}_r)\}$  where  $C_l$  and  $C_r$  are the optimal codes given the relative frequencies of the observed values of  $\mathbf{y}^l$  and  $\mathbf{y}^r$  respectively. The quantity  $\hat{K}_C(\mathbf{Q} | \theta)$  is computed as:

$$\begin{aligned} \hat{K}_C(\mathbf{Q} | \theta) &= \hat{K}_{C_l}(\mathbf{Q}_l) + \hat{K}_{C_r}(\mathbf{Q}_r) = \\ &= - \sum_{i=1}^k \log_2 \frac{\sum_{j=1}^{n_l} I(y_j^l = g_i)}{n_l} - \sum_{i=1}^k \log_2 \frac{\sum_{j=1}^{n_r} I(y_j^r = g_i)}{n_r} \end{aligned} \quad (8.2)$$

In this particular case (if we use as compression algorithm a code with optimal lengths, and continuous variables have been discretized) it turns out that both expressions are equivalent given the following relation:

$$\tilde{H}(\mathbf{Q}, \theta) = \frac{1}{n} \hat{K}_C(\mathbf{Q} | \theta)$$

We prefer to talk of encoding length instead of weighted entropy because it has an easier interpretation in the context of the theory of nescience.

---

<sup>4</sup>Ideally, for the categorical case, instead of a single feature  $w$  we should search over all the elements of the power set of the set of features  $\mathcal{P}\{1, 2, \dots, p\}$ . Unfortunately, that would imply to check  $2^p$  cases, something that is time-expensive from the computational point of view.

**R**

Strictly speaking, if we want to implement a decision trees search algorithm fully compliant with the minimum nescience principle, instead of using a total length encoding as splitting criteria, we should have computed the nescience at each split and select that one that makes it minimal. However, early experiments have shown that at local level it works better to group the values of  $y$  than to reduce the nescience. Further research is required to confirm and explain this point.

### Practical Implementation

In the web page that accompanies this book<sup>5</sup> we provide an open-source implementation of our algorithm in Python. Our software can be used together with other machine learning tools from the `scikit-learn` library, since we adhere to their API guidelines. For example, our algorithm can be used as part of an ensemble of classifiers, like the `BaggingClassifier` meta-estimator, or the results of the classification could be cross-validated with tools like `cross_val_score`. As an example, to provide a model for the breast cancer dataset, we could do something like the following:

```
from NescienceDecisionTree import NescienceDecisionTreeClassifier
from sklearn.datasets import load_breast_cancer

data = load_breast_cancer()

model = NescienceDecisionTreeClassifier()
model.fit(data.data, data.target)
print("Score: ", model.score(data.data, data.target))
```

#### 8.11.2 Algorithm Evaluation

In this section we are going to evaluate our new algorithm, and compare its performance against the well-known algorithm CART. CART, *Classification and Regression Trees*, is the de-facto standard algorithm used in the machine learning industry for the derivation of decision trees. For this particular experiment we have used the CART implementation provided by `scikit-learn`.

Figure 8.16 shows a synthetic dataset consisting of 1000 random points lying on a two dimensional plane, where all the points with an  $X_1$  attribute less than 50 are colored blue, and the rest as red. We have artificially introduced a red point, simulating a measurement error, in the blue area. The black lines correspond to the decisions performed by CART. Since the CART algorithm will not stop until all the points have been properly classified, we have to specify an expected count condition to limit the number

<sup>5</sup><http://www.mathematicsunknown.com>

of splits. The figure correspond to the tree generated by CART setting the `min_samples_leaf` hyperparameter to 5.

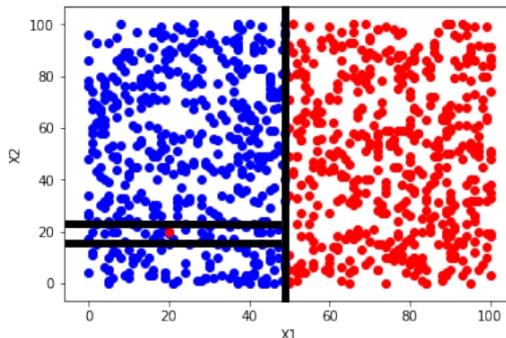


Figure 8.16: Synthetic dataset with CART algorithm splits.

The tree obtained by applying our algorithm to the dataset of Figure 8.16 can be seen in Figure 8.17. The nescience based algorithm does not try to model the error point, since the gain due to an increment in the accuracy does not compensate the surfeit introduced in the model. Recall that the algorithm stops when the total nescience of the tree, based on the measures of miscoding, inaccuracy and surfeit, does not decrease when adding new nodes to the tree. Our algorithm presents a lower sensitivity to the errors found in datasets, at least if the number of errors is small compared with the number of valid points.

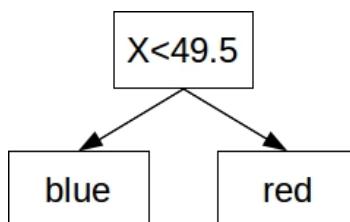


Figure 8.17: Decision tree obtained by the nescience algorithm.

Our second experiment, again with synthetic dataset, is depicted in Figure 8.18. There, we create two isotropic Gaussian blobs that partially overlap. We start with a standard deviation of 2.5 for each cluster, so they are easy to separate, and we increase the standard deviation in increments of 0.01, until we reach 4.5, which causes significant overlaps. For each value of the standard deviation, we run the experiment 100 times and we compute the average accuracy for the two algorithms using different datasets for training

(70% of the data) and testing (30% of the data). The results of this experiment are shown in Figure 8.19.

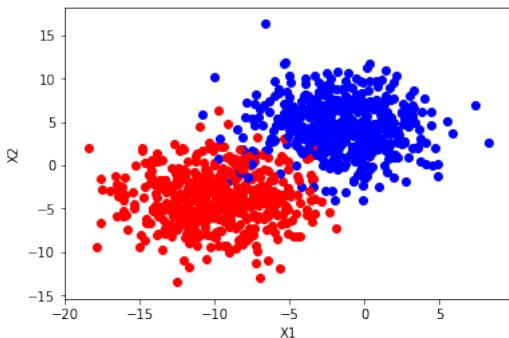


Figure 8.18: Isotropic Gaussian Blobs.

As we can see, the performance of both algorithms, in terms of accuracy, is similar. However we should note that the hyperparameter `minimum_leaf_size` of the CART algorithm has been optimized to achieve the best accuracy. For this particular experiment, the best value was achieved with a minimum leaf size of 26 points. By definition, given the fact that CART has one degree of freedom more than the nescience algorithm, it should produce better accuracy; something that it is not observed (both algorithms have a mean accuracy of 0.87).

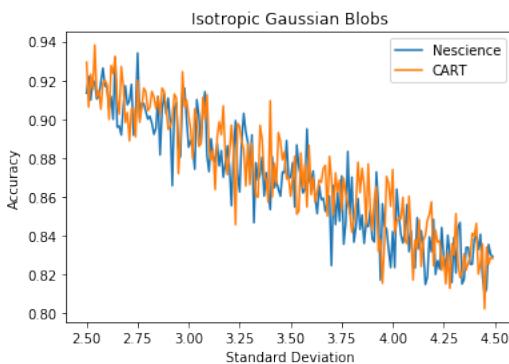


Figure 8.19: Accuracy of Isotropic Gaussian Blobs.

For each iteration of the experiment, we have also computed the average number of nodes, including internal and leaf nodes, required by the models to properly classify the clouds in the dataset. The results of this measurement are shown in Figure 8.20. Our algorithm requires an average of 4 nodes

compared to 23 nodes for the CART algorithm. Moreover, our algorithm is more stable than CART, in the sense that it produces models of similar complexity when it gets similar input datasets (a standard deviation of 0.31 compared to 3.77 for CART).

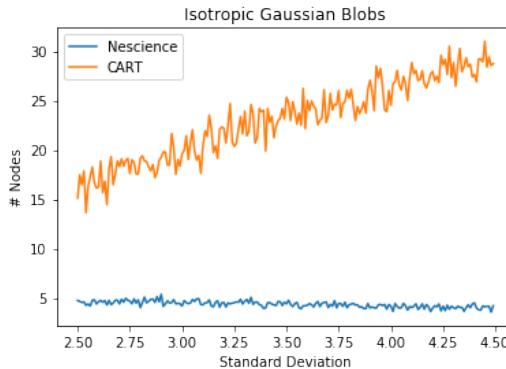


Figure 8.20: Number of Nodes.

In Figure 8.21 we show the maximum depth of the tree, defined as the longest path from the root of the tree to any of its leaves. The maximum depth of the tree is a good measure of the average time it will require for the model to provide a classification. The nescience algorithm has an average depth of 1.6 nodes, whereas the average depth yielded by the CART algorithm is 4.8 nodes.

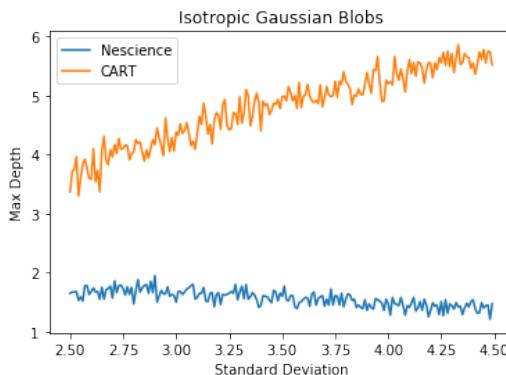


Figure 8.21: Maximum depth of the model.

The last part of the evaluation consists in comparing the performance of our algorithm and CART with a collection real datasets. More specifically, we have selected 12 well known datasets from the UCI Machine Learning

Repository. The selected datasets are: diagnosis of breast cancer (cancer), optical recognition of handwritten digits (digits), predicting protein localization sites in gram-negative bacteria (yeast), classification of NASA space shuttle data (shuttle), classification of blocks in web pages (page), segmentation of outdoor images (image), predicting the age of abalones from physical measurements (abalone), predicting the quality of red and white variants of Portuguese wine (wine), filter spam emails (spam), wall-following robot navigation (wall), classification of land use based on Landsat satellite images (landsat), and distinguishing signals from background noise in the MAGIC gamma telescope images (magic). For each dataset, we have repeated the experiment 100 times, by randomly selecting the training (70%) and testing (30%) subsets at each iteration.

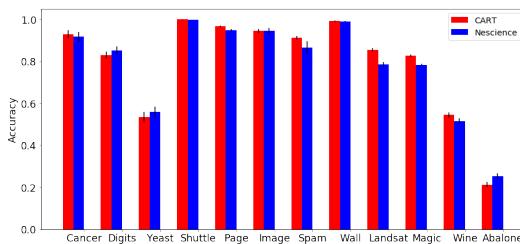


Figure 8.22: Maximum depth of the model.

In Figure 8.22 we compare the accuracy of the resulting models obtained by applying the CART algorithm and the nescience algorithm to the above datasets. In 4 of the 12 datasets, our algorithm provides better accuracy than CART. In the remaining 8 cases, the accuracy is, on average, less than 1% smaller.

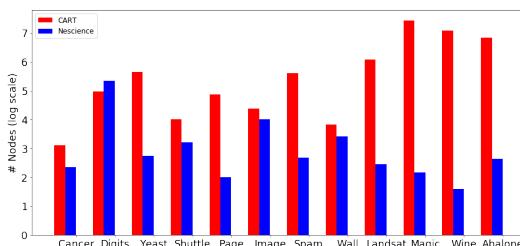


Figure 8.23: Maximum depth of the model.

In Figure 8.23 it is shown a comparison of the total number nodes (internal nodes plus leaf nodes) of the resulting models. Only for one of the datasets (digits), our model produces a slightly more complex tree than those generated by CART. In the rest of the cases, the number of nodes in

the trees generated by the nescience algorithm have between two and three orders of magnitude fewer nodes (in this figure the y axis is in logarithmic scale).

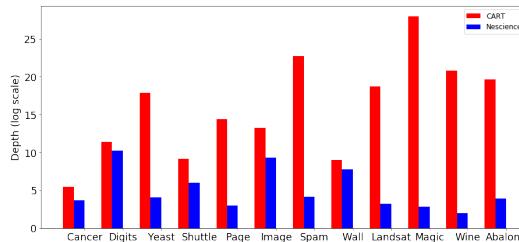


Figure 8.24: Maximum depth of the model.

Finally, in Figure 8.24 we provide a comparison of the depth of the tree of the resulting models. Our algorithm always yields a shallower tree than the CART algorithm.

We would like to mention that the nescience algorithm is highly robust with respect to the compressor selected or the nescience function implemented. In Table ??, we have apply the nescience algorithm to the datasets described above, and evaluate different alternatives for the definition of the nescience function  $N(X, M)$ : arithmetic mean  $(\mu(M, D) + \iota(X, M) + \sigma(M, D))/3$ , geometric mean  $(\mu(M, D) + \iota(X, M) + \sigma(M, D))1/3$ , harmonic mean  $3/(\mu(M, D) + \iota(X, M) + \sigma(M, D)) - 1$ , Euclidean distance  $(\mu(M, D) + \iota(X, M) + \sigma(M, D))1/3$ , sum  $\mu(M, D) + \iota(X, M) + \sigma(M, D)$ , and product  $\mu(M, D) + \iota(X, M) + \sigma(M, D)$ . The table shows limited difference between the different functions.

	Euclid	Arithm.	Geometric	Product	Addition	Harmonic
Acc.	0.758	0.784	0.803	0.803	0.784	0.81
Stdev.	0.051	0.041	0.033	0.033	0.041	0.038

Table 8.2: Comparison of nescience functions

Similarly, Table ?? shows the performance of our algorithm when using the LZMA, zlib, and bz2 compressors. We observe that all of them yield similar performance. The above results suggest that the performance our algorithm is independent of the specific choice made for either implementation aspect.

We emphasize that the CART algorithm requires to optimize a configuration hyperparameter in order to obtain good results, whereas the algorithm proposed in this book does not require from this optimization.

	<b>bz2</b>	<b>lzma</b>	<b>zlib</b>
Accuracy	0.813	0.804	0.81
Stdev	0.03	0.045	0.038

Table 8.3: Comparison of compressors

Shallower trees means faster forecasting times when the models used in production, since the number of `if-else` conditions to be evaluated is smaller. Moreover, smaller trees makes easier to interpret the results by human analysts, and much shorter training times, something very relevant in case of training ensembles of trees, like random forest or boosted trees (although the use of ensembles of models is highly discouraged by the theory of nescience, given their high surfeit).

## 8.12 Algebraic Model Selection

As it was the case for the definition of nescience based on the encyclopedic description of research topics, the nescience of structured datasets can be used to evaluate alternative descriptions of research topics (mathematical models), and to identify how far these descriptions are from an ideal perfect knowledge. This evaluation could be used to identify those topics which require further research. Moreover, the same methodology could be applied to collections of datasets to identify our current knowledge of research areas (collections of topics).

If we combine the concept of nescience of a model, with our concepts of relevance and applicability of research topics, we could apply our methodology for the assisted discovery of interesting questions to collections of datasets; a very useful methodology now that big datasets are becoming widely available.

In order to evaluate the methodology developed, we are going to apply it to a particular research topic: *Multipath Wave Propagation and Fading*. The problem at hand is to understand the effect of a propagation environment on a radio signal, such as the one used by wireless devices. The signals reaching the receiving antenna could follow multiple paths, due to atmospheric reflection and refraction, and reflection from water and objects such as buildings. The effects of these multiple wave paths include constructive and destructive interference (fading), and phase shifting of the original signal, resulting a highly complex received signal (see Figure 8.25).

In many circumstances, it is too complicated to describe all reflection, diffraction and scattering processes that determine the different paths the sig-

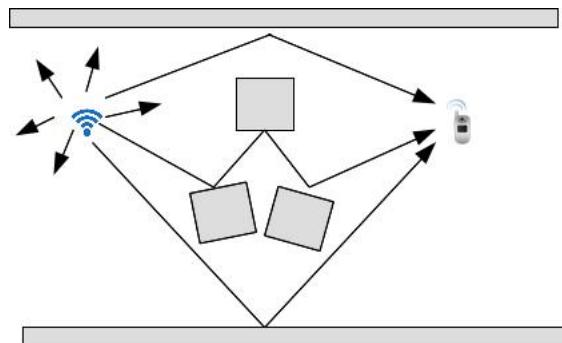


Figure 8.25: Multipath Signal Propagation

nal will follow. Rather, it is preferable to describe the probability (stochastic model) that the received signal attains a certain value. We are interested in to analyze how well these stochastic models (our current knowledge) are able to describe what happen in reality.

The *Rayleigh fading model* assumes that the magnitude of a signal will vary randomly, or fade, according to a Rayleigh distribution (the radial component of the sum of two uncorrelated Gaussian random variables). The Rayleigh probability density function of the power signal is given by:

$$P_\sigma(x) = \frac{1}{\sigma} \exp\left[-\frac{x}{\sigma}\right]$$

where  $\sigma$  is the mean of the received signals. Rayleigh fading is viewed as a reasonable model for the effect of heavily built-up urban environments, when there is no dominant propagation along a line of sight between the transmitter and receiver.

The Rice or *Rician distribution* describes the power of the received signal when the target consists in many small scatterers of approximately equal strength, plus one dominant scatterer whose individual received signal equals all that of all the small scatterers combined (there is a dominant line of sight). The probability density function of the power of the received signal is given by:

$$P(x) = \frac{1}{\bar{\sigma}} (1 + a^2) \exp\left[-a^2 - \frac{x}{\bar{\sigma}} (1 + a^2)\right] I_0\left[2a\sqrt{(1 + a^2) \frac{x}{\bar{\sigma}}}\right]$$

where  $\bar{\sigma}$  is the mean of the received signals, and it is equal to  $\bar{\sigma} = (1 + a^2) \bar{\sigma}_R$ , being  $a^2 \bar{\sigma}_R$  the power of the dominant scatterer, and  $I_0$  is the modified zeroth order Bessel function of the first kind.

An experiment (see Figure 8.26) was set up to collect a real dataset to analyze. The experiment was run on a  $135m^2$  office full of obstacles (interacting objects). The transmitter was an Odroid C1 Linux computer with a Ralink RT5370 USB Wifi adapter. The receiver was a (fixed in space) Motorola Moto G mobile phone. Data was collected using the Kismet<sup>6</sup> platform (an 802.11 layer2 wireless network detector, sniffer, and intrusion detection system), with some ad hoc, home made, software extensions, mostly for data aggregation. A total of 3,177 samples (power level measured in dBm) were collected during one hour experiment.

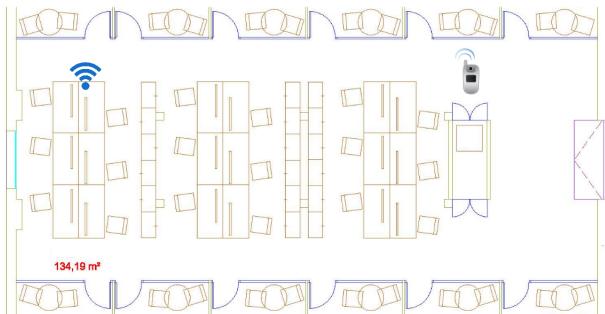


Figure 8.26: Experimental Set Up

Next table summarizes the results of applying the three considered models (uniform, Rayleigh and Rice) and the optimal encoding using a Huffman code:

Model	LDM	Nescience
Uniform	17,351	1.30
Rayleigh	13,229	0.75
Rice	11,118	0.47
Huffman	7,541	-

Table 8.4: Nescience of Models

The uniform model, that is, assuming zero knowledge about the topic covered by the dataset, has a nescience of 1.30. This value is a kind of upper level for the nescience associated with that particular topic and dataset; any model with a higher nescience should be classified as zero knowledge model. If we introduce the knowledge that in a environment with multiple obstacles the signal propagation can be described as a Gaussian process (Rayleigh

<sup>6</sup><https://www.kismetwireless.net/index.shtml>

distribution), we are able to decrease our nescience to 0.75, that is, there were a 43% improvement in our understanding of the topic. If we add the knowledge that there is usually a strongly dominant signal seen at the receiver caused by a line of sight between the antenna and the mobile phone (Rician distribution), the nescience decreases to 0.47, and so, we have achieved an additional 23% gain in our understanding. Given that numbers we can conclude that the Rayleigh model increases our knowledge with respect to the uniform model, and that the Rice model does so with respect to Rayleigh. However, the nescience of this last model is 0.47. That means that there still patterns in the dataset that are not explained by the Rice model, or what it is equivalent according to our methodology, there is still some knowledge to discover and learn.

The methodology has been applied to a dataset gathered in a single experiment under a controlled environment, since the goal of this Chapter was to provide a methodology to quantify the nescience of structured datasets, not to evaluate models for signal propagation and fading. In order to conclude that, in general, the Rice model is an improvement over Rayleigh, a more realistic experiment is required, with multiple datasets gathered in real environments.

## References

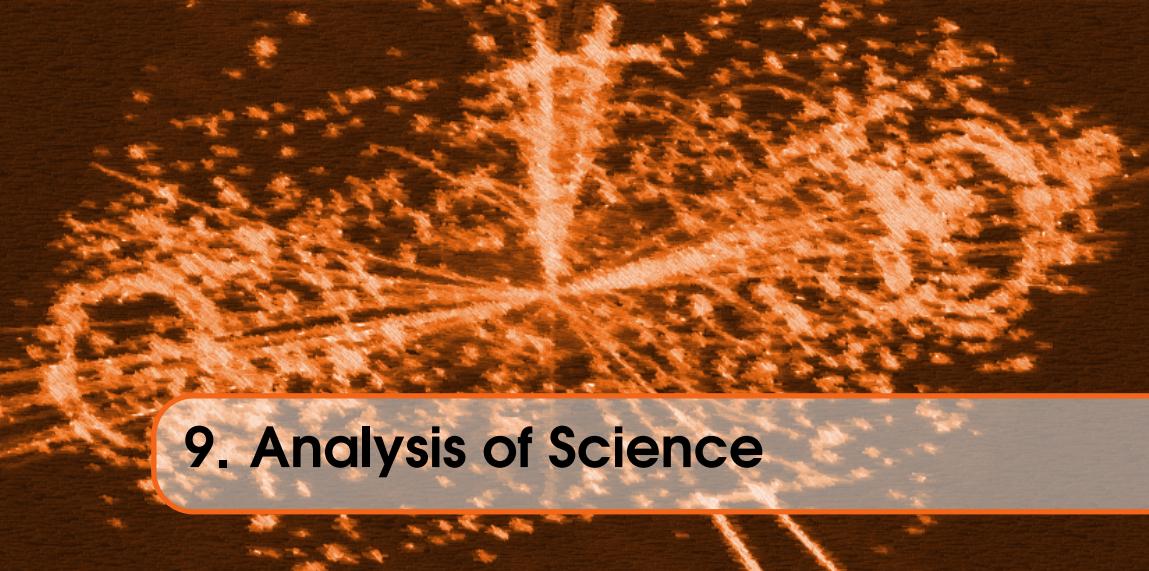
The quantity  $1 - I(\mathbf{x}; \mathbf{y}) / \max \{H(\mathbf{x}), H(\mathbf{y})\}$  has previously been proposed in [\[ferri2009experimental\]](#) as a candidate definition of *normalized mutual information*. However, to the best of our knowledge, it has not been used in practice.

A insightful description of the differences between explanation models and predictive models, how these models are used in different scientific disciplines, and what are the implications for the process of statistical modeling can be found in [\[Shm+10\]](#).

The application of the minimum description length principle to the identification of optimal decision trees have been proposed in [\[QR89\]](#), further refined and clarified in [\[WP93\]](#); however the coding method proposed by those authors is different from the one used in this book.

The Minimum Description Length [\[Grü07\]](#) and the Minimum Message Length [\[Wal05\]](#) techniques have been applied to the problem of inferring decision trees in [\[QR89\]](#), later on clarified and extended in [\[WP93\]](#), in [\[MRA+95\]](#) as a technique for pruning, and in [\[RS98\]](#), among others. Although the underlining concepts behind the cost function proposed in this chapter are the same (namely, that learning is equivalent to the capability to compress), our approach is very different from the ones described in these

works.



## 9. Analysis of Science

*Science may be regarded as the art of data compression.*

Li & Vitányi

The purpose of this chapter is to apply the theoretical framework of nescience developed in the previous chapters to the study of our current scientific knowledge. Until now, our focus has been primarily on the conceptual and mathematical foundations of nescience. We have defined and analyzed the core notions of miscoding, inaccuracy, redundancy, and nescience itself as abstract quantities, exploring their properties and interrelations. In this chapter, we will shift from theory to practice: we will operationalize these notions and use them to analyze the state of contemporary science as it is documented today.

Our empirical analysis will be grounded on Wikipedia's collection of scientific pages. Wikipedia provides a vast, open, and relatively well-structured repository of scientific knowledge that spans across virtually all research domains. While Wikipedia articles are not primary scientific sources, they offer an invaluable meta-level view of what the scientific community currently considers established knowledge, as well as how that knowledge is organized, debated, and maintained. By systematically processing this corpus, we can obtain large-scale approximations of the quantities defined in our theory and

thereby measure the current degree of understanding achieved in different parts of science.

We will begin by presenting practical methods to approximate, for any given scientific topic, its level of miscoding, inaccuracy, redundancy, and resulting nescience. Miscoding will capture the extent to which the information contained in an article fails to describe its intended subject; inaccuracy will measure the degree of controversy or error that remains unresolved; redundancy will reflect the amount of superfluous or repetitive information; and nescience will integrate these dimensions into a single estimate of our overall ignorance about the topic. This analysis will allow us to identify which topics appear to be well understood and which remain poorly understood or even actively confused.

After establishing methods for individual topics, we will extend the analysis to broader research areas. We will study how the values of miscoding, inaccuracy, and redundancy of a collection of related topics combine to produce a macroscopic picture of the maturity and clarity of an entire field. This will enable us to compare different disciplines, discover which areas are advancing rapidly, and detect fields that may be stagnating or lacking conceptual coherence.

In addition to these cross-sectional analyses, we will also perform a longitudinal study, examining the evolution of knowledge—and, symmetrically, of nescience—over time. By analyzing historical versions of Wikipedia articles, we will trace how the levels of miscoding, inaccuracy, and redundancy have changed as scientific understanding has advanced. This will provide a first empirical glimpse of the dynamics of knowledge accumulation, revealing how scientific topics move from being poorly understood and controversial to becoming stable and concise.

Finally, this chapter will also explore a long-standing philosophical question: the demarcation problem, that is, the problem of distinguishing scientific knowledge from non-scientific or pseudoscientific claims. We will propose a practical approach to this problem based on the metrics of nescience, showing how our framework can help identify topics that fail to exhibit the patterns of knowledge growth characteristic of science.

## 9.1 Describing Current Knowledge

According to the theory of nescience, evaluating our current knowledge begins by selecting a specific set of entities we want to understand. Within this framework, universal sets that include every possible entity, whether known or unknown, are not permitted. This restriction arises from the logical paradoxes associated with such sets, including Cantor's theorem, which

demonstrates that the set of all subsets has a strictly greater cardinality than the set itself, and Russell's paradox, which involves sets that are not members of themselves (see Section 2.1). Therefore, in this chapter, since our objective is to assess the current state of human knowledge, we focus on a finite set composed of entities that are already known and have been studied by science. These entities are suitable for analysis because humanity possesses at least some information about them.

The next step involves identifying the best available encoding of the selected entities as representations, typically composed of data or facts, followed by determining the most accurate descriptions we have, such as models, theories, and laws. In the theory of nescience, we maintain a clear distinction between representations and descriptions because this separation facilitates the discovery of new knowledge: either by improving the representation or by refining the description. However, since our aim here is to evaluate rather than expand our knowledge, we simplify the process by using only descriptions. These descriptions serve both as the representation and the description of an entity. Moreover, we will drop the requirement that these descriptions be computable, since very few scientific descriptions in practice are actually computable. Requiring computability would severely limit the scope of our analysis and exclude a significant portion of scientific knowledge.

For our analysis, we use descriptions derived from the collection of scientific pages from Wikipedia. Wikipedia is a free, collaborative online encyclopedia launched in 2001 that allows anyone with internet access to create, edit, and update articles on a wide array of topics. It is maintained by a global community of volunteer contributors and aims to provide reliable, neutral, and verifiable information to the public. Several features make Wikipedia particularly suitable for analyzing scientific knowledge: it maintains a transparent version history that tracks content changes over time; supports collaborative validation that reduces individual bias; enforces strict citation requirements to ensure information is verifiable and based on authoritative sources; offers broad and consistent coverage across scientific disciplines; and is dynamically updated to reflect new findings and corrections.

Wikipedia is fundamentally a tertiary source. It compiles and synthesizes information from primary sources, such as firsthand accounts of events or discoveries, and secondary sources, which interpret and evaluate those primary materials. Tertiary sources play a key role in academia by summarizing information that has already been thoroughly discussed and vetted. High-quality Wikipedia articles strive to be comprehensive, covering all major aspects of a topic in appropriate detail while avoiding undue emphasis on

minor or peripheral information.

Furthermore, Wikipedia is inherently citational. Every statement included in a Wikipedia article must be supported by a published, reliable, and verifiable source. Original research is not allowed, meaning that novel claims, hypotheses, or breakthroughs must already be published in credible sources such as academic journals or books before being included. Wikipedia also employs a form of peer review, where one or more editors review and suggest improvements to an article. While this process is not anonymous or uniformly applied like traditional academic peer review, it still acts as a quality control mechanism that enhances the coverage, clarity, and accuracy of articles through community oversight.

Wikipedia pages are written in the MediaWiki Markup Language, a simplified system for formatting text that allows users without technical knowledge of XHTML or CSS to easily edit articles. Before we can analyze the content of a scientific page, it is essential to remove all markup tags and formatting elements to isolate only the meaningful textual information. To achieve this, we used the Python library `wikitextparser`, which allows us to parse and process the raw wikitext of Wikipedia articles to extract the relevant content. In addition to stripping out the markup, the library was configured to eliminate other non-relevant elements, such as images, references, and lists, which do not contribute directly to our analysis of knowledge.

Wikipedia articles are written in the MediaWiki Markup Language, a simplified formatting system that allows users without technical expertise to contribute easily. Before we can analyze this content, it is necessary to remove all markup tags and formatting elements to isolate only the meaningful text. To do this, we used the Python library `wikitextparser`, which parses raw wikitext and extracts relevant content. In addition to removing markup, the library was configured to exclude other non-essential elements such as images, references, and lists, which do not directly support our knowledge analysis.

Wikipedia hosts a vast collection of articles covering disciplines such as history, literature, politics, entertainment, and science. Since our goal is to evaluate scientific knowledge specifically, we limited our analysis to articles categorized under the "Science" category. This subset includes topics related to the natural and formal sciences—such as physics, biology, chemistry, and mathematics—ensuring that our study remains aligned with the goal of assessing scientific understanding. We further refined this selection by excluding articles focused on individuals, organizations, awards, lists, portals, and similar metadata-oriented content. This filtering ensures that our analysis includes only those articles centered on scientific concepts, processes, or

phenomena.

Both the list of articles and the full content of each article were retrieved directly from Wikipedia using the MediaWiki API. This API provides programmatic access to Wikipedia's content, allowing automated queries for page metadata, article text in wikitext format, category information, and revision history. It ensures we are working with the most current and accurate versions of each article while offering a reproducible and efficient method for gathering large-scale data<sup>1</sup>.

## 9.2 Measuring Knowledge

As mentioned in the previous section, to evaluate our current knowledge in practice, we used the collection of scientific pages from Wikipedia, which we pre-processed to extract the relevant information for our analysis. We focused exclusively on the topic descriptions, treating descriptions and representations as equivalent. In this context, since descriptions and representations are considered the same, the inaccuracy of a topic is effectively zero, leaving miscoding and surfeit as our primary metrics of interest.



Figure 9.1: Categories in Theoretical Computer Science

<sup>1</sup>The articles analyzed in this chapter correspond to April 2025.

To demonstrate how our theory can be practically applied to assess the current state of knowledge, we selected a collection of topics listed under the Theoretical Computer Science category on Wikipedia. This includes all pages within the main category, as well as those in its subcategories and their respective subcategories, up to a depth of five levels. Figure 9.1 illustrates the Theoretical Computer Science category (Level 1) alongside all its Level 2 subcategories, such as Theory of Computation, Graph Theory, and Logic in Computer Science.

Only pages classified as "GA" have been considered. Articles in the GA class are considered complete, and they have been examined by one or more impartial reviewers. From these list, we have removed pages related to non-relevant topics, such as journals, symposiums, associations, or awards. Additionally, we have removed articles too generic pages (e.g. parallel computing) or pages describing multiple algorithms (e.g. Tropical cyclone forecasting).

Only pages classified as "GA" (Good Article) have been considered. GA articles are regarded as complete and have been reviewed by one or more impartial evaluators. From this list, we excluded pages related to non-relevant topics, such as journals, symposiums, associations, or awards. We also removed overly generic articles (e.g., Parallel Computing) and pages describing multiple algorithms (e.g., Tropical Cyclone Forecasting).

The topics analyzed cover a wide range of areas within theoretical computer science and mathematics. In graph theory, we explored topics such as graph coloring (Greedy coloring), structural properties and special classes (Snark, Perfect graph, Halin graph, Laves graph, Rook's graph, Rado graph, Well-covered graph, Pseudoforest, Component, Unit distance graph, Cop-win graph), graph transformations (Graph homomorphism, Logic of graphs), and graph-based theorems (Steinitz's theorem, De Bruijn-Erdős theorem, Turán's brick factory problem). In algorithms and data structures, we considered classical techniques like Binary search, Selection algorithm, Euclidean algorithm, Fast inverse square root, Farthest-first traversal, Linear probing, Trie, as well as more specialized algorithms such as the Gale-Shapley algorithm, Widest path problem, and data structures like the Cartesian tree. Topics in logic and computational foundations included 2-satisfiability, the Rule of inference, and the BIT predicate. In mathematical methods and theorems, we covered results such as Pick's theorem, Viète's formula, Sylvester-Gallai theorem, Sylvester's sequence, Shapley-Folkman lemma, and the Handshaking lemma. The list also touches on applied mathematics and modeling, through topics like the Dirac delta function, Earth-Moon problem, Tropical cyclone forecast model, Finite subdivision rule, and Network synthesis. Furthermore, in machine learning and computational complexity, we included

Reinforcement learning from human feedback and the Small set expansion hypothesis. Finally, some topics address computer science system design and principles, such as the Allocator (C++), Book embedding, Three utilities problem, the Commutative property, and visualization techniques like the Arc diagram. Overall, the list covers a broad spectrum of topics within the area of theoretical computer science, highlighting its diversity and interdisciplinary connections.

### 9.2.1 Surfeit

In Section 5.2, we introduced the concept of surfeit as a relative measure to quantify the unnecessary effort involved in explaining an entity using a particular description. The surfeit of a description  $d$  for a representation  $r$  is defined as:

$$\sigma(d, r) = \frac{|l(d) - K(r)|}{l(d)}$$

where  $r \in \mathcal{B}^*$  and  $d \in \mathcal{D}$  is a description of  $r$ . Intuitively, the less we know about an entity, the longer our description tends to be. As our understanding of the entity improves, we should be able to remove all redundant elements from its description.

In practice, when descriptions and representations are considered equivalent, the concept of surfeit simplifies to:

$$\rho(d) = 1 - \frac{K(d)}{l(d)}$$

This is referred to as the redundancy of a description.

The Kolmogorov complexity of the Wikipedia pages was estimated by compressing the raw text. For compression, we used bzip2, a free and open-source program based on the Burrows-Wheeler algorithm. bzip2 is commonly used in practice to estimate Kolmogorov complexity because it employs a very large compression buffer. It is well known that if the buffer size is too small, the estimation of a text's Kolmogorov complexity can be significantly distorted. For example, the gzip compressor uses a 32 KB buffer, which is too small for our purposes. In contrast, bzip2 offers a buffer size of 900 KB at its highest compression setting (compression level of 9), which is more than sufficient for compressing Wikipedia pages.

Figure 9.2 (left) depicts an histogram of the redundancy for the 50 theoretical computer science articles studied. The histogram shows that, once markup and boiler-plate are stripped, redundancy of articles is tightly concentrated between  $\approx 0.75$  and  $0.82$ , peaking near 0.80. In practical terms, even the clean prose of these pages can typically be shrunk to about three-quarters

Algorithm	Analysis of Boolean functions
Automated reasoning	Farthest-first traversal
Gale-Shapley algorithm	Widest path problem
Cartesian tree	Viète's formula
Dirac delta function	Euclidean algorithm
Fast inverse square root	Rule 184
Binary search	Linear probing
Selection algorithm	2-satisfiability
Graph homomorphism	Logic of graphs
Steinitz's theorem	De Bruijn-Erdős theorem
Earth-Moon problem	Turán's brick factory problem
Laves graph	Rook's graph
Snark	Perfect graph
Clique problem	Feedback arc set
Euclidean minimum spanning tree	Component
Unit distance graph	Halin graph
Cop-win graph	Pseudoforest
Well-covered graph	Rado graph
Network synthesis	Telephone number (mathematics)
Book embedding	Three utilities problem
BIT predicate	Pick's theorem
Sylvester-Gallai theorem	RL from human feedback
Finite subdivision rule	Sylvester's sequence
Tropical cyclone forecast model	Shapley-Folkman lemma
Theil-Sen estimator	Trie
Rule of inference	Commutative property
Allocator (C++)	Handshaking lemma

Table 9.1: Topics in Theoretical Computer Science

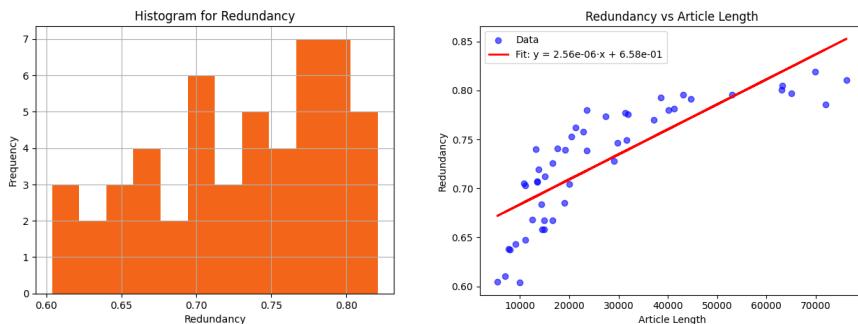


Table 9.2: Redundancy of Topics

of its original size, indicating a common stock of recurring phrases, technical jargon, and definitional patterns across the corpus. The scatter plot in figure 9.2 (right) adds a systematic dimension: redundancy rises by roughly 2-3 percentage points for every additional 10000 characters. Because every file fits inside the single 900 kB block used at bzip2's highest setting (-9), block-size effects are neutralised and the only compressor artefacts left—headers and model warm-up—explain at most a few tenths of that increase. The bulk of the slope therefore reflects the prose itself: longer articles increasingly reuse terminology, theorem-proof templates, and explanatory scaffolding, making them intrinsically more compressible. In short, both figures together show that theoretical-computer-science pages share a fairly uniform baseline of linguistic redundancy, and that this redundancy grows gradually with article length because the writing becomes more internally self-similar rather than because of any quirk of the compression tool.

Table 9.3 lists the ten topics from Wikipedia within the area of Theoretical Computer Science that exhibit higher redundancy. Each page revolves around a single, well-defined object—a classic algorithm (binary search, Euclidean algorithm, clique problem), a mathematical construct (Dirac delta, Rado graph), or a concise metatheorem (2-SAT, Shapley-Folkman lemma). Pages of this kind inevitably iterate the same symbol palette, definitions, and step-by-step explanations: every subsection restates the object, walks through an example, then re-expresses the same idea in a different formalism (pseudocode, recurrence, proof sketch, code snippet). That self-similar scaffolding gives compressors a wealth of repeated n-grams to exploit, driving redundancy up to 0.79 - 0.82 even after boiler-plate has been stripped. While the list includes some of the longer articles in the corpus (e.g. Dirac delta at 117 kB), it also contains mid-sized pages such as Rado graph (38 kB). In other words, high redundancy is not just a by-product of length; rather, it

Topic	Redundancy	Length	Compressed
Dirac delta function	0.821	117121	20969
Binary search	0.819	69838	12634
Euclidean algorithm	0.811	113242	21455
2-satisfiability	0.811	76225	14442
Shapley-Folkman lemma	0.805	63172	12323
Clique problem	0.801	63125	12574
RL from human feedback	0.797	64981	13160
Book embedding	0.796	53046	10831
Rule of inference	0.795	43158	8826
Rado graph	0.793	38626	7989

Table 9.3: Topics with higher redundancy

emerges when a compact conceptual core is expanded through multiple parallel presentations (history, intuition, formal statement, variations, applications, proof, pseudocode).

Table 9.4 lists the ten topics that exhibit lower redundancy. These articles sit in the low-redundancy tail (between 0.60 and 0.67) differ from the high-redundancy ones in both size and content mix. (i) Every entry is under 17 kB—well below the corpus median. Because bzip2’s fixed header and model “warm-up” overhead account for a larger share of such small files, their redundancy starts a few percentage points lower even before content is considered. This matches the positive slope in the scatter-plot: shorter pages naturally cluster toward the left-hand foot of the trend line. (ii) Many of these topics centre on a single formula, sequence, or code idiom (e.g. Viète’s formula, Sylvester’s sequence, Allocator). They pack unique symbols, numeric constants and one-off identifiers that appear only once or twice, so compressors see far fewer repeated n-grams than in algorithm-walk-through articles. Others, such as “Tropical cyclone forecast model” and “Finite subdivision rule”, read more like concise surveys, stringing together disparate subtopics rather than iterating one core definition from multiple angles; that heterogeneity also suppresses repetition.

Together, the list illustrates the floor of the redundancy spectrum: pages that are (i) short enough for compression overhead to matter and (ii) information-dense or eclectic enough to avoid recycling the same prose patterns never reach the 0.70-0.80 plateau seen elsewhere. Their position reinforces the earlier finding that rising redundancy with length is not merely a compressor artefact; it also depends on how much authors reuse terminology and explanatory scaffolding as an article grows.

Topic	Redundancy	Length	Compressed
Tropical cyclone forecast model	0.667	16675	5550
Allocator (C++)	0.658	14887	5085
Finite subdivision rule	0.658	14416	4933
Snark (graph theory)	0.648	11020	3882
Theil-Sen estimator	0.643	9171	3275
Earth-Moon problem	0.638	7640	2762
Halin graph	0.637	7953	2885
Viète's formula	0.61	7016	2733
Turán's brick factory problem	0.605	5427	2146
Sylvester's sequence	0.604	9994	3960

Table 9.4: Topics with Lower redundancy

### 9.2.2 Inaccuracy

Inaccuracy serves as the second metric in assessing our understanding of a research entity. The underlying idea is that the more accurate our model, the better our understanding of the entity. Formally, we calculate the inaccuracy of a description  $d$  as the normalized information distance between the original representation  $r$  and the output representation  $r'$  generated by our description  $d$ . That is, inaccuracy is quantified as the length of the smallest computer program capable of correcting the erroneous output of our model.

Inaccuracy, serving as the second gauge to measure our comprehension of a research entity, is based on the principle that the more precise our model, the better our grasp of the entity. Formally, the inaccuracy of a description  $d$  is computed as the normalized information distance between the original representation  $r$  and the output representation  $r'$  generated by the description  $d$ . Thus, inaccuracy is assessed as the extent of the smallest computer program that can rectify the incorrect output of our model.

Inaccuracy evaluates how well the output of our description aligns with the selected representation encoding the entity. However, this representation could be flawed itself, as discussed in the preceding chapter. Inaccuracy focuses solely on the description  $d$ , neglecting the potential miscoding within the representation  $r$ . Furthermore, even though it doesn't require an oracle, inaccuracy cannot be calculated for every case, so it needs to be estimated in practical situations, as we will explore in Part III of this book.

Let us consider  $r \in \mathcal{B}^*$  as a representation, and  $d \in \mathcal{D}$  as a description, where  $d = \langle TM, a \rangle$ . We then define the *inaccuracy* of the description  $d$  with respect to the representation  $r$ , denoted as  $\iota(d, r)$ , according to the following

formula:

$$\iota(d, r) = \frac{\max\{K(r \mid \delta(d)), K(\delta(d) \mid r)\}}{\max\{K(r), K(\delta(d))\}}$$

Intuitively, accuracy measures the difficulty in converting an incorrect representation  $r'$  produced by a description  $d$  into the original representation  $r$ . In essence, this involves the computation of the normalized information distance between  $r'$  and  $r$ .

In practice, and in the particular case of descriptions based on the scientific articles of Wikipedia, we cannot apply this definition, because an article of Wikipedia is not a Turing machine that produces a string based output.

Figure 9.8 left depicts a histogram showing the inaccuracy of the analyzed topics, and right side shows a scatterplot of the article lengths vs. the length of the correspondign talk pages, together with a regression line fitted to the data. The two plots suggest that, within this sample of good-quality theoretical computer science pages, talk pages chatter is generally modest and becomes proportionally smaller as articles grow. Inaccuracy, approximated as talk / (article + talk), is highly right-skewed. Roughly half the pages fall below 0.05 and more than three-quarters below 0.15, meaning their talk pages contain at most one word of discussion for every six to twenty words of main-text content. Only a small tail of articles pushes beyond 0.40, indicating a handful of topics that still attract extensive debate or revision despite their “good” label. In the scatter plot we can observe that the best-fit line ( $y \approx -2.44 \times 10^{-6} \text{length} + 0.271$ ) slopes downward, so predicted inaccuracy drops from about 0.27 for a 0-length stub to  $\approx 0.10$  at 70 kB. Empirically, short articles (< 15 kB) show the widest spread—from virtually no talk to talk pages two-thirds the size of the article—whereas long articles (> 50 kB) cluster below 0.15 with only rare outliers. In other words, once an article expands to tens of thousands of characters, the relative volume of unresolved discussion shrinks, suggesting that length correlates with maturity and consensus. Taken together, the figures imply that most good theoretical-CS pages are comparatively settled, that the few contentious ones are disproportionately short, and that growing an article tends to absorb or resolve the issues reflected on its talk page rather than magnifying them.

Table 9.9 lists topics exhibiting the lowest levels of nescience. The topics with the lowest inaccuracy values ( $\leq 0.04$ ) look like settled science. They cover classical, mathematically-rigorous results—matching algorithms (Gale-Shapley), structural graph properties (perfect graphs, Steinitz’s theorem), and well-studied data-structures or routing primitives (Cartesian tree, widest-path problem). For such topics there is little room for interpretive dispute: statements are either formally correct or not, and once a clean exposition

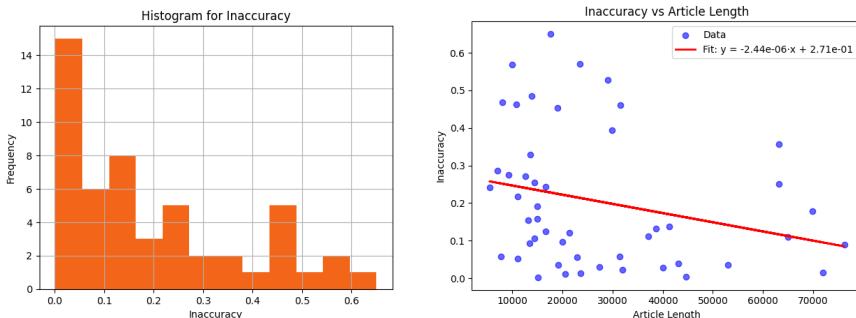


Table 9.5: Inaccuracy of Topics

Topic	Inaccuracy	Article length	Talk length
Gale-Shapley algorithm	0.001656	15072	25
Perfect graph	0.004412	44683	198
Unit distance graph	0.012203	20479	253
Rook's graph	0.013267	23576	317
Network synthesis	0.014493	71945	1058
Logic of graphs	0.022260	31933	727
Steinitz's theorem	0.027765	40094	1145
Cartesian tree	0.030849	27395	872
Widest path problem	0.035644	19128	707
Book embedding	0.036141	53046	1989

Table 9.6: Topics with lower inaccuracy

is in place subsequent editors rarely need lengthy back-and-forth. That is why some talk pages are only a few dozen words long (25 for Gale-Shapley) even when the article itself spans tens of kilobytes. Length alone does not guarantee a quiet talk page—the list includes both mid-sized texts ( $\approx 20$  kB) and the 72 kB Network synthesis article—yet all of them keep the ratio of discussion to content below 4%. Where talk does appear (e.g. 1 989 words for Book embedding or 1 145 for Steinitz's theorem) it is still dwarfed by the main text, implying that open issues tend to be minor wording tweaks, sourcing details, or peripheral expansions rather than fundamental disagreements. In short, low-inaccuracy articles are those whose subject matter is uncontroversial, formally locked-down, and already presented in a stable, comprehensive fashion, leaving editors with little need for ongoing debate.

Finally, Table 9.7 lists the topics with the highest nescience values. The

Topic	Inaccuracy	Article length	Talk length
Sylvester-Gallai theorem	0.394610	29810	19431
Pseudoforest	0.454130	18999	15806
Selection algorithm	0.461374	31578	27049
BIT predicate	0.463016	10817	9327
Halin graph	0.467671	7953	6987
Three utilities problem	0.485948	13846	13089
Fast inverse square root	0.528723	29001	32536
Sylvester's sequence	0.567808	9994	13130
Trie	0.570980	23503	31280
Commutative property	0.650826	17583	32773

Table 9.7: Topics with higher inaccuracy

articles with the highest inaccuracy scores are not necessarily the most error-ridden; rather, they are the ones whose talk pages have become arenas for protracted discussion because their subjects invite disagreement or continual refinement—foundational concepts like the commutative property and tries that draw many first-time editors, folklore-tinged algorithms such as the fast inverse square root, or theorems and graph classes (e.g., Sylvester-Gallai, pseudoforests, Halin graphs) that have multiple equivalent statements, proofs, or naming conventions. In every case the talk page approaches or even exceeds the size of the main text, so anywhere from 40% to 65% of the combined bytes are debate rather than exposition; this ratio is amplified by the fact that most of these articles are relatively short ( $\approx 10 - 30$  kB), meaning even modest amounts of conversation loom large. Together they illustrate that a high inaccuracy score chiefly signals ongoing editorial contention—about scope, presentation order, historical credit, or performance claims—rather than a simple lack of factual correctness, and that such contention is most pronounced where definitions are ambiguous, folklore collides with formalism, or broad audiences repeatedly revisit basic material.

### 9.2.3 Nescience

According to the theory of nescience, our understanding of an entity should be based on the quality of the description used to explain it. In Chapter 6, we introduced a quantitative measure of our ignorance regarding a research entity. This measure depends on the miscoding of a string-based representation of the entity, as well as the inaccuracy and surfeit of the model describing this representation. Ideally, we seek representations and descriptions that simultaneously minimize these three aspects.

In practical terms, particularly when entities correspond to topics covered by scientific Wikipedia pages, we consider only the surfeit and inaccuracy of a description. In this scenario, the representation of a topic coincides with its description, rendering the computation of miscoding irrelevant.

To address the multi-objective optimization problem posed by nescience, we apply a global criterion approach (see Section F.6.3). This method solves multi-objective optimization problems by minimizing the distance between a chosen reference point and the feasible region of the objective space. We select the origin vector  $(0, 0)$  as our reference point. The distance metric used will be the harmonic mean of surfeit and inaccuracy, defined as:

$$\frac{2}{\iota(d, r)^{-1} + \sigma(d, r)^{-1}}$$

As described in previous sections, surfeit is approximated by the redundancy of the topic's description (the Wikipedia page), while inaccuracy is estimated by the length of the corresponding Talk page.

We are using a decision maker to compute nescience based on redundancy and inaccuracy, and since these quantities do not have the same scale, it is highly convenient to apply the following additional transformation to them:

$$\mu_t = \frac{\mu_t - \min(\mu)}{\max(\mu) - \min(\mu)}$$

where  $\mu_t$  refers to the considered metric (nescience, relevance, ...).

Figure 9.8 left depicts a histogram showing the nescience levels of the analyzed topics, and right side shows a scatterplot of the article lengths vs. the nescience, together with a regression line fitted to the data. The two panels paint a mixed picture of how much “unknown-ness” remains in these 50 “good” theoretical-CS articles once both redundancy (compressibility of the prose) and inaccuracy (relative size of the talk page) are factored in. Histogram — a long, uneven tail. Roughly one-third of the articles cluster below nescience  $\approx 0.10$ , meaning they are simultaneously concise (low redundancy) and largely uncontested (little talk). Frequency then stays fairly flat through the 0.15 - 0.35 band and drops again, before a second bump appears between 0.60 and 0.75. That small peak corresponds to pages that are both highly repetitive and heavily debated—the same ones that topped the earlier redundancy and inaccuracy lists. In short, while many good-class pages are well understood, a non-trivial minority still exhibits a pronounced knowledge gap. Scatter plot — almost length-neutral. The best-fit line rises only 0.007 points for every additional 10 000 characters (slope  $\approx 7.5 \times 10^{-7}$ ), so article length alone is not a strong predictor of nescience. Short pages can swing from near-zero to 0.7, depending on how contentious they are,

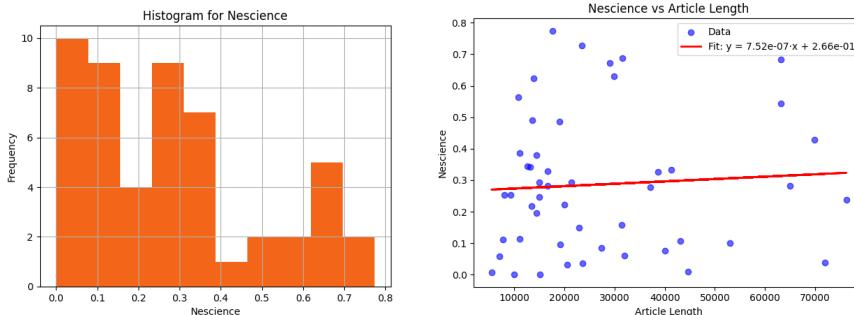


Table 9.8: Nescience of Topics

whereas long pages ( $> 50$  kB) compress more (pushing nescience up) but also attract proportionally less talk (pushing it down), leaving them parked in a mid-range around 0.25–0.35. The gentle upward tilt therefore reflects the fact that redundancy grows with length a bit faster than inaccuracy falls, but the wide vertical spread shows that editorial stability—rather than size—is the decisive factor. Overall, “good” status keeps most theoretical-CS articles in a low-to-moderate nescience zone, yet about a quarter of them still suffer from enough repetition and unresolved discussion to signal that our coverage of those topics remains less mature than the label might suggest.

Table 9.9 lists topics exhibiting the lowest levels of nescience. The articles that score best on the nescience metric fall into two complementary archetypes, each driving the harmonic mean down by making one of the two ingredients—redundancy or inaccuracy—almost vanish. The first archetype is epitomised by “Sylvester’s sequence” and “Turán’s brick-factory problem”: they are short, information-dense pages whose prose compresses poorly (normalized redundancy  $\approx 0$ ), so even a rather lively talk page cannot lift their nescience above zero. The second archetype—illustrated by “Gale-Shapley algorithm”, “Perfect graph”, “Network synthesis”, and several other graph-theoretic topics—shows the opposite pattern: their text is highly repetitive (normalized redundancy  $\approx 0.8 – 0.86$ ), but the talk pages are almost silent, so inaccuracy is near zero and the harmonic mean again collapses. Taken together, the list reveals that low lack-of-knowledge can arise either from concise, debate-heavy but non-redundant expositions or from long, internally repetitive but well-settled treatments; what matters for nescience is that at least one dimension of uncertainty is squeezed to the floor.

Finally, Table Finally, Table 9.10 lists the topics with the highest nescience values. The upper tail of the nescience scale is populated by pages that score high on both axes at once —they are repetitive and heavily de-

Topic	Nescience	N. Inaccuracy	N. Redundancy
Sylvester's sequence	0.000000	0.872117	0.000000
Gale-Shapley algorithm	0.000000	0.000000	0.499166
Turán's brick factory	0.007362	0.370945	0.003718
Perfect graph	0.008448	0.004245	0.864084
Unit distance graph	0.031744	0.016248	0.686717
Rook's graph	0.035003	0.017887	0.812327
Network synthesis	0.038635	0.019774	0.836612
Viète's formula	0.057626	0.437415	0.030845
Logic of graphs	0.061030	0.031738	0.791546
Steinitz's theorem	0.076638	0.040219	0.811139

Table 9.9: Topics with lower nescience

bated—so the harmonic mean refuses to average the trouble away. Half of the list ("BIT predicate", "Three-utilities problem", "Fast inverse square root", "Selection algorithm", "Trie", "Commutative property") shows normalized-inaccuracy  $\geq 0.70$ , signalling talk pages that rival the article in size; the same set attracts a steady stream of newcomers or folklore-laden claims, which keeps discussion alive. The other half ("Shapley-Folkman lemma", "Clique problem", "Sylvester-Gallai theorem", "Farthest-first traversal") earns its place chiefly through very high redundancy ( $\approx 0.90$  in the case of the clique problem), reflecting articles that restate definitions, proofs and examples in multiple near-duplicate forms. In every case the other dimension is still uncomfortably large ( $\geq 0.38$ ), so neither trimming repetition nor settling open talk-page issues alone would be enough to pull these topics out of the danger zone. In short, high-nescience entries are those where unresolved editorial contention co-exists with a self-similar writing style—pages that simultaneously need consensus-building and structural tightening before they can be considered well understood.

#### 9.2.4 Conclusion

Across the 50 “good-class” Wikipedia pages in theoretical computer science, our indicators tell a nuanced story about what the encyclopaedia can reveal—and obscure—about collective understanding. Talk-page activity, captured by our inaccuracy ratio, does a credible job of flagging articles whose content is still contested: pages with minimal discussion almost always cover settled, formally unambiguous material, whereas those whose talk pages rival the main text correspond to concepts riddled with naming ambiguity, folklore or pedagogical disagreement. Redundancy, distilled from

Topic	Nescience	N. Inaccuracy	N. Redundancy
Farthest-first traversal	0.490109	0.505215	0.475880
Shapley-Folkman lemma	0.543853	0.384947	0.926181
BIT predicate	0.562673	0.710692	0.465683
Three utilities problem	0.622535	0.746017	0.534125
Sylvester-Gallai theorem	0.629670	0.605318	0.656063
Fast inverse square root	0.671295	0.811908	0.572197
Clique problem	0.683333	0.548081	0.907206
Selection algorithm	0.688247	0.708163	0.669420
Trie	0.727080	0.877003	0.620932
Commutative property	0.774591	1.000000	0.632109

Table 9.10: Topics with higher nescience

compression ratios after stripping boiler-plate, complements this view by signalling how tightly authors have distilled the topic: low values align with concise, information-dense expositions, while high values mark articles that re-iterate definitions, proofs and code snippets in multiple guises. Each metric thus illuminates a different facet—editorial consensus on one side, stylistic efficiency on the other—yet neither by itself captures “knowledge quality” outright.

Their harmonic mean, which we dubbed nescience, succeeds as a triage tool because it punishes a topic the moment either dimension becomes extreme. Articles that are both verbose and hotly debated—such as those on the fast inverse square root or the clique problem—cluster at the high end, clearly pointing to areas where Wikipedia’s coverage remains unsettled. Conversely, pages with either crisp, irredundant prose or near-empty talk pages sink to the bottom, reflecting mature, well-understood subjects like the Gale-Shapley algorithm or perfect graphs. In that sense, nescience offers a practical snapshot of our residual ignorance as reflected in the encyclopaedia: it reliably highlights where clarification, consolidation or further research are still needed, even if it cannot, on its own, disentangle whether the root cause is factual dispute, pedagogical complexity, or simple editorial neglect.

### 9.3 Measuring Research Areas

In this section, we evaluate the nescience of entire research areas rather than individual topics, measuring how much knowledge each discipline encompasses. We examined all English-language “good-class” articles in six broad disciplines (biology, chemistry, mathematics, philosophy, physics,

Area	Num. Topics
Biology	214
Philosophy	33
Psychology	44
Mathematics	127
Chemistry	138
Physics	52

Table 9.11: Number of topics analyzed per area

and psychology) comprising roughly 600 pages (see Table 9.11). For each discipline, we assessed three metrics: inaccuracy, redundancy, and nescience.

Wikipedia organizes articles into categories by topic to simplify navigation and discovery. Rather than a strict hierarchy, categories form an interconnected network: each category can have multiple subcategories, and a subcategory may belong to multiple parent categories. Editors avoid circular links, where a category would indirectly include itself through its descendants. Conceptually, this structure resembles a partially ordered set in mathematics and computer science, and it allows diverse classification schemes to coexist within a unified framework. When one category clearly falls under another, it is designated as a subcategory to preserve a logical “is-a” relationship. Articles are typically placed only in their most specific relevant category to avoid redundancy. For example, the article on Claude Shannon appears in “Category:American information theorists,” not directly under the broader “Category:Mathematicians.”

The top level of Wikipedia’s category system is “Category:Contents,” which has no parent. From there, the path we have used proceeds as: Contents → Articles → Main topic classification → Academic disciplines → Science → Branches of Science (Applied Sciences, Formal Sciences, Natural Sciences, Social Sciences). Our analysis focuses on five scientific areas: Mathematics, which falls under Formal Sciences; Biology, classified within Life Sciences under Natural Sciences; Physics and Chemistry, both part of Physical Sciences in the Natural Sciences; and Psychology, included in Social Sciences. We also analyze Philosophy, categorized under Humanities within Academic disciplines. These selections ensure comprehensive coverage of both theoretical and empirical fields.

First, we looked at redundancy (see Figure 9.2), which shows how much an article repeats itself. We took out templates, lists, and other markup, then compressed each page using bzip2 at its highest setting. Pages with more repeated text stay larger after compression. We found that most good

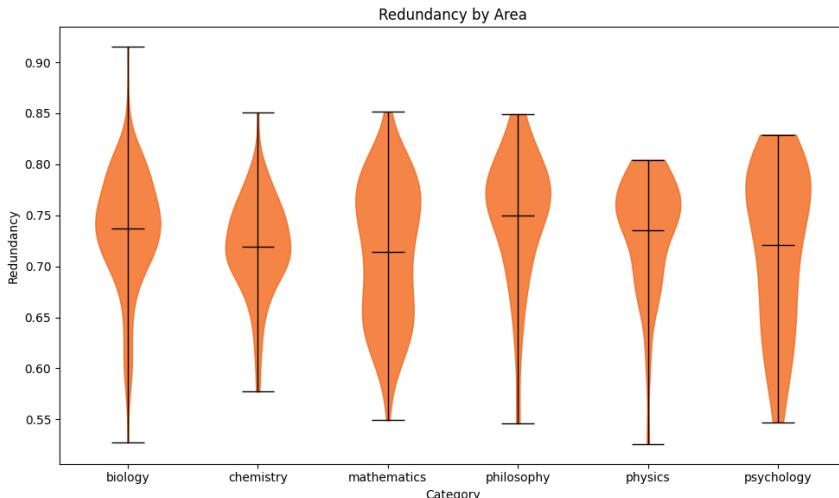


Figure 9.2: Redundancy by Area

articles in every field fall in a redundancy range of about 0.70 to 0.80, so they repeat text at similar rates. Biology had the widest range, from about 0.52 up to over 0.90, meaning some articles are very concise while others include many standard sections. Chemistry was the most consistent group, clustering around 0.72, possibly because of regular naming rules and reaction sections. Mathematics, philosophy, physics, and psychology were all in the middle, with medians near 0.75 but small differences in spread. For example, math and philosophy pages sometimes repeat proofs or arguments, raising their upper range, while psychology has some short experiment-focused articles. Physics looks like psychology but with a slightly lower median. In short, redundancy is similar across fields: most good articles keep about three-quarters of their text after compression, showing that writing style and citation rules has a higher impact than the topic itself.

Next, we estimated inaccuracy by looking at editorial debate rather than factual error (see Figure 9.3). We looked at each article's talk page size and compared it to the total size of both the talk page and the article. A larger talk page means more discussion or disagreement. The violin plot shows that most articles in every field spend only a small part of their bytes on talk pages, but a few pages have much longer debates. In chemistry, most articles keep their talk pages below about 0.1 of the total bytes, and the median is just above that. Only a few articles reach up to 0.80. Biology, philosophy, and psychology also have most pages under roughly 0.15, but these fields have taller plots in the upper half. That means a few topics, like controversial medical issues in biology or big theoretical arguments in philosophy and

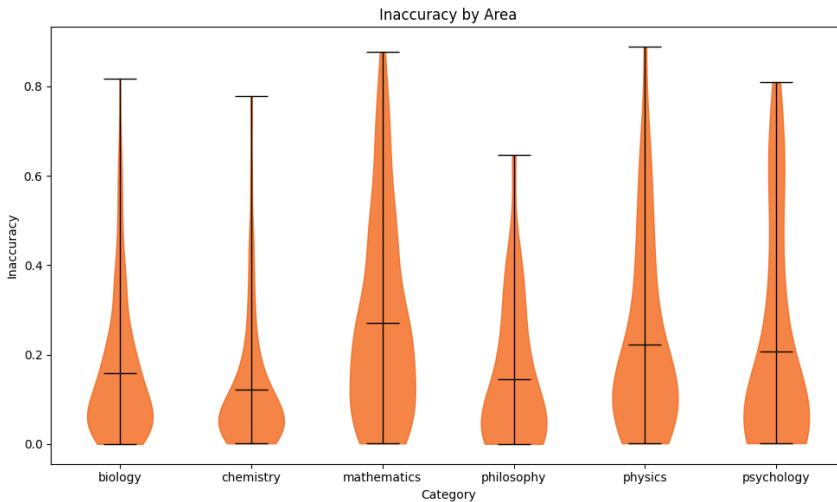


Figure 9.3: Inaccuracy by Area

psychology, spark long discussions. Mathematics and physics have the highest and widest range of debate. Their median sits around 0.25, and many pages go up to 0.5 or even 0.9, showing that some articles have as much discussion as main content. Overall, most scientific topics settle into low debate, but math and physics pages often have more back-and-forth about definitions, notation, and sources, while chemistry articles are more stable.

Finally, we combined redundancy and inaccuracy into one score, nescience (see Figure 9.4). We first scaled each measure to range from 0 to 1, then took their harmonic mean. This score highlights pages that are either very repetitive, heavily debated, or both. In the violin plot, chemistry articles have low nescience (mostly under 0.20), showing they are concise and rarely debated. Biology is similar but has a few articles with higher scores, reflecting some ongoing talk-page discussions. Mathematics has the highest median nescience (around 0.40) and reaches up to 0.90, because many pages repeat concepts and have active debates over definitions. Philosophy and physics each show two peaks. one for well-settled articles near zero and one above 0.50 for more contested topics. Psychology mainly has low nescience, but some articles still face long debates. Overall, while every field has many well-covered pages, our nescience score shows that gaps in knowledge are biggest in mathematics and, to a lesser degree, in philosophy, physics, and psychology.

Table 9.12 shows the average scores for each discipline. The numbers show that redundancy is almost the same across all six fields, ranging from 0.71 to 0.75, but inaccuracy varies more than a factor of two, from 0.12

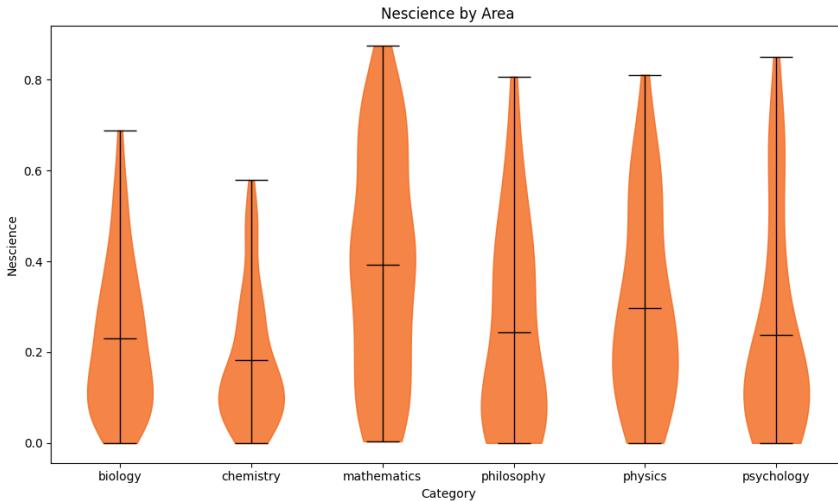


Figure 9.4: Nescience by Area

Category	Inaccuracy	Redundancy	Nescience
Biology	0.157773	0.736704	0.229765
Chemistry	0.122821	0.719500	0.183116
Mathematics	0.271307	0.714365	0.392008
Philosophy	0.144690	0.750007	0.244622
Physics	0.223447	0.735314	0.297802
Psychology	0.207355	0.721208	0.237835

Table 9.12: Average metrics by area

in chemistry to 0.27 in mathematics. Because redundancy stays nearly constant, the differences in the combined nescience score come mostly from inaccuracy. In other words, after removing repeated boilerplate, the writing style is similar across subjects; what really changes is how much editors discuss or disagree on the talk pages. For example, chemistry's low inaccuracy (0.12) gives it the smallest nescience (0.18), even with average redundancy. In contrast, mathematics has the highest inaccuracy (0.27) and therefore the highest nescience (0.39), despite slightly lower redundancy. Physics and psychology have similar redundancy (around 0.72-0.74), but physics sees more talk-page debate (0.22 vs. 0.21), so its nescience is higher than psychology's. Philosophy shows the highest redundancy (0.75) but only a medium nescience (0.24) because its talk pages are quieter. Overall, this table tells us that for high-quality articles, the main difference between fields is how much editors argue, not how the text is written.

Putting together the redundancy and inaccuracy measures gives a single nescience score. This shows that all fields have similar writing patterns, most high-quality articles compress to about 70-80% of their original length, but they differ in how much they spark online discussions. Chemistry and biology have the quietest talk pages, so they score lowest on nescience. Mathematics and physics have more debates, so they score highest, even though their writing density is similar. Philosophy and psychology fall in between. These gaps happen because each field has many settled articles and a smaller group that need more discussion. In short, Wikipedia's top science articles are usually concise and well written, but the talk pages reveal where editors still disagree most. The biggest knowledge gaps remain in math topics heavy on definitions and in physics areas open to different interpretations.

## 9.4 The Evolution of Knowledge

Our understanding of a scientific topic typically improves through sustained research efforts over time, resulting in a reduction of the topic's nescience, our lack of knowledge. Specifically, this improvement occurs through reductions in redundancy (repeated, unnecessary information) and inaccuracy (incorrect or misleading information), without significant increases in either component offsetting these improvements. New explanatory frameworks may emerge from innovative theories, refinements of existing theories, or simplified models. Ideally, each successive explanation will become more concise and accurate, removing previously redundant details and errors. Occasionally, new descriptions may temporarily lengthen as additional factual information is incorporated, causing a temporary increase in nescience. Nevertheless, the overall trend in scientific research should consistently be a progressive decrease in nescience as our knowledge advances.

To demonstrate how our theory can be used to characterize the evolution of our understanding of a scientific topic, that is, how nescience decreases with time, we have selected three highly relevant research topics as illustrative examples: graphene, CRISPR, and deep learning. Graphene is a single layer of carbon atoms arranged in a two-dimensional honeycomb lattice, known for its exceptional strength, conductivity, and versatility. CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats) is a revolutionary gene-editing technology that allows for precise, targeted modifications to DNA. Deep learning is a subset of machine learning based on artificial neural networks, enabling complex pattern recognition and predictive modeling. These topics were chosen for their scientific importance and the richness of their associated descriptions, allowing us to apply our methodology and illustrate its effectiveness.

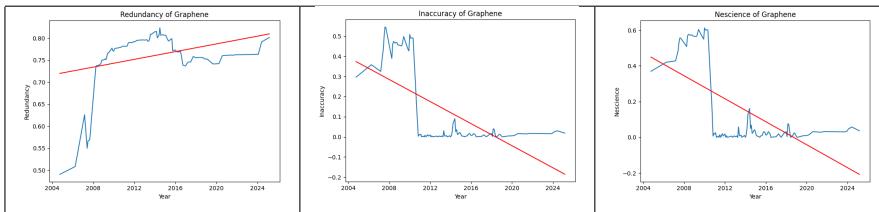


Table 9.13: Knowledge evolution of graphene

Figure 9.13 illustrates the evolution over time of redundancy, inaccuracy, and nescience metrics for the topic of Graphene. Each point on the graph corresponds to the analysis of a cleaned-up version of the Wikipedia page at a specific point in time. The early history of the graphene article shows the classic growth-and-turmoil phase of a fast-moving research topic. Between 2005 and 2010 redundancy climbed from  $\approx 0.5$  to just over 0.7 as the page ballooned with additional sections, figures and repeated explanations, while inaccuracy, our proxy for debate, spiked above 0.5. That surge coincides with the period when graphene research exploded in the literature, patents, and the popular press, so editors were actively negotiating scope, terminology and sourcing. The high redundancy plus high inaccuracy pushed the nescience curve to its peak near 0.9 in 2009-2010, signalling that the encyclopaedia's coverage lagged behind the field's rapid advances.

A sharp inflection occurs in 2011. The talk-page ratio collapses almost to zero-likely because extensive discussion was archived or split into sub-pages after the article stabilised-driving inaccuracy, and therefore nescience, down to negligible levels. From that point on the page continues to lengthen: redundancy drifts upward into the 0.78-0.82 band as definitions, production methods and applications are reiterated for different audiences. Yet talk-page activity remains a small fraction of total bytes, so nescience stays close to zero with only minor blips when new findings (e.g., commercial production techniques around 2014 or 2019) briefly rekindle debate. The negative regression slope on both the inaccuracy and nescience plots confirms a long-term convergence toward consensus, while the positive slope on redundancy simply reflects a mature, template-rich article that keeps accumulating detail without reopening fundamental disputes.

CRISPR's Wikipedia trajectory (see Figure 9.14) mirrors the arc of a scientific whirlwind maturing into mainstream knowledge. When the page first took shape (2010 - 2013) redundancy sat in the low-to-mid 0.60, reflecting a compact article that still compressed poorly, while inaccuracy-driven by an energetic talk page-hovered around 0.40. Those years coincide with the burst of laboratory discoveries that turned CRISPR-Cas9 from

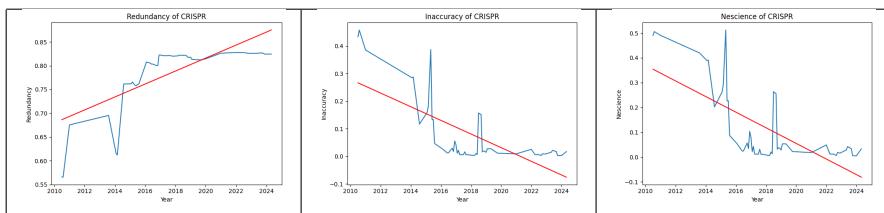


Table 9.14: Knowledge evolution of CRISPR

an obscure bacterial defence into a headline-making gene-editing tool, so editors were constantly renegotiating scope and sourcing. Both factors pushed nescience above 0.70, signalling a large knowledge gap between fast-moving science and the encyclopaedia's ability to consolidate it.

A decisive shift occurs in 2015-2016. The talk-page share collapses, dropping inaccuracy to near zero almost overnight; the page was substantially rewritten and much of the debate was archived once standard terminology, mechanism diagrams and milestone experiments had stabilised. Redundancy, meanwhile, jumps above 0.75 and keeps inching up as new sections on ethics, patents and clinical trials are added—material that inevitably repeats the core biology and acronyms. With one input shrinking and the other growing only slowly, nescience plummets and stays near the floor, apart from brief spikes that map neatly onto external flashpoints (the 2018 gene-edited-babies scandal, for example).

The overall downward regression slopes for both inaccuracy and nescience confirm a steady convergence toward consensus, while the upward slope in redundancy simply records the article's swelling, template-rich structure. By 2020 the CRISPR page looks much like that of graphene in its mature phase: highly compressible, rarely contested, and updated incrementally rather than rewritten from scratch each time the field advances.

The deep-learning article (see 9.15) traces a pattern typical of a field that shifted from niche research to global headline status almost overnight. In its early years (2011-2014) the page was compact and still being drafted: redundancy climbed from  $\approx 0.40$  to the low 0.70 as basic definitions, algorithm lists and seminal breakthroughs were added and often restated, while inaccuracy oscillated but stayed below 0.30, reflecting moderate talk-page traffic. The real turbulence began in 2017: the talk-page share abruptly jumped above 0.40 and stayed there for nearly five years, mirroring the community's debates over benchmark claims, ethics, and the hype surrounding "AI booms." Because redundancy had already settled in the mid-0.70s, this burst of discussion drove nescience to a sustained plateau above 0.85, marking the article as one of Wikipedia's most actively contested science

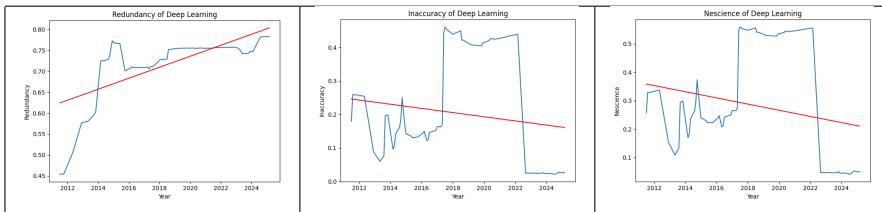


Table 9.15: Knowledge evolution of deep learning

pages during the height of deep-learning publicity.

A dramatic correction occurs in late 2022. Large chunks of deliberation were apparently archived or spun off, slashing inaccuracy to near zero almost overnight and dropping nescience with it. Since then the page has remained highly compressible (redundancy  $\approx 0.75 - 0.78$ ) but largely uncontested; only small upticks appear as new foundation-model milestones are folded in. The negative regression slopes for inaccuracy and nescience therefore chart a long-run movement toward consensus, even though the five-year plateau of high nescience serves as a reminder that Wikipedia's knowledge gap can stay wide for a protracted period when a discipline's methods, jargon and social implications are evolving faster than editors can lock down a stable narrative.

Across three emblematic case studies (graphene, CRISPR and deep learning) the time-series confirm that Wikipedia's nescience reliably chronicles the lifecycle of a scientific topic: initial growth brings a surge of redundancy as material is layered on, and a spike of inaccuracy as editors negotiate scope and sourcing; once consensus forms, talk-page activity collapses, redundancy stabilises in the high-0.70s, and nescience falls toward zero. Graphene reached this mature phase around 2011, CRISPR around 2016, while deep learning lingered in a high-gap state from 2017 to 2022 before a mass archiving of debate produced an abrupt convergence. The trajectory is thus quantifiable: rising redundancy without rising inaccuracy signals steady accretion of detail, whereas persistent high nescience marks periods when the science itself, or its social framing, is still in flux.

## 9.5 The Demarcation Problem

In this section, we propose a practical method, based on the theory of nescience, to address the demarcation problem, specifically, the challenge of distinguishing scientific from non-scientific knowledge in real-world contexts. Although demarcation is a longstanding philosophical issue, our aim is not to conclusively solve this complex problem but rather to provide insights into its nature and outline potential paths toward future solutions through

practical and operational methods.

For our experiments and analysis, we have selected six scientific topics and six pseudoscientific topics to evaluate our approach to the demarcation problem. Our analysis is based on the descriptions of these topics provided by Wikipedia and their associated Talk pages. As in previous analyses conducted in this chapter, we have preprocessed the Wikipedia pages by using the `wikibetxtparser` Python library to remove Wikimedia tags and other irrelevant elements such as tables and images.

The six scientific topics selected are: Climate Change (the long-term alteration of temperature and weather patterns), Graphene (a single-layer carbon material with extraordinary physical properties), Dark Matter (a hypothesized form of matter making up approximately 85% of the universe), Deep Learning (a subset of machine learning using neural networks with multiple layers), Lithium-ion Battery (a type of rechargeable battery widely used for portable electronics and electric vehicles), and Brain-Computer Interface (a technology enabling direct communication between the brain and external devices). These scientific topics have been selected due to their intensive research activity over the past 20 years.

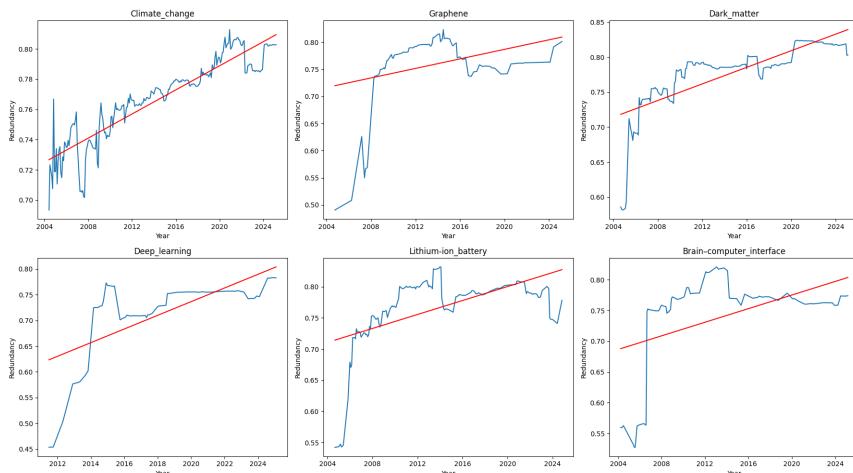


Figure 9.5: Evolution of Redundancy in Scientific Topics

Redundancy, as an approximation of the concept of surfeit, is computed, as in the previous case, by comparing the ratio of the length of a text to its compressed version. Figure 9.5 shows the evolution of redundancy for these selected scientific topics over the past 20 years. As observed, redundancy for these topics demonstrates an increasing trend, as confirmed by the computed regression line. Although one might generally expect redundancy to decrease

as our understanding improves, new discoveries and emerging knowledge frequently necessitate additional details and explanations, thus increasing redundancy in descriptions.

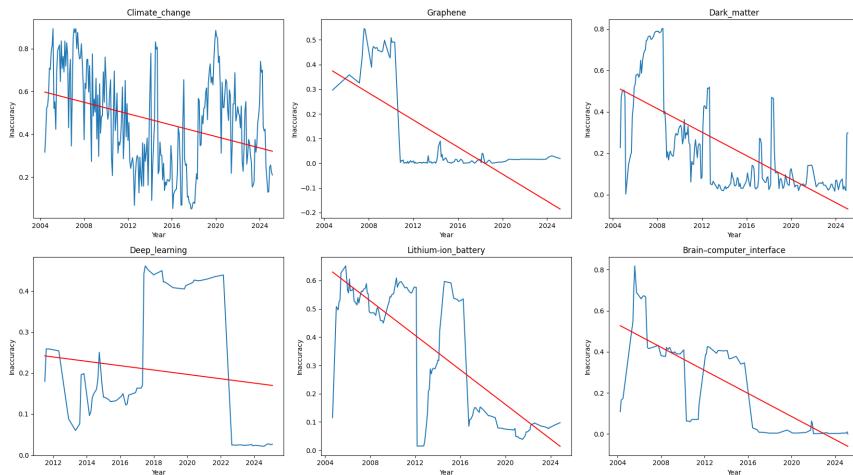


Figure 9.6: Evolution of Inaccuracy in Scientific Topics

Inaccuracy is computed based on the ratio  $\text{length\_talk} / (\text{length\_talk} + \text{length\_article})$ , where  $\text{length\_talk}$  represents the length of the Talk page for each topic, and  $\text{length\_article}$  is the length of the corresponding Wikipedia article, as done in the previous sections. Utilizing Talk pages is an effective method for approximating article inaccuracy since these pages typically contain discussions, disputes, and clarifications regarding inaccuracies or controversies in the articles. Figure 9.6 illustrates the evolution of inaccuracies for the selected scientific topics, showing a clear decreasing trend across all topics.

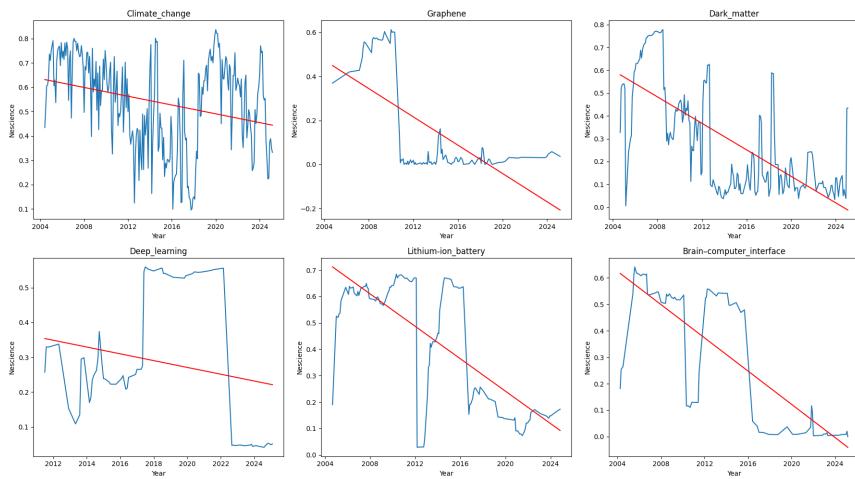


Figure 9.7: Evolution of Nescience in Scientific Topics

Finally, Figure 9.7 shows the evolution of nescience for all selected scientific topics, where nescience is estimated as the harmonic mean of the metrics of redundancy and inaccuracy. As depicted, despite the positive trend in redundancy, nescience exhibits a decreasing trend, suggesting our overall understanding of these topics improves over time.

Additionally, we have selected six pseudoscientific topics to further evaluate our demarcation method: Lunar Effect (the belief that lunar cycles influence human behavior), Water Memory (the claim that water retains a memory of substances previously dissolved in it), Astral Projection (the claimed ability of consciousness to leave the physical body and travel in the astral plane), Enneagram of Personality (a model describing personality types based on a geometric figure with nine interconnected points), Perpetual Motion (the hypothetical concept of a machine that operates indefinitely without energy input), and Dowsing (a technique claiming the ability to locate water, minerals, or other hidden substances through intuitive means). These topics represent different pseudoscientific categories—Lunar Effect (Astrology), Water Memory (Homeopathy), Astral Projection (Parapsychology), Enneagram of Personality (Numerology), Perpetual Motion (Physics-related pseudoscience), and Dowsing (Divination)—and were selected based on classifications from Wikipedia itself as pseudoscience.

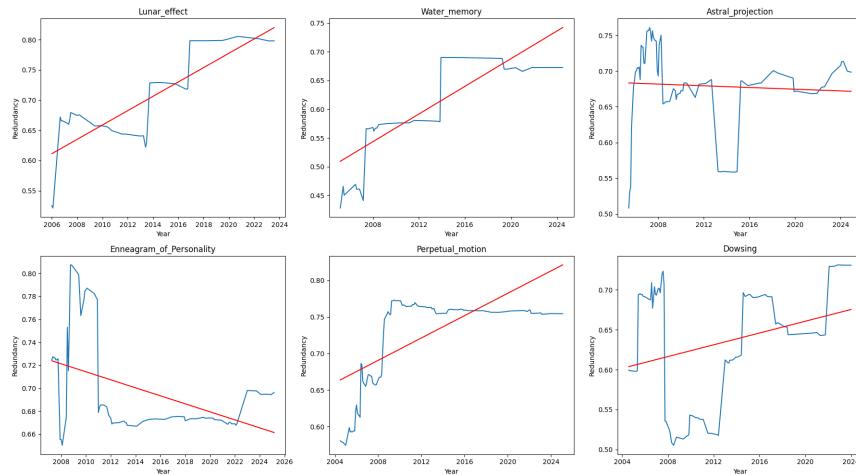


Figure 9.8: Evolution of Redundancy in Pseudoscientific Topics

Figure 9.8 shows the evolution of redundancy for these selected pseudoscientific topics over the past 20 years. As it was the case of scientific topics, redundancy for these topics demonstrates an increasing trend for "Lunar Effect", "Water Memory", "Perpetual Motion", and "Dowsing", and a rather surprising non-increasing trend for "Astral Projection" and "Enneagram of Personality". This decrease in redundancy for the latter two topics may indicate that their Wikipedia articles have undergone substantial editing aimed at streamlining or simplifying the content.

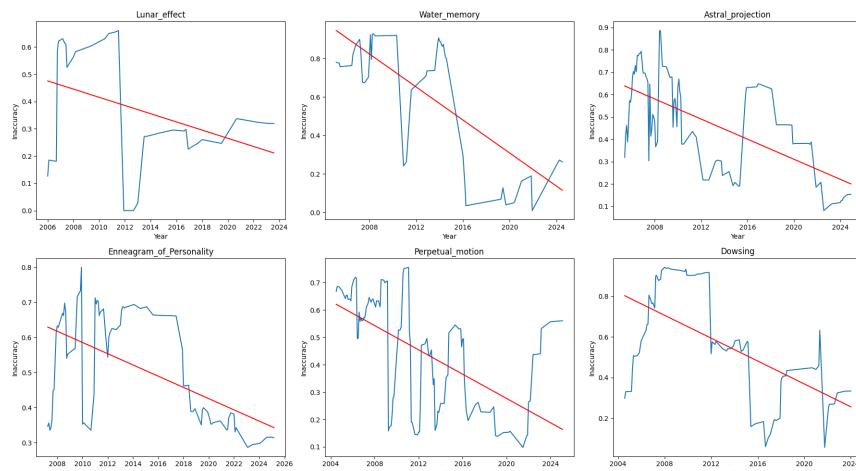


Figure 9.9: Evolution of Inaccuracy in Pseudoscientific Topics

Figure 9.9 illustrates the evolution of the estimated inaccuracies for the same pseudoscientific topics. A clear decreasing trend in inaccuracies is observable across most topics, suggesting, somewhat unexpectedly, that controversies regarding these topics have started to settle down.

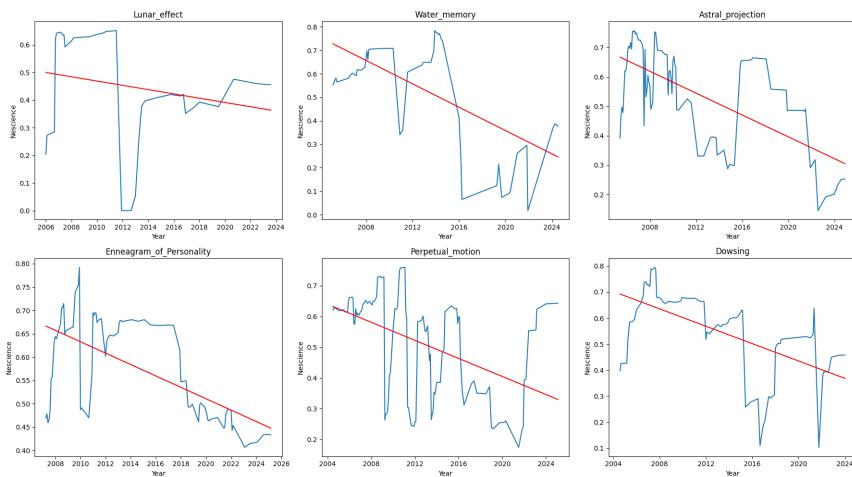


Figure 9.10: Evolution of Nescience in Pseudoscientific Topics

Finally, Figure 9.10 shows the evolution of the estimated nescience for all selected pseudoscientific topics. As expected, given the previous figures for redundancy and inaccuracy, a decreasing trend is observed for most topics, suggesting that our knowledge about this topics has, in fact, decreased with time.

Our preliminary analysis is too superficial to draw definitive conclusions. A more detailed analysis, employing randomized controlled experiments and rigorous hypothesis testing, needs to be conducted. Additionally, the approximations for redundancy and inaccuracy could be refined further (e.g., citation analysis, scientific community surveys, bibliometric analysis). Based on our initial findings, there appear to be no significant practical differences between science and pseudoscience—both communities seem capable of increasing our knowledge about their respective topics. The essential distinction between science and pseudoscience may lie in the validity and truthfulness of claims, rather than in their methodologies. Alternatively, demarcation could also be based on science's demonstrated capability to transform theoretical results into practical, real-world applications—a capability generally lacking in pseudoscience.

## References

The following is a list of references that are directly relevant for the chapter on "Analysis of Science", where we have applied the theory of nescience to real-world scientific knowledge using Wikipedia as a proxy. These references support our work in the following ways: philosophical foundations of science, empirical studies of scientific practice, metrics for information content and complexity, and critiques of scientific reliability.

[Cha13]: This book is a classic introduction to the philosophy of science, offering an accessible yet rigorous discussion of how science operates, what counts as scientific knowledge, and how theories change over time. It provides the conceptual background needed to frame your empirical analysis of scientific topics and their epistemic status.

[Mos16]: Mosterín examines the structure and evolution of scientific concepts and theories, discussing how they are created, revised, and sometimes abandoned. His perspective will support your treatment of how individual topics evolve and how nescience diminishes over time.

[Pop14]: Popper's seminal work on the falsifiability criterion and the dynamics of theory change provides the philosophical grounding for your historical analysis of scientific knowledge growth. It will also help frame your treatment of the demarcation problem within your nescience framework.

[LSW13]: Latour's ethnographic study shows how scientific facts are socially constructed within laboratories. Including it emphasizes that Wikipedia articles reflect not only facts but the negotiated outcomes of scientific communities, which ties directly to your measurement of inaccuracy and controversy.

[Ioa05]: Ioannidis empirically argues that much published science is unreliable due to biases, small samples, and statistical issues. This is crucial to motivate the need for metrics like inaccuracy in your analysis of scientific content.

[Van80]: Van Fraassen's constructive empiricism offers a philosophical counterpoint to scientific realism, questioning what it means for a theory to be considered "true." This perspective enriches your discussion of what your nescience metrics are actually measuring in terms of epistemic commitment.

[Sup02]: Suppes' work on the formal representation of scientific theories will support your transition from individual Wikipedia articles (topics) to aggregated structures (research areas), helping ground your methodology conceptually.

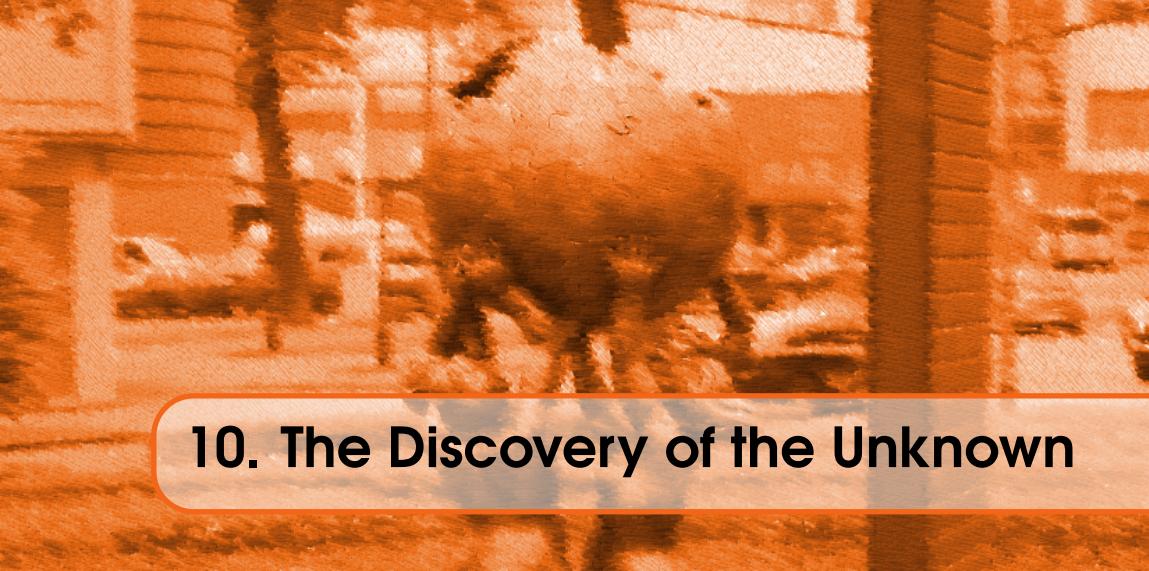
[Li+04]: This paper introduces the normalized compression distance (NCD), a practical information-theoretic tool for measuring similarity between data objects. It underpins your empirical estimation of redundancy and miscoding in Wikipedia articles.

[LV13]: This is the definitive reference for Kolmogorov complexity, the

theoretical foundation for your measures of miscoding, redundancy, and ultimately nescience. It provides both the mathematical background and many applied examples.

[CAO+05]: This article warns about practical issues when using real compressors to approximate Kolmogorov complexity. It is highly relevant for the methodological section where you will approximate your metrics on Wikipedia data.





## 10. The Discovery of the Unknown

*To be surprised, to wonder,  
is to begin to understand.*

José Ortega y Gasset

In this chapter, we are going to see how to apply in practice our methodology for the assisted discovery of interesting research questions. As was the case in the previous chapter, in which we studied the concept of nescience from a practical point of view, here we will focus on the empirical side of our framework, showing how the theoretical concepts introduced earlier can be operationalized and used to guide actual research activities.

In the first part of this chapter, we will examine how to approximate the two new metrics introduced: relevance and applicability. The relevance of a topic will be estimated from the number of web pages on the Internet that link to the topic's page on Wikipedia (external links). Applicability will be estimated from the number of links among Wikipedia's scientific pages themselves (internal links). We will illustrate both metrics with examples drawn from the set of topics that make up the research area of theoretical computer science.

Next, we will describe how to apply our methodology for the discovery of interesting questions and present several examples of new research questions

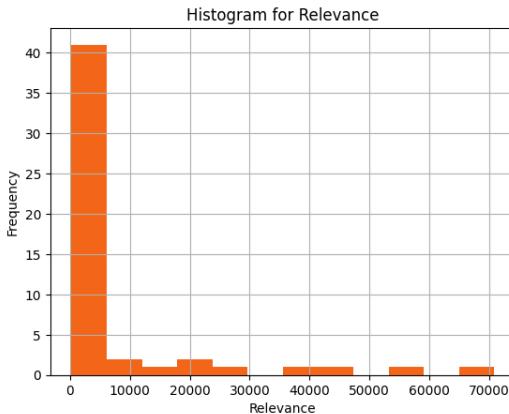


Figure 10.1: Relevance of Topics.

that, in principle, could be addressed by science. Some of these questions will be intradisciplinary (arising within theoretical computer science) while others will be interdisciplinary, combining theoretical computer science with philosophy or biochemistry. Finally, we will derive a number of new research topics which, according to our subjective interpretation of the combinations found, are sufficiently promising to warrant further investigation. We will also assess whether these proposed topics satisfy the criteria for interesting questions established in Chapter 7.

In the last part of the chapter, we will apply the set of metrics defined for the classification of individual research topics to entire research areas. This will allow us to estimate the interestingness of different disciplines both as sources of new problems and as sources of useful tools for solving open problems. These metrics will also enable us to compare the relative merits of different fields of knowledge, and to highlight some examples of research areas that appear to be in decline.

## 10.1 Relevance

In Definition ?? we introduced the concept of the relevance of a research topic as a measure of the impact that this topic could have on people's lives. The idea was that the higher the relevance, the greater its potential as a source of interesting questions, since it would address problems that affect many people. Relevance was defined as the degree of the research topic in the relevance graph, a bipartite graph connecting topics and people (see Definition ??). Of course, this relevance graph is a mathematical abstraction that is very difficult to compute in practice, since we do not have information

Topic	Relevance
Dirac delta function	70,869
Rule of inference	58,356
Fast inverse square root	46,570
Euclidean algorithm	38,032
Trie	29,116
Binary search	23,246
Reinforcement learning from human feedback	22,294
Commutative property	17,322
Gale–Shapley algorithm	11,251
Pick's theorem	7,255

Table 10.1: Most relevant topics

about how people are affected by each topic.

As a practical proxy for the relevance of a topic, we use its Wikipedia pageviews over the last two months (i.e., the sum of daily visits during the most recent 60 complete UTC days). The intuition is that the more relevant a topic is, the more frequently people will consult its Wikipedia article. Our goal is not an absolute notion of relevance but a measure of relative relevance across topics. Moderate under- or over-estimation is acceptable provided it affects all topics roughly uniformly—for example, due to seasonal effects or platform-wide trends—so that comparative rankings remain stable. Further work is needed to quantify how well pageview-based relevance aligns with the theoretical construct of relevance and to correct for known biases (e.g., transient news spikes or bot traffic).

In Figure 10.3, which shows the histogram of the relevance of all good articles in the area of theoretical computer science, total traffic over the last 60 days is 393,672 pageviews, with an average relevance of 7,719. The histogram shows a classic head–long-tail pattern: most topics cluster in the lowest bins (a few hundred to a few thousand pageviews), while a small number of articles receive very large traffic, extending the right tail out past 70k. The top three pages—Dirac delta function (70,869), Rule of inference (58,356), and Fast inverse square root (46,570)—collect 44.7% of all views. This strong right-skew implies that the mean relevance will sit far above the median and be dominated by a handful of high-visibility pages (e.g., widely taught or broadly cited topics). The sparsity of mid-range bins suggests a sharp separation between everyday background interest and a small set of standout articles; outliers are genuine and not just binning artifacts. For modeling purposes we will use log-transformed counts to stabilize variance.

Topic	Relevance
Well-covered graph	228
BIT predicate	242
Halin graph	360
Laves graph	462
Tropical cyclone forecast model	510
Book embedding	516
Turán's brick factory problem	520
Farthest-first traversal	626
Cop-win graph	647
Finite subdivision rule	662

Table 10.2: Least relevant topics

Table 10.1 lists the ten most relevant topics according to this metric. For each topic, it shows its raw relevance (number of external links). The top ten articles account for 82.4% of the total page views. Overall, the list aligns reasonably well with what one would expect to attract broad attention around theoretical computer science. Judged purely by intuitive relevance (irrespective of formal classification), most items in the list plausibly merit attention from a theoretical computer science audience: Euclidean algorithm, Binary search, Trie, and the Gale–Shapley algorithm are canonical in algorithms, data structures, and algorithmic game theory; Rule of inference underpins proof techniques used throughout complexity and verification; Fast inverse square root is a celebrated algorithmic trick illustrating implementation–theory trade-offs; and Reinforcement learning from human feedback reflects an influential line of modern learning that interacts with algorithmic and statistical theory.

A few highly viewed pages are better understood as adjacent or cross-disciplinary anchors rather than core theoretical computer science articles: Dirac delta function (analysis/mathematical physics), Commutative property (elementary algebra), or Pick's theorem (discrete geometry). As it was explained in Section 9.3, Wikipedia's hierarchical category system is not a strict ontological taxonomy with hard area–subarea boundaries; it is closer to a bibliographic organization in which topics are linked across neighboring fields when readers, editors, or sources commonly associate them. In this sense, it is unsurprising—and entirely reasonable—that a few non-pure TCS pages appear alongside canonical entries such as the Euclidean algorithm, binary search, tries, and the Gale–Shapley algorithm.

Table 10.2 lists the ten least relevant topics according to this metric. The

bottom ten articles account for 1.2% of the total page views. Well-covered graph, Halin graph, Cop-win graph, and Book embedding speak to structural and algorithmic themes in graph theory and graph drawing; Turán’s brick factory problem and Finite subdivision rule touch classic extremal and geometric constructions; Farthest-first traversal is a recognizable clustering/approximation heuristic; and the BIT predicate underlies descriptive complexity formalisms that, while niche, are conceptually central. By contrast, Laves graph and Tropical cyclone forecast model draw smaller, more domain-specific audiences yet still intersect algorithmic or structural concerns. The common thread is their narrow scope: these topics serve expert or subfield-focused needs rather than broad readership, which naturally depresses pageviews. Consequently, pageviews are a sensible proxy for broad relevance (what many people consult), that match or intuition of how many people could be affected by the topic.

Taken together, both lists are consistent with what we would expect people to consult when engaging with theoretical computer science ideas. Using Wikipedia pageviews as a proxy for relevance is therefore a sensible first-order approximation: it captures widespread interest in core topics, is easy to compute at scale, and yields stable comparative rankings over multi-week windows, while benefiting from standard safeguards (e.g., log-scaling, smoothing, and outlier checks) to mitigate transient spikes or teaching-cycle effects.

## 10.2 Applicability

Applicability measures how likely it is that a research topic can be applied to solve open problems. The underlying idea is that if a tool has already been applied to solve multiple problems, then there is a high probability that it can be used again to solve new ones. Formally, the number of problems to which a tool has been applied is computed using the applicability graph (see Definition ??), and the applicability of a topic is defined as its out-degree in this graph (see Definition ??).

In practice, we have approximated the applicability graph by using the network of internal links between the scientific pages of Wikipedia. Specifically, we approximate the applicability of a topic by counting the number of Wikipedia pages that link to its page (using the “*What links here*” facility, which lists the pages that link to—but do not redirect to—the current page).

Figure ?? shows a histogram of the applicability values for the selected set of topics. The histogram’s shape is dominated by an enormous spike in the very first bin and a sparsely populated, elongated right-hand tail. Roughly three-quarters of the theoretical computer science topics attract fewer than

about 50 incoming Wikipedia links, while only a few reach the triple-digit range and just one or two exceed 300. This steep drop-off is the hallmark of a power-law (or at least heavy-tailed) distribution: applicability, as measured here, is concentrated in a tiny set of "super-connectors," while the median topic sees only modest reuse. In practical terms, the average link count is a misleadingly optimistic figure—pulled upward by a few giants—whereas the typical topic remains relatively niche.

Such inequality has several implications. First, it reinforces the idea that a small core of foundational concepts underpins a large share of problem-solving across the field; investing effort in those hubs yields the greatest leverage. Second, the near-emptiness of the mid-range suggests that rising topics face a kind of applicability "valley of death": they must bridge a substantial gap before joining the elite group of widely referenced concepts. Finally, the long tail highlights opportunity—numerous specialized notions lie at low link counts, potentially poised for breakout if new cross-disciplinary applications emerge.

Table 10.3 lists the ten most applicable topics according to this metric. The backlink data confirm the long-tailed picture hinted at by the histogram: applicability, at least as proxied by Wikipedia's "What links here" counts—is highly unequal. A single super-hub (the *Dirac delta function*, 625 links) dwarfs the rest, while only a handful of topics even cross the 200-link mark. This means that when researchers look for broadly reusable tools, the "return on attention" is greatest in a very small set of concepts that already function as connective tissue across many areas.

Equally striking is *who* those hubs are. Alongside classic algorithmic staples such as the *Euclidean algorithm*, *Trie*, and *Binary search*, we see foundational math and logic notions (e.g., *Rule of inference*, *Commutative property*), a fast-rising AI methodology (*Reinforcement learning from human feedback*), and even a domain-specific meteorology model. In other words, high applicability favors breadth rather than disciplinary purity: ideas that spill into multiple conversations—whether introductory, theoretical, or applied—accumulate the most links. For anyone mapping future research bets, this suggests monitoring backlink growth over time (to catch newcomers like RLHF early) and normalizing counts by sub-field size to filter out topics whose popularity is driven by narrow, self-referential clusters.

## 10.3 Maturity

As introduced in Definition ??, *maturity* is the opposite of *surfeit*. While *surfeit* measures the amount of redundant or superfluous information present in our current description of a topic, *maturity* quantifies how concise and

Topic	Internal Links
Dirac delta function	625
Rule of inference	409
Reinforcement learning from human feedback	351
Euclidean algorithm	343
Trie	244
Telephone number (mathematics)	228
Commutative property	214
Tropical cyclone forecast model	195
Cartesian tree	155
Binary search	116

Table 10.3: Applicability of topics

efficient that description has become. In other words, a topic is considered mature when our understanding of it is sufficiently advanced that its essential content can be stated without unnecessary verbosity. High maturity thus indicates that knowledge about the topic has been largely consolidated, leaving little room for removing redundant parts from its description.

The maturity of a topic is estimated based on the length of its Wikipedia article (only the textual content) and the length of a compressed version of that same text. The intuition behind this approach is that the more mature a topic is, the less redundant information its article will contain, and therefore the better it can be compressed. Topics that are still immature often accumulate loosely organized or repetitive content, which resists compression. By comparing the original and compressed lengths, we obtain a practical proxy for surfeit (as the ratio between the two lengths), and therefore for maturity as its inverse.

Table 10.4 lists the ten most mature topics according to this metric. For each topic, it shows its maturity value and its normalized version. Well-classified topics—those that our intuition tells us are well understood—include *Read-only right moving Turing machines*, *Crossing sequence (Turing machines)*, and perhaps the *P*” language. Other topics that might be misclassified include *communication X-Machine*, *Power DEVS*, *MPIR*, and *constraint automaton*.

## 10.4 Interestingness

The *interestingness* of a topic, when considered as a tool, measures the likelihood that this tool can be applied to other problems. Table 10.5 shows the average applicability and average maturity of each of the selected areas,

Topic	Maturity	Norm.
Carry operator	5.34	1.00
Binade	4.54	0.99
Comm. X-Machine	3.01	0.97
PowerDEVS	2.35	0.94
MPIR	2.00	0.92
Constraint automaton	1.84	0.90
RO right moving TM	1.73	0.89
P"	1.71	0.89
Crossing sequence (TM)	1.63	0.88
Microsoft Binary Format	1.53	0.86

Table 10.4: Maturity of topics

Research Area	Applicability	Maturity	Tools
Sociology	$1.00 \times 10^{-3}$	$2.93 \times 10^{-3}$	$3.09 \times 10^{-3}$
Biology	$9.20 \times 10^{-4}$	$4.65 \times 10^{-3}$	$4.74 \times 10^{-3}$
Chemistry	$3.11 \times 10^{-3}$	$5.01 \times 10^{-3}$	$5.90 \times 10^{-3}$
Psychology	$1.14 \times 10^{-3}$	$6.91 \times 10^{-3}$	$7.00 \times 10^{-3}$
Mathematics	$9.32 \times 10^{-3}$	$9.47 \times 10^{-3}$	$1.32 \times 10^{-2}$
Epistemology	$1.55 \times 10^{-3}$	$1.75 \times 10^{-2}$	$1.76 \times 10^{-2}$
Computer_science	$9.93 \times 10^{-3}$	$1.90 \times 10^{-2}$	$2.15 \times 10^{-2}$

Table 10.5: Interestingness of Areas as Tools

together with their average interestingness as a source of tools. The table largely fits our intuitive expectations about which areas are more important as sources of tools: computer science appears as the most interesting area, while sociology is the least. The only surprising case is epistemology, which emerges as a source of very interesting tools, even more so on average than mathematics. This anomaly may be explained by the high proportion of poorly written articles in this area, which artificially lowers maturity and thereby increases the calculated interestingness.

Table 10.6 shows the relevance and nescience of the selected areas, as well as their interestingness as sources of problems. Once again, the results align with our intuition: sociology is the area with the highest number of interesting problems, while mathematics is the area with the fewest.

Table 10.7 lists the ten most relevant topics according to their interestingness as tools. Out of these ten, only two (*ternary numeral system* and *recursion*) appear in the lists of the top ten most mature or most applicable

	Relevance	Nescience	Problems
Mathematics	$4.22 \times 10^{-2}$	$3.51 \times 10^{-1}$	$3.53 \times 10^{-1}$
Computer_science	$2.35 \times 10^{-2}$	$4.43 \times 10^{-1}$	$4.44 \times 10^{-1}$
Chemistry	$5.95 \times 10^{-2}$	$4.66 \times 10^{-1}$	$4.70 \times 10^{-1}$
Biology	$3.85 \times 10^{-2}$	$4.75 \times 10^{-1}$	$4.77 \times 10^{-1}$
Psychology	$5.06 \times 10^{-2}$	$5.28 \times 10^{-1}$	$5.31 \times 10^{-1}$
Epistemology	$4.54 \times 10^{-2}$	$5.30 \times 10^{-1}$	$5.32 \times 10^{-1}$
Sociology	$4.21 \times 10^{-2}$	$5.43 \times 10^{-1}$	$5.44 \times 10^{-1}$

Table 10.6: Interestingness of Areas as Problems

Topic	Interestingness
GNU MPAL	0.49
Ternary numeral system	0.48
IEEE 854-1987	0.47
Arithmetic logic unit	0.43
Recursion	0.42
Barrel shifter	0.42
State space	0.42
Abstract machine	0.41
Computational model	0.39
Arithmetic overflow	0.39

Table 10.7: Interestingness of Tools

topics; the rest are new. Among these we find clear examples of tools, such as the *GNU Multiple Precision Arithmetic Library* and the *standard for radix-independent floating-point arithmetic* (IEEE 854-1987). Other cases are less clear, such as *arithmetic logic unit*, *barrel shifter*, or *arithmetic overflow*. Topics that match our intuitive notion of a tool include *recursion*, *state space*, *abstract machine*, and *ternary numeral system*. Finally, some topics, like *computational model*, are too broad to be meaningful in the context of a single research question.

Table 10.8 lists the ten most interesting topics as problems. Examples that align with our intuitive notion of problems—concepts that are both highly relevant and poorly understood—include *arithmetical hierarchy*, *halting problem*, *floating point*, *quantum computer*, and *computable function*. Interestingly, the topic *recursion* appears in both lists: as a tool it refers to the general concept, while as a problem it refers to the specific implementation of recursion in computer science. The classification of *regular expression* as

Topic	Interestingness
Arithmetical hierarchy	0.72
Regular expression	0.68
Computability theory	0.65
Halting problem	0.65
Recursion (CS)	0.64
Lambda calculus	0.63
Floating point	0.61
Quantum computer	0.57
Computability	0.55
Computable function	0.55

Table 10.8: Interestingness of Problems

a problem, rather than as a tool, can be explained by the length and detail of its Wikipedia article, which provides an extensive reference-level description. This observation raises the broader methodological question of how to distinguish introductory from reference articles when analyzing Wikipedia data. Finally, some topics, such as *computability theory*, *lambda calculus*, and *computability*, are too broad to be treated as individual problems.

## 10.5 Interesting Research Questions

Before computing the new interesting questions, it is highly convenient to normalize the metrics of the topics involved in the study. Otherwise, a small number of topics with extreme values could dominate the results. For the normalization process we have used the Box-Cox method, which is based on the identification of the best transformation from a family of power transformations. This method is particularly useful when the data is skewed or exhibits heteroscedasticity, as it stabilizes variance and makes the distributions of the metrics more comparable across topics.

A major challenge in identifying topics as tools, that is, topics with very high maturity (or, equivalently, very low nescience), is to distinguish between cases where a description is short because the topic is genuinely well understood (for example, a mathematical theorem), and cases where the article is short simply because it is unfinished or poorly written. Our approach uses Wikipedia's classification of articles as stubs, but this classification is far from reliable, since many articles that are essentially stubs are not labeled as such. Many of the misclassified topics in our dataset suffer from precisely this problem. Developing an automatic method to distinguish well-understood topics from poorly written descriptions remains an open research question.

Tool	Problem	Interestingness
Ternary numeral system	Regular expression	1.21
GNU MPAL	Arithmetical hierarchy	1.19
IEEE 854-1987	Arithmetical hierarchy	1.17
Quantum computer	Regular expression	1.17
Ternary numeral system	Arithmetical hierarchy	1.15
Division by zero	Regular expression	1.15
Turing machine	Regular expression	1.14
GNU MPAL	Regular expression	1.13
Ternary numeral system	Halting problem	1.13
Recursion	Halting_problem	1.13

Table 10.9: Interesting Intradisciplinary Questions

By combining the results of Table 10.7 and Table 10.8, we can generate new ideas for how existing tools might be applied to open problems. As stated above, the goal of this approach is not to provide definitive research problems, but to highlight potentially interesting applications. It is ultimately the responsibility of the researcher to decide whether a given combination of topics is meaningful and worth pursuing. Table 10.9 shows the results of this intradisciplinary combination process.

Most of the interesting questions identified in this way have relatively low quality. As noted earlier, the main difficulty is that it is very hard to distinguish automatically, and in an unsupervised fashion, between a short article that reflects a well-understood topic and a short article that is simply poorly written. In this section we review some of the identified intradisciplinary questions in order to clarify what we mean by *interesting question* and how such questions should be interpreted. A few examples include:

- Interesting Question 7: *Can we apply Turing machines to regular expressions?* The answer is clearly yes, and this is a well-known result: regular expressions define regular languages, which are recognized by finite automata, and finite automata can be simulated by Turing machines.
- Interesting Question 10: *Can we apply recursion to the halting problem?* Again, the answer is yes. The proof of the undecidability of the halting problem is itself based on a machine that calls itself recursively.

Both examples demonstrate that the methodology does not always produce original questions, but sometimes recovers already established results.

The most genuinely interesting questions arise when combining topics

Tool	Problem	Interestingness
State space	Action potential	1.17
Turing machine	Action potential	1.16
Quantum computer	Action potential	1.16
Abstract machine	Action potential	1.14
Computational model	Action potential	1.13
State space	Membrane potential	1.12
State space	Meiosis	1.11
Arithmetic logic unit	Meiosis	1.11
GNU MPAL	Flashbulb memory	1.11
Ternary numeral system	Working memory	1.10

Table 10.10: Interesting Interdisciplinary Questions

from two different disciplines. However, the probability that such automatically identified questions are meaningful is typically lower than in the intradisciplinary case, due to larger semantic gaps between fields.

For the interdisciplinary analysis we used the same collection of pages from the theory of computation as in the intradisciplinary study, combined with a new collection of topics from the area of bioinformatics. The topics were selected using the Wikipedia categories *natural sciences*, *biology*, and *biological processes*. In total, more than  $10^5$  combinations were analyzed. Table 10.10 lists the most relevant interdisciplinary applications.

The interdisciplinary set of questions also suffers from the stub-article problem, and thus the quality of the results remains limited. Nevertheless, some examples are worth discussing:

- Interesting Question 1: *Can we apply state space to action potential?* Questions 1-5 all point toward the same idea: whether the notion of action potential can be formalized in a way that allows it to be simulated or reproduced computationally.
- Interesting Question 7: *Can we apply state space to meiosis?* This is analogous to Question 1, but now concerning the possibility of formalizing and simulating the process of meiosis.

These examples illustrate how interdisciplinary combinations can hint at possible directions for formal modeling, even if many of the suggested questions are too vague or already well established to constitute truly novel research avenues.

## 10.6 New Research Topics

If we combine the list of highly relevant and not very well understood problems with themselves, we might obtain new topics that lie in the *unknown unknown* area.

Table 10.11 presents the top 25 candidates for potential new topics according to their interestingness. In this analysis we included all the topics from all knowledge areas. A striking result is that many of the suggested combinations involve the concept of intellectual property (*copyright*, *open access*, *public domain*, and perhaps *wiki*), which suggests that this is an area where there are still many issues to explore, far more than we might expect. On the other hand, it is possible that the outcome reflects a bias in Wikipedia toward these topics. Further investigation is needed to clarify this point.

To illustrate how new topics are generated, we consider the following examples:

- New topic 17: *Public domain + Earth*. This question raises the issue of whether the Earth should be considered a public resource, touching upon the fundamental concept of private property. The methodology suggests that this is not a well-understood topic.
- New topic 18: *Public domain + Internet*. This raises a similar issue to Question 17, but in the context of Internet governance.

In both cases, however, the combinations fail to provide well-defined, innovative, and genuinely novel research topics.

We can also restrict our search for new topics to a reduced number of knowledge categories. For example, Table 10.12 shows the ten most interesting new topics arising from the already studied area of *theory of computation* and the new area of *phenomenology* (from Level 2 *philosophy of mind*, and Level 1 *cognitive science*). Based on this list, we might consider the following potential new topics:

- New topic 2: *Turing machine + synesthesia*. This could suggest a new type of Turing machine that incorporates synesthetic properties. These *synesthetic Turing machines* might be defined as a group of Turing machines linked together so that when one machine reads a symbol from its tape, it automatically triggers a state change in another machine. This property of synesthesia could also be extended to nondeterministic Turing machines.
- New topic 4: *Kolmogorov complexity + self-awareness*. This could be interpreted as an investigation into the minimum complexity required for a computer program to exhibit self-awareness.

Problem	Problem	Interestingness
Public domain	Open access	1.71
Public domain	REST	1.70
Public domain	Wiki	1.70
Open access	REST	1.70
Copyright	Public domain	1.69
Open access	Wiki	1.69
Public domain	QR code	1.69
Copyright	Open access	1.68
Wiki	REST	1.68
Open access	QR code	1.68
Public domain	Transport Layer Security	1.68
Copyright	REST	1.68
QR code	REST	1.67
Open access	Transport Layer Security	1.67
Copyright	Wiki	1.67
Wiki	QR code	1.67
Public domain	Earth	1.67
Public domain	Internet	1.67
REST	Transport Layer Security	1.66
Copyright	QR code	1.66
Earth	Open access	1.66
Internet	Open access	1.66
Public domain	Open source	1.66
Public domain	Web 2.0	1.66
Wiki	Transport Layer Security	1.66

Table 10.11: New Topics

Question	Question	Interestingness
Kolmogorov complexity	Change blindness	1.24
Turing machine	Synesthesia	1.23
Kolmogorov complexity	Qualia	1.23
Kolmogorov complexity	Self-awareness	1.22
Turing machine	Qualia	1.22
Kolmogorov complexity	Synesthesia	1.21
Turing completeness	Synesthesia	1.20
Turing machine	Self-awareness	1.20
Turing completeness	Qualia	1.20
Turing completeness	Self-awareness	1.18

Table 10.12: Restricted New Topics

## 10.7 The Unknown Unknowns Project

The Unknown Unknowns Project is rooted in the theory of nescience, and in particular in its metrics for quantifying how well we understand a scientific topic. It builds upon the idea that by combining topics that are not yet well understood, we can generate new and previously unknown areas of knowledge. The project leverages the rich collection of scientific descriptions contained in Wikipedia to map the current extent of human understanding and to detect where it begins to fail. Its ultimate goal is to discover what lies hidden in the scientific unknown unknown, the research topics that humanity not only does not understand but is not yet aware it does not understand. In essence, the project aims to bring to the present the research topics of the future.

### Wikipedia Crawling

We treat Wikipedia as a vast, labeled graph in which vertices represent pages and categories, and edges connect categories to their subcategories and contained pages. Starting from a small set of root scientific categories, we perform a breadth-first traversal to identify relevant pages. For each discovered article, we (i) retrieve structured metadata in batches via the MediaWiki API, (ii) apply structural and quality filters to remove out-of-scope entries (e.g., redirects, disambiguation pages, and non-scientific entities identified through Wikidata typing), and (iii) compute per-page metrics, including pageviews (relevance) and compression-based maturity or surfeit derived from article text. Finally, we normalize these metrics and export both a tabular dataset and a hierarchical JSON representation of the category-page structure.

Our pipeline relies on three Wikimedia services: the MediaWiki Action API (English Wikipedia) for category traversal, page metadata, content, and assessments; the Wikidata Query Service (WDQS) to verify whether a page's topic should be excluded by checking instance of (P31) and transitive subclass of (P279\*) relationships; and the Wikimedia REST Pageviews API to aggregate daily user pageviews over a specified time window.

*Step 1.- Category traversal.* We begin with a list of root categories ("Applied sciences," "Formal sciences," "Natural sciences," and "Social sciences") and discover associated scientific pages via a breadth-first search (BFS) up to a user-defined maximum depth (by default 5). In Wikipedia, a category is itself a page that organizes subcategories and pages. Categories reside in namespace 14, while encyclopedic articles (our targets) belong to namespace 0 or mainspace. Using the MediaWiki Action API's `list=categorymembers`, which returns both subcategories and pages directly contained in a category, we iterate level by level. For each category, we call `action=query`, following pagination through the returned `cmcontinue` and `continue` tokens, which are pagination tokens provided by the MediaWiki API and must be included in subsequent requests to retrieve additional results until no further tokens are returned. Since the category graph is a directed acyclic graph (DAG) rather than a strict hierarchy (categories may have multiple parents), we explicitly record edges rather than assuming a unique tree. At each iteration, discovered subcategories are added to the BFS frontier, and all discovered pages are added to the dataset. Visited sets for categories and pages prevent duplicates and cycles. We record the structure as category edges (parent category id, child category id) for the category DAG and page to categories (page id, category id) for direct page-category relationships.

*Step 2.- Batch enrichment of page properties.* For the distinct page ids identified in Step 1, we obtain metadata from the MediaWiki Action API in batches of at most 50 ids per request using `action=query`. The `info` module provides metadata (page ids, titles), redirect status, and content model; `pageprops` includes structured flags such as disambiguation and `set_index_article`, along with the Wikidata identifier (`wikibase_item`); `revisions` retrieves the latest wikitext, byte size, and timestamp; `categories` lists all page categories (for completeness); and `pageassessments` returns community quality ratings (e.g., Stub, Start, C, B, GA, FA) assigned by WikiProjects.

*Step 3.- Filtering.* We retain only pages passing a series of structured checks, avoiding heuristic title filters (e.g., avoiding regular-expression checks on page titles). First, we apply structural filters to ensure valid encyclopedic content: the page must be in mainspace, not a redirect, not a

disambiguation page, not a set-index page, and must use the wikitext content model. Next, we apply quality filters to exclude low-quality or list-style pages: we prioritize the community PageAssessments label and discard Stubs (including List/FL and Start quality classes). Finally, we remove non-scientific topics via Wikidata typing: each page's Wikidata entity (its Q-ID in `pageprops.wikibase_item`) is tested through WDQS to determine whether a path—following "instance of" (P31) and recursively "subclass of" (P279) relationships in Wikidata, tracing upward through the class hierarchy— $P31 / (P279)^*$  leads to any of several high-level deny anchors (e.g., human Q5, book Q571, film Q11424, city Q515, Wikimedia list article Q13406463). Here, P31 means "instance of," P279 means "subclass of," and  $(P279)^*$  represents zero or more subclass steps (the transitive closure). If such a path exists, the page is excluded. This three-part filter removes people, places, creative works, organizations, products, and other non-topic entities, yielding a clean collection of scientific articles.

*Step 4.- Metric collection.* For each remaining page, we compute a consistent set of features. We first measure the article's length from its wikitext after removing low-signal sections—headings such as "See also", "References", "Further reading", and "External links", using a `wikitext` parser. From this cleaned text we record `article_length` in UTF-8 bytes (and optionally raw characters). We compute compression-based measures by compressing the text with `zlib` (level 9) and compute Redundancy as  $1 - (article\_comp\_length/article\_length)$ . For relevance, we sum daily user pageviews over a defined window (e.g., 30 or 60 days) via the REST endpoint `all-access`, using canonical titles resolved during enrichment.

*Step 5.- Derived metrics and interestingness.* From these primary measurements we define Relevance as total pageviews and Maturity from compression. We then normalize the dataset: NormMaturity using min-max scaling, and NormRelevance using min-max scaling on  $\log(x+1)$  to reduce heavy-tail effects. A composite indicators is then computed: Interestingness (Problem) = NormRelevance x NormRedundancy, identifying topics that are popular yet textually redundant (potentially under-explained and research-worthy).

*Step 6.- Outputs.* For reproducibility and visualization, we export canonical tables: `categories` (`cat_id`, `title`), `category_edges` (`parent_cat_id`, `child_cat_id`) capturing the category DAG, `page_categories` (`page_id`, `cat_id`) for direct memberships, and `pages_with_metrics` containing per-page metrics and labels. We also produce a hierarchical JSON projection of the category graph by identifying root categories (those without incoming edges), recursively nesting subcategories, and inserting page nodes beneath each category they belong to. Because categories may have multiple parents,

a category can legitimately appear in multiple locations within this JSON structure, a deliberate duplication preserving the DAG while supporting tree-like visualizations.

To ensure stability, all enrichment requests are batched (< 50 IDs per call). We log pagination tokens, HTTP errors, and retry decisions for auditing. Concurrency is avoided to respect Wikimedia's usage guidelines and to maintain deterministic runs. As pageviews depend on both the chosen window and the execution date, we record the UTC end date of the REST query; since article content, assessments, and backlinks change over time, we also store lastrevid and timestamps. The pipeline gracefully degrades for optional services: if WDQS is unavailable or rate-limited, processing continues with missing labels or deny checks defaulting to "keep," and gaps noted in the logs. The software stack uses Python with `requests`, `pandas`, `mwparserfromhell`, and optionally `pyarrow` for Parquet output.

## The Known and the Unknown

The crawl covered the scientific content of the English Wikipedia, starting from the four principal branches of knowledge—Applied sciences, Formal sciences, Natural sciences, and Social sciences—and expanding through their subcategories to a maximum depth of five levels. In total, 38,074 categories were explored, of which approximately 18,000 did not contain articles of sufficient quality to be included in the analysis. The resulting dataset comprises 18,149 high-quality scientific topics, distributed across the major branches as follows: 10,785 articles in the Applied sciences, 2,776 in the Formal sciences, 9,826 in the Natural sciences, and 11,794 in the Social sciences. Each article was classified according to Wikipedia's internal quality assessment scale, which ranks entries based on their completeness, accuracy, and sourcing. *C-class* articles are moderately developed, providing substantial information but often lacking depth or comprehensive references. *B-class* articles are more complete, with reasonably good structure, accuracy, and coverage of the topic, though they may still require some polishing or additional citations. *A-class* articles are near-publication quality, representing a high standard of completeness and reliability, just below the *Good Article* and *Featured Article* levels. Within this classification, the crawl collected 11,894 C-class, 4,928 B-class, 882 Good Articles (GA), 429 Featured Articles (FA), and 16 A-class entries. Given that Wikipedia functions as the largest collaborative encyclopedia ever created, this corpus constitutes one of the most comprehensive and accurate descriptive representations of current human knowledge, offering a unique empirical foundation for analyzing the state of science and its frontiers.

In Figure 10.2 it is shown an histogram of the maturity of the known

topics, computed as the ratio of the size of the text of the Wikipedia article and its compressed version. The histogram reveals that the maturity of Wikipedia's scientific topics follows an approximately normal distribution centered around 0.4, indicating that most articles exhibit a balanced level of informational density. This suggests that, on average, about 40% of each article's content represents essential, non-redundant information, while the rest reflects natural linguistic and structural redundancy typical of encyclopedic writing.

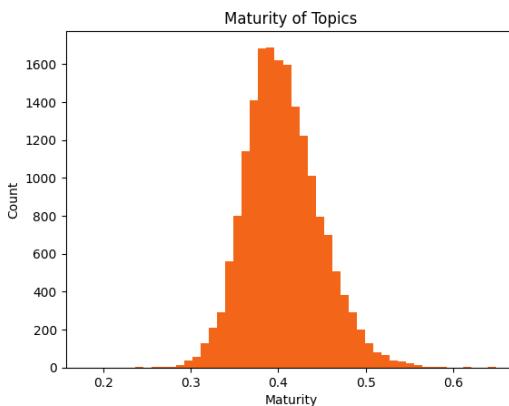


Figure 10.2: Maturity of scientific topics.

In Figure 10.3 it is shown an histogram of the logarithm of the relevance of a topic, as the total number of visits to the Wikipedia page over the last two months. The histogram shows that the logarithm of topic relevance, measured by the total number of visits to each Wikipedia page over the last two months, follows a roughly log-normal distribution centered around a log relevance of about 8–9. This implies that most scientific topics receive between  $10^8$  and  $10^9$  visits, while only a few attract extremely high or low levels of attention. The right-skewed shape indicates the presence of a small number of highly popular topics that dominate user interest, reflecting the uneven distribution of public attention across scientific fields. Overall, the data suggest that while most topics maintain a moderate and steady level of visibility, a minority of them capture a disproportionately large share of the audience, highlighting the concentration of public interest in a limited set of scientific areas.

Computation of the new research topics, as the creative combination of already existing topics, has been performed automatically using a large language model. The OpenAI API provides programmatic access to a family of advanced large language models developed by OpenAI, allowing developers

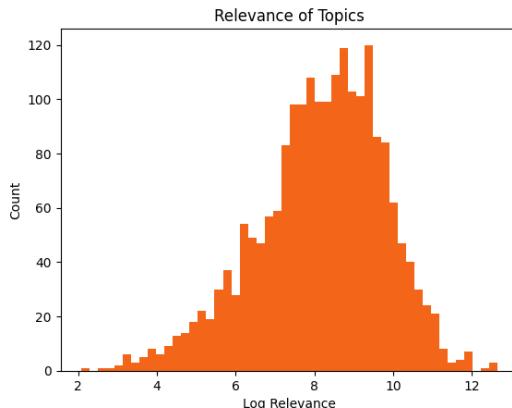


Figure 10.3: Log-relevance of scientific topics.

and researchers to integrate natural language understanding, reasoning, and generation capabilities into their own applications. Among these models, *GPT-5-nano* represents a lightweight yet remarkably capable variant of the GPT-5 architecture—optimized for efficiency, speed, and cost while retaining much of the creativity and fluency of larger models. Despite its compact size, GPT-5-nano demonstrates an impressive ability to generate coherent, context-aware, and often original ideas across diverse domains. Its design enables it not only to respond accurately to prompts but also to combine existing concepts in novel ways, making it a powerful tool for creative exploration, brainstorming, and the discovery of new research directions or innovative solutions. The following is the prompt used to generate a new research topic:

"You are an innovative scientist exploring unexplored frontiers of knowledge. Your task is to conceive a new and original research topic that emerges from the creative synthesis or inspiration of the two given scientific concepts. The proposed topic must:

- Represent a novel idea that, to your knowledge, has never been formally studied or proposed before.
- Have the potential, if successfully developed and applied, to produce a major positive impact on humanity by advancing science, technology, or understanding.
- Be scientifically plausible—grounded in real principles, not fantasy or pseudoscience.
- Be expressed in a single, concise paragraph written in formal academic language, rich in scientific and technical terminology, and free of filler or speculation markers (avoid “perhaps”, “maybe”, etc.).

- Avoid generic formulations like 'a study of the relationship between...,' and instead describe a specific, well-defined new direction.
- Respond directly with the proposed research topic, nothing else."

The exploration of the unknown unknown encompassed an unprecedented combinatorial space of potential scientific discoveries. In total, 164 million new research topics were generated and evaluated, each representing a possible novel research area. Within single disciplines, or intradisciplinary combinations, the analysis produced approximately 12 million new topics in the Applied sciences, 8 million in the Formal sciences, 96 million in the Natural sciences, and 139 million in the Social sciences. When considering interdisciplinary combinations—those that bridge distinct domains—the numbers were even higher, with 132 million in the Applied sciences, 45 million in the Formal sciences, 125 million in the Natural sciences, and 138 million in the Social sciences. Although these figures are immense, they reflect not ambition for its own sake, but the scale of the challenge involved in systematically exploring what remains beyond our current understanding. This constitutes, to date, one of the most extensive and systematic attempts to identify and describe the unknown unknown in science—an experiment that seeks not to claim discovery, but to illuminate the vast and still largely uncharted territory of human ignorance.

### **Unknown Unknowns App**

The Unknown Unknowns application (see Figure 10.4) serves as an exploratory instrument for discovering what it is hidden in the scientific unknown. It transforms our current vast scientific corpus into an interactive landscape where both what is known and what remains undiscovered can be visually explored. At its core lies the idea that new knowledge emerges not in isolation, but from the interplay of existing concepts. By combining less mature, not yet well understood topics with automatically generated hypotheses about potential new ones, the application becomes a map of the scientific frontier, a cartographic representation of the boundary between understanding and ignorance.

When launched, the program constructs an internal graph that mirrors the structure of scientific knowledge. Categories and subcategories form a tree that organizes the disciplines, while individual articles represent specific research topics. Each page is associated with several indicators: maturity, which measures how developed or well-established a topic is; relevance, which reflects its significance and social impact; and interestingness, which captures its potential to raise new questions. A complementary dataset describes the "unknown topics"—novel research ideas automatically inferred from pairs of existing ones. These unknown topics, described textually and

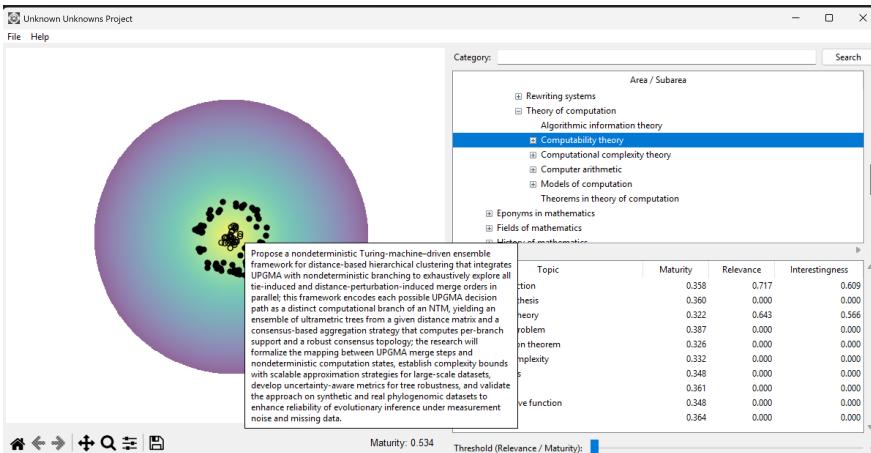


Figure 10.4: Unknown Unknowns App.

assigned a preliminary maturity level, represent possible directions in which human knowledge could expand.

The interface is divided into two complementary regions. On the left, a circular diagram depicts the maturity of topics as a constellation of points over a concentric gradient. The center of the diagram glows brightly, symbolizing the domains of science that are most mature and consolidated. The darker outer regions represent areas of uncertainty and opportunity (the unknown unknowns), where new ideas are still forming. Known topics appear as filled black dots, positioned according to their maturity: the more developed the topic, the closer it lies to the center. Unknown topics appear as empty circles, marking the places where new combinations of knowledge could give rise to discoveries. Their distribution reveals the topology of ignorance, an intricate periphery of potential that surrounds and extends the boundaries of current science. Hovering over any unknown displays the research opportunity in a multi-line tooltip, while the lower status bar shows its numerical maturity.

The right side of the window offers the tools to navigate this landscape. A search bar allows the user to locate specific scientific categories by name, while the hierarchical tree below displays the organization of disciplines and subdisciplines. Expanding a branch of the tree simply reveals its subareas, but selecting a category triggers a full update of the diagram and the topics table. In this mode, the application displays all the known topics that belong to the chosen category and its descendants, together with the unknown topics that emerge from them. The unknown topics are filtered so that only those whose maturity exceeds a minimum threshold are shown, ensuring that the visualization remains focused on meaningful possibilities rather than random

combinations.

Beneath the tree, the topics table lists all the pages that directly belong to the selected category. Each entry includes its maturity, relevance, and interestingness. Clicking on a row switches the visualization into topic mode, in which the diagram focuses on that particular topic and displays all the unknown topics that are conceptually derived from it. This view allows the user to see how a not very well-understood idea can serve as a seed for new research directions, how from a single concept, many potential discoveries can emerge. Double-clicking a topic opens its Wikipedia page in a browser, connecting the abstract representation back to the underlying content.

The scale at the bottom of the interface controls the threshold of relevance and maturity used to filter the visualization. It allows the user to move smoothly from a wide, inclusive view of the scientific landscape to a more focused view that highlights only those topics that are socially impactful. By raising the threshold, the user filters out marginal or less relevant areas, concentrating instead on the regions of knowledge that have the greatest influence on human life and technological progress. In this way, the application not only reveals the unknown unknowns but also helps prioritize which of them might lead to discoveries of real significance.

Every design element in the application serves the same epistemological goal: to make the frontier of human knowledge visible. The bright center of the diagram stands for what science already understands, while the dark periphery shows where ignorance still dominates. The empty circles of the unknown topics are markers of potential discovery, each one a hypothetical domain of future research, a conceptual territory awaiting exploration. The relevance filter allows the user to discern which of these emerging ideas connect most strongly to society's needs and interests. The result is not merely a visualization of data, but a dynamic, exploratory tool that guides the intellect toward the edges of what is currently thinkable.

In this sense, the application is both cartographic and philosophical. It functions as a visual compass for discovery, helping the observer navigate between the solid continents of established knowledge and the open ocean of possibility. By grounding the abstract notion of the "unknown unknown" in concrete, measurable data, it transforms ignorance into something that can be mapped, explored, and eventually reduced. In doing so, it exemplifies the central idea of the theory of nescience: that by quantifying what we do not yet know, we can systematically move beyond the limits of present understanding.

## References

Core references for the chapter "Interesting Research Questions"

[Pop14]: Popper's account of conjectures, refutations, and the growth of knowledge provides the philosophical foundation for thinking about unanswered questions and how science progresses through them.

[Cha13]: Chalmers offers a clear overview of debates about what constitutes science. This is essential background for addressing the demarcation problem in identifying valid research questions.

[Li+04]: Introduces the normalized compression distance (NCD), which provides a universal method for comparing objects. This is a key methodological tool for estimating redundancy and discovering connections between topics.

[CAO+05]: Warns about practical limitations of using real compressors to approximate Kolmogorov complexity. Essential for a sound application of compression-based metrics in identifying research frontiers.

[LV13]: The standard reference on Kolmogorov complexity. It connects theoretical information content with practical methods like NCD, underpinning the discovery of redundancy, miscoding, and research gaps.

[Sup02]: Suppes discusses how scientific theories can be formally represented. This supports the extension from individual topics to structured research areas and the search for interdisciplinary questions.

[LSW13]: Latour highlights how scientific knowledge emerges from negotiation and practice. This adds a sociological perspective to the identification of research questions and debates.

[Ioa05]: Ioannidis shows that much of published research is flawed or irreproducible. This underscores the importance of using independent measures (like nescience) to identify which questions are still genuinely open.



# Part 3: Prerequisites

<b>A</b>	<b>Discrete Mathematics</b>	283
<b>B</b>	<b>Discrete Probability</b>	301
<b>C</b>	<b>Computability</b>	353
<b>D</b>	<b>Coding</b>	371
<b>E</b>	<b>Complexity</b>	395
<b>F</b>	<b>Learning</b>	415
<b>G</b>	<b>Philosophy of Science</b>	451



## A. Discrete Mathematics

*Mathematics may be defined as the subject in which  
we never know what we are talking about,  
nor whether what we are saying is true.*

Bertrand Russell

The majority of mathematical concepts used throughout this book belong to the domain of *discrete mathematics*. This field focuses on mathematical objects that take on distinct, separate values, as opposed to continuous ones. In this book, we make use of various discrete structures, including integers, strings, graphs, and computer programs. A key feature of discrete sets is their countability, meaning they can be put into one-to-one correspondence with the natural numbers. By contrast, continuous mathematics, such as calculus, will play only a minimal role in the theoretical development of the theory of nescience.

Our primary interest in discrete mathematics arises from its direct relevance to computation. The theory of nescience draws upon various aspects of computer science, including algorithms, coding, and string complexity. Computers operate in discrete steps and manipulate information stored in discrete memory units. Our interest in computers stems from the aspiration to apply our theoretical framework to a broad range of real-world entities.

We consider computers to be the most suitable tools for modeling the world around us. While pure mathematics often engages with abstract objects independent of their representations, the theory of nescience places significant emphasis on the representation (or encoding) of objects.

This chapter provides a brief overview of the fundamental concepts of discrete mathematics, introducing topics such as sets, strings and languages, counting methods, matrices, and graphs. While we do not present formal definitions or prove theorems in this overview, these subjects lay the groundwork for the theories and ideas explored in later sections. Discrete mathematics is a broad and diverse field; here, we focus only on those elements essential for understanding the theory of nescience. More advanced topics (such as computability, information theory, and complexity) require a deeper treatment and are addressed in dedicated chapters.

The References section provides a list of recommended books that explore the topics introduced in this chapter in greater depth. Readers interested in further developing their understanding can use this list to delve more deeply into each subject, thereby complementing the foundational overview presented here.

## A.1 Sets, Relations and Functions

The sets of *natural*, *integer*, *rational*, and *real* numbers are denoted by  $\mathbb{N}$ ,  $\mathbb{Z}$ ,  $\mathbb{Q}$ , and  $\mathbb{R}$ , respectively. Each of these sets includes the number 0. The *positive integers* are represented by  $\mathbb{Z}^+$ , and the *positive reals* by  $\mathbb{R}^+$ ; both sets also include 0. Let  $A$  be a *set*. We indicate that  $x$  is an *element* of  $A$  using the notation  $x \in A$ , and that  $x$  is not an element of  $A$  with  $x \notin A$ . Elements of a set can be listed explicitly using braces, as in  $A = \{0, 1, 2, 3\}$ , or defined by a condition using *set-builder* notation, for example,  $A = \{x \in \mathbb{N} : x < 4\}$ , provided that the *universe* of discourse is clearly specified.

Suppose  $A$  and  $B$  are two sets. We use the notation  $A = B$  to indicate that the sets are *equal*. The expression  $A \subseteq B$  signifies that  $A$  is a *subset or equal* to  $B$ , while  $A \subset B$  denotes that  $A$  is a *proper subset* of  $B$  (meaning that  $A$  is contained in  $B$  but is not equal to  $B$ ). The condition  $A = B$  holds if and only if both  $A \subseteq B$  and  $B \subseteq A$  are true. The symbol  $\emptyset$  denotes the *empty set*, which is the set containing no elements.

■ **Example A.1** For every set  $A$ , we have that  $\emptyset \subseteq A$  and  $A \subseteq A$ . ■

The term *cardinality* refers to the number of elements in a finite set  $A$ , denoted by  $d(A)$ . Accordingly, the cardinality of the empty set  $\emptyset$  is 0, since it contains no elements. For any two sets  $A$  and  $B$ , the notation  $A \cup B$  denotes the *union* of  $A$  and  $B$ , while  $A \cap B$  represents their *intersection*. For a collection

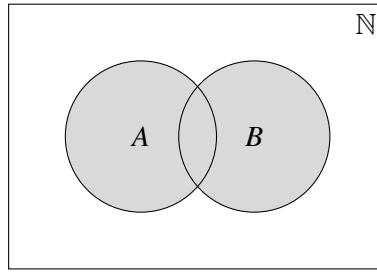


Figure A.1: Representation of  $A \cup B$  as a Venn Diagram

of  $n$  sets  $A_1, A_2, \dots, A_n$ , we write their union and intersection as  $\cup_{i=1}^n A_i$  and  $\cap_{i=1}^n A_i$ , respectively. For an arbitrary collection of sets indexed by a set  $I$ , we use the notations  $\cup_{i \in I} A_i$  and  $\cap_{i \in I} A_i$ . When dealing with an infinite collection of sets, we may use the shorthand  $\cup_{i=1}^{\infty} A_i$  and  $\cap_{i=1}^{\infty} A_i$ . Occasionally, we will make use of *Venn diagrams* to visually represent set operations, as illustrated in Figure A.1.

Given any two sets  $A$  and  $B$ , the *set difference* is denoted by  $A \setminus B$ , and the *complement* of the set  $A$  is written as  $A^c$ . *De Morgan's laws* state that for any sets  $A$  and  $B$ , we have  $(A \cup B)^c = A^c \cap B^c$  and  $(A \cap B)^c = A^c \cup B^c$ .

Two sets  $A$  and  $B$  are said to be *disjoint* if their intersection is empty, i.e.,  $A \cap B = \emptyset$ . A collection of sets  $A_1, A_2, \dots, A_n$  is disjoint if  $A_i \cap A_j = \emptyset$  for all  $i \neq j$ . A *partition* of a set  $A$  is a collection of nonempty, pairwise disjoint subsets  $A_1, A_2, \dots, A_n$  such that  $A = \cup_{i=1}^n A_i$ . The *power set* of  $A$ , denoted by  $\mathcal{P}(A)$ , is the set of all possible subsets of  $A$ . If the cardinality of  $A$  is  $n$ , i.e.,  $d(A) = n$ , then the cardinality of its power set is  $2^n$ , so  $d(\mathcal{P}(A)) = 2^n$ .

■ **Example A.2** Given the set  $A = \{1, 2, 3\}$ , its power set is:

$$\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, A\}$$

■

Consider a non-empty set  $A$  and a collection  $\mathcal{F}$  of subsets of  $A$ . The pair  $(A, \mathcal{F})$  is called a *field* over  $A$  if it satisfies the following conditions: it contains the empty set, that is,  $\emptyset \in \mathcal{F}$ ; it is closed under complementation, meaning that for every  $F \in \mathcal{F}$ , the complement  $F^c$  also belongs to  $\mathcal{F}$ ; and it is closed under finite unions, meaning that for any subsets  $F_1, \dots, F_n \in \mathcal{F}$ , the union  $F_1 \cup \dots \cup F_n$  is also in  $\mathcal{F}$ . Additionally, it can be shown that a field also satisfies two further properties: the universal set  $A$  itself belongs to  $\mathcal{F}$ , and it is closed under finite intersections, so that  $F_1 \cap \dots \cap F_n \in \mathcal{F}$  for all subsets  $F_1, \dots, F_n \in \mathcal{F}$ .

Consider two elements,  $x$  and  $y$ . An *ordered pair*, denoted as  $(x, y)$ , is a pairing in which the order of the elements matters. Generalizing this

idea, an *n-tuple* is an ordered sequence of  $n$  elements, written as  $(x_1, \dots, x_n)$ . The *Cartesian product* of two sets  $A$  and  $B$ , denoted by  $A \times B$ , is the set of all ordered pairs  $(x, y)$  such that  $x \in A$  and  $y \in B$ . This concept extends naturally to  $n$  sets  $A_1, A_2, \dots, A_n$ , where the Cartesian product is expressed as  $A_1 \times A_2 \times \dots \times A_n$ . Additionally, the *n-fold Cartesian product* of a set  $A$  with itself is denoted as  $A^n$ .

Let  $R$  be a subset of the Cartesian product of a set  $A$  with itself, i.e.,  $R \subseteq A \times A$ . Such a subset is called a *binary relation*. We write  $aRb$  to indicate that the ordered pair  $(a, b)$  belongs to  $R$ . A binary relation is said to be *reflexive* if, for every element  $a \in A$ , it holds that  $aRa$ . It is called *symmetric* if, for all  $a, b \in A$ , the condition  $aRb$  implies  $bRa$ . A relation is *antisymmetric* if, for all  $a, b \in A$ , the coexistence of  $aRb$  and  $bRa$  implies  $a = b$ . It is *transitive* if, for all  $a, b, c \in A$ , the conditions  $aRb$  and  $bRc$  together imply  $aRc$ . A relation is *total* if, for every pair  $a, b \in A$ , either  $aRb$  or  $bRa$  holds. Binary relations can also be defined between two different sets  $A$  and  $B$ , in which case  $R$  is a subset of  $A \times B$ . Furthermore, the concept can be generalized to *n-ary* relations, represented as  $R \subseteq A_1 \times A_2 \times \dots \times A_n$ .

Let  $R$  be a binary relation that is a subset of the Cartesian product  $A \times A$ , i.e.,  $R \subseteq A \times A$ . If this relation is reflexive, symmetric, and transitive, it is called an *equivalence relation*, typically denoted by the symbol  $\sim$ . Under an equivalence relation, two elements  $a, b \in A$  are said to be *equivalent* if  $a \sim b$ . The *equivalence class* of an element  $a$ , denoted by  $[a]$ , is the set of all elements in  $A$  that are equivalent to  $a$ . That is, the equivalence class of  $a$  is defined as  $[a] := \{b \in A : a \sim b\}$ . An equivalence relation partitions the set  $A$  into disjoint subsets known as *equivalence classes*, collectively forming what is called the *quotient set*. The quotient set is denoted by  $A/\sim$  and is defined as the set of all equivalence classes:  $A/\sim := [a] : a \in A$ .

A binary relation that is reflexive, transitive, and antisymmetric is known as a *partial order*, typically denoted by the symbol  $\preceq$ . A set equipped with a partial order is called a *partially ordered set*, or *poset* for short. In a poset, an element  $a \in A$  is considered *minimal* if there is no element  $b \in A$  such that  $b \preceq a$  and  $b \neq a$ . Similarly, an element  $a$  is said to be *maximal* if there is no element  $b \in A$  such that  $a \preceq b$  and  $b \neq a$ . A relation that is reflexive, transitive, antisymmetric, and total is called a *total order*, often denoted by the symbol  $\leq$ . A set endowed with a total order is referred to as a *totally ordered set*. In such a set  $A$ , the *maximum element*, denoted by  $\max(A)$ , satisfies  $\max(A) \geq x$  for all  $x \in A$ , and the *minimum element*, denoted by  $\min(A)$ , satisfies  $\min(A) \leq x$  for all  $x \in A$ .

■ **Example A.3** Let  $R$  be a relation that is a subset of the Cartesian product of the set of natural numbers  $\mathbb{N}$  with itself, i.e.,  $R \subset \mathbb{N} \times \mathbb{N}$ . In this relation, an ordered pair  $(a, b)$  belongs to  $R$  if and only if  $a$  is a divisor of  $b$ . The set

$\mathbb{N}$ , together with the relation  $R$ , forms a partially ordered set. In this context, 1 is the unique minimal element, and every prime (e.g. 11) is maximal. ■

A *function* is defined as a binary relation  $f \subseteq A \times B$  such that for each element  $x \in A$ , there exists at most one  $y \in B$  for which  $(x, y) \in f$ . In this context, elements  $(x, y) \in f$  are written as  $f(x) = y$ , and the function is denoted by  $f : A \rightarrow B$ . The set  $A$  is called the *domain* of  $f$ , and  $B$  is the *codomain*. The set  $y \in B : \exists x \in A$  such that  $f(x) = y$  is known as the *range* of  $f$ . If the relation is not defined for every  $x \in A$ , the function is called a *partial function*, and we write  $f(x) \uparrow$  to indicate that  $f$  is undefined at  $x$ .

A function is said to be *injective* if, for all elements  $x$  and  $y$ , the condition  $f(x) = f(y)$  implies that  $x = y$ . A function is *surjective* if, for every  $y$  in the codomain, there exists at least one  $x$  in the domain such that  $f(x) = y$ . A function is described as *bijective* if it is both injective and surjective. The *identity* function  $I_A : A \rightarrow A$ , defined by  $I_A(a) = a$  for all  $a \in A$ , is an example of a bijective function. These concepts (function, partial function, injective, surjective, and bijective) can be extended to  $n$ -ary functions, which are functions of the form  $f : A_1 \times A_2 \times \cdots \times A_n \rightarrow B$ .

The *inverse* of a bijective function  $f$ , denoted by  $f^{-1}$ , is defined as  $f(f^{-1}(x)) = f^{-1}(f(x)) = x$ , for all  $x$  in the domain of  $f^{-1}$  and the range of  $f$ , respectively. Given two functions  $f$  and  $g$ , where the domain of  $f$  includes the range of  $g$ , the *composition* of  $f$  with  $g$ , denoted by  $f \circ g$ , is defined as  $(f \circ g)(x) = f(g(x))$ .

■ **Example A.4** In Section C.4, we will explore an alternative interpretation of a function as a procedure or algorithm that assigns an element of  $B$  to each element of  $A$ . For example, the following C code defines a partial function from  $\mathbb{R}$  to  $\mathbb{R}$ , which is partial because  $inv(0) \uparrow$ :

```
double inv(double x) {
    return 1 / x;
}
```

The *characteristic function* of a set  $A$  is denoted by  $1_A : A \rightarrow \{1, 0\}$ , where  $1_A(x) = 1$  if  $x \in A$  and  $1_A(x) = 0$  otherwise.

An infinite set  $A$  is said to be *countable* if there exists a bijective function that maps the elements of  $A$  onto the set of natural numbers  $\mathbb{N}$ . In contrast, a set is considered *uncountable* if it is neither finite nor countable. A set is said to have *countably many* elements if it is either finite or countably infinite.

■ **Example A.5** The sets  $\mathbb{N}$ ,  $\mathbb{Z}$ , and  $\mathbb{Q}$  are countable, whereas  $\mathbb{R}$  is uncountable. ■

Considering a real number  $x \in \mathbb{R}$ , its *absolute value*, denoted by  $|x|$ , is defined as  $x$  if  $x \geq 0$  and  $-x$  if  $x < 0$ . The *ceiling* function of  $x$ , written as  $\lceil x \rceil$ , is the smallest integer greater than or equal to  $x$ . The *floor* function of  $x$ , denoted by  $\lfloor x \rfloor$ , is the largest integer less than or equal to  $x$ . Given two positive integers  $a$  and  $b$ , the *modulo* operation, written as  $a \bmod b$ , yields the remainder when  $a$  is divided by  $b$ .

For two functions  $f$  and  $g$  defined as  $f, g : \mathbb{N} \rightarrow \mathbb{R}^+$ , we say that  $f(n)$  is of the *order of*  $g(n)$ , denoted  $f(n) = O(g(n))$ , if there exist positive constants  $c > 0$  and  $m$  such that  $f(n) \leq cg(n)$  for all integers  $n \geq m$ . In this context,  $g$  is called an *upper bound* for  $f$ .

## A.2 Strings and Languages

Consider a non-empty finite set  $\mathcal{S} = \{s_1, s_2, \dots, s_q\}$ , referred to as the *alphabet*. The elements of this set are called *symbols*. A *sequence* over  $\mathcal{S}$  is defined as an ordered arrangement of symbols  $x_1 x_2 \dots x_n$ , where each  $x_i$  belongs to  $\mathcal{S}$ . In the special case where the alphabet is  $\mathcal{B} = \{0, 1\}$ , such sequences are known as *binary sequences*. We use the term *string* to denote a finite sequence. This book primarily focuses on binary strings.

The *length* of a string  $s$ , denoted by  $l(s)$ , refers to the total number of symbols contained in  $s$ . The symbol  $\lambda$  is used to denote the *empty string*, which is defined as the unique string over  $\mathcal{S}$  with length 0. Given a symbol  $x \in \mathcal{S}$ , the string consisting of  $x$  repeated  $n$  times is denoted by  $x^n$ . If  $s = x_1 x_2 \dots x_n$  is a string, its *reverse*, denoted by  $s^R$ , is given by  $x_n x_{n-1} \dots x_1$ .

The set of all strings  $s_1 s_2 \dots s_n$  of length  $n$  over the alphabet  $\mathcal{S}$  is denoted by  $\mathcal{S}^n$ <sup>1</sup>. We denote by  $\mathcal{S}^+$  the union of all  $\mathcal{S}^n$  for  $n \geq 1$ , and by  $\mathcal{S}^*$  the set  $\mathcal{S}^+ \cup \lambda$ . Note that all strings in  $\mathcal{S}^*$  have finite length. The term *Kleene closure* refers to  $\mathcal{S}^*$ .

■ **Example A.6** The following relations hold: the cardinality of the set of binary strings of length  $n$  is  $d(\{s \in \mathcal{B}^* : l(s) = n\}) = 2^n$ , and the cardinality of the set of binary strings of length up to  $n$  is  $d(\{s \in \mathcal{B}^* : l(s) \leq n\}) = 2^{n+1} - 1$ . ■

Given two strings  $s$  and  $t$  from  $\mathcal{S}^*$ , the *concatenation* of  $s$  and  $t$ , denoted as  $st$ , is the sequence obtained by placing the sequence of symbols in  $t$  immediately after the sequence in  $s$ . Consequently, the length of the concatenated string,  $l(st)$ , is the sum of the lengths of  $s$  and  $t$ . This indicates that  $\mathcal{S}^*$  is closed under the operation of concatenation. Moreover, the set  $\mathcal{S}^*$ , together

<sup>1</sup>It is important to avoid confusing the set of strings of length  $n$  over an alphabet,  $\mathcal{S}^n$ , with the  $n$ -fold Cartesian product of a set,  $S^n$ . The use of calligraphic fonts helps distinguish between alphabets and other sets.

with concatenation, forms a *free monoid*. This means that concatenation is associative ( $s(tr) = (st)r$ ), and that there exists an identity element, specifically, the empty string  $\lambda$ , for which  $\lambda a = a\lambda = a$  holds for any string  $a$ .

A string  $s$  is called a *substring* of a string  $t$  if there exist strings  $u$  and  $v$  (possibly empty) such that  $t = usv$ . If there exists a string  $u$  such that  $t = su$ , then  $s$  is said to be a *prefix* of  $t$ , denoted by  $s <_p t$ . A subset  $S \subset \mathcal{S}^*$  is described as *prefix-free* if, for any  $s, t \in S$ , the condition  $s <_p t$  implies  $s = t$ . Given two sets of strings  $S, T \subset \mathcal{S}^*$ , the (left) *quotient*  $S^{-1}T$  is defined as the set of residual strings obtained from  $T$  by removing a prefix in  $S$ ; formally,  $S^{-1}T = \{t \mid st \in T \wedge s \in S\}$ .

We denote the *self-delimited* form of a string  $s \in \mathcal{S}^*$  by  $\bar{s}$ , and define it as  $\bar{s} = 1^{l(s)}0s$ . Consequently, the length of  $\bar{s}$ , denoted  $l(\bar{s})$ , is given by  $l(\bar{s}) = 2l(s) + 1$ , meaning it is twice the length of  $s$  plus one.

■ **Example A.7** The set  $\bar{\mathcal{S}}^*$ , consisting of all self-delimited strings from  $\mathcal{S}^*$ , is prefix-free. ■

In cases where  $\mathcal{S}$  is a totally ordered set, we can define a total order on  $\mathcal{S}^*$ . This ordering, known as *shortlex ordering*, arranges sequences primarily by length, with shorter sequences appearing first. Among sequences of the same length, lexicographical order is used to break ties.

■ **Example A.8** Given  $\mathcal{S} = a, b, c$  with  $a < b < c$ , the shortlex order on  $\mathcal{S}^*$  produces the sequence  $\lambda < a < b < c < aa < ab < \dots < cc < aaa < aab < \dots < ccc < \dots$ . ■

For any arbitrary object  $O$ , we use the notation  $\langle O \rangle$  to denote its string based representation, assuming the existence of a standard encoding scheme. For objects  $O_1, O_2, \dots, O_k$ , the expression  $\langle O_1 O_2 \dots O_k \rangle$  refers to the plain concatenation of their string representations:  $\langle O_1 \rangle \langle O_2 \rangle \dots \langle O_k \rangle$ . By contrast, the notation  $\langle O_1, O_2, \dots, O_k \rangle$  indicates a structured concatenation that allows for the decoding and unique identification of each individual object. For example, this may be implemented as  $\langle \bar{O}_1 \rangle \langle \bar{O}_2 \rangle \dots \langle \bar{O}_k \rangle$ .

■ **Example A.9** Natural numbers can be represented by binary strings via the following encoding method:  $\langle 0 \rangle = \lambda$ ,  $\langle 1 \rangle \rightarrow 0$ ,  $\langle 2 \rangle \rightarrow 1$ ,  $\langle 3 \rangle \rightarrow 00$ ,  $\langle 4 \rangle \rightarrow 01$ ,  $\langle 5 \rangle \rightarrow 10$ ,  $\langle 6 \rangle \rightarrow 11$ ,  $\langle 7 \rangle \rightarrow 000$ , and so on. Therefore, the pair of numbers  $\langle 3, 7 \rangle$  would be represented as  $110001110000$ . Given this particular encoding, it follows that  $l(\langle n \rangle) = \lfloor \log_2(n+1) \rfloor$ . ■

A *language*, denoted by  $L$ , over an alphabet  $\mathcal{S}$ , is defined as a subset of strings, that is,  $L \subseteq \mathcal{S}^*$ . The individual elements of  $L$  are called *words*. The unique language that contains no words is referred to as the *empty language*, and is denoted by  $L = \emptyset$ .

Consider two languages  $L_1$  and  $L_2$  over a common alphabet  $\mathcal{S}$ . Several

standard operations can be applied to these languages. The *union* of  $L_1$  and  $L_2$  is defined as  $L_1 \cup L_2 = w \in \mathcal{S}^* \mid w \in L_1 \text{ or } w \in L_2$ . The *intersection* of  $L_1$  and  $L_2$  is given by  $L_1 \cap L_2 = w \in \mathcal{S}^* \mid w \in L_1 \text{ and } w \in L_2$ . The *complement* of  $L_1$  is defined as  $\overline{L_1} = w \in \mathcal{S}^* \mid w \notin L_1$ . Finally, the *Kleene closure* of  $L_1$ , denoted  $L_1^*$ , is defined as  $L_1^* = \lambda \cup wz \mid w \in L_1 \text{ and } z \in L_1^*$ .

Languages can be systematically generated using a finite set of string rewriting rules, commonly referred to as grammars. A *grammar*, denoted by  $G$ , is defined as a 4-tuple  $(N, \Sigma, P, S)$ , where:  $N \subseteq \mathcal{S}$  is a finite set of *nonterminal symbols*;  $\Sigma \subseteq \mathcal{S}$  is a finite set of *terminal symbols*;  $P$  is a finite set of *production rules* of the form  $(\Sigma \cup N)^* N (\Sigma \cup N)^* \rightarrow (\Sigma \cup N)^*$ ; and  $S \in N$  is a distinguished *start symbol*. Each production rule allows one string of symbols to be rewritten into another, beginning with the start symbol and proceeding through successive applications of the rules.

■ **Example A.10** Consider the alphabet  $\mathcal{S} = \{S, a, b\}$ , and define the grammar  $(N, \Sigma, P, S)$  where  $N = \{S\}$ ,  $\Sigma = \{a, b\}$ ,  $P = \{S \rightarrow aSb, S \rightarrow ba\}$ , and the start symbol is  $S \in N$ . This grammar generates the language  $L = \{a^n bab^n \mid n \geq 0\} = \{ba, abab, aababb, aaababbb, \dots\}$ . ■

The *Chomsky hierarchy* is a classification scheme for grammars, organized according to their expressive power, that is, the types of languages they are capable of generating. The hierarchy, arranged from the most to the least restrictive class of grammars, is described as follows (where  $a$  denotes a terminal symbol;  $A, B$  are nonterminal symbols; and  $\alpha, \beta$ , and  $\gamma$  are strings composed of terminals and/or nonterminals):

**Type-3** Known as *regular grammars*. In these grammars, the left-hand side of each production rule consists of a single nonterminal symbol.

The right-hand side must be either the empty string, a single terminal symbol, or a terminal symbol followed by a nonterminal symbol. Formally, the rules are of the form  $A \rightarrow \lambda$ ,  $A \rightarrow a$ , or  $A \rightarrow aB$ .

**Type-2** Referred to as *context-free grammars*. In this class, the left-hand side of each production rule is exactly one nonterminal symbol. The general form of the rules is  $A \rightarrow \alpha$ .

**Type-1** These are *context-sensitive grammars*. Here, production rules allow a nonterminal to be replaced by a string in a specific context. The rules take the form  $\alpha A \beta \rightarrow \alpha \gamma \beta$ .

**Type-0** This category includes *recursively enumerable grammars*, which place no restrictions on the structure of production rules. These can be written in the general form  $\gamma \rightarrow \alpha$ .

■ **Example A.11** The grammar presented in Example A.10 is a Type-2 grammar, that is, a context-free grammar. ■

The *Backus-Naur form* (BNF) is a notation system specifically designed to describe context-free grammars. It is widely used in computer science to formally specify the syntax of programming languages and communication protocols. A BNF grammar consists of a set of production rules structured as follows:

```
<symbol> ::= __expression__
```

Here,  $\langle \text{symbol} \rangle$  denotes a non-terminal symbol,  $\_\_\text{expression}\_\_$  represents a sequence of terminal and/or non-terminal symbols, and  $::=$  signifies that the symbol on the left-hand side can be replaced by the expression on the right. Multiple alternatives for the expression can be provided in a single rule by separating them with a vertical bar  $|$ , indicating that any one of the alternatives may be chosen during substitution. Symbols that never appear on the left-hand side of any production rule are considered terminal symbols. In contrast, those that do appear on the left-hand side are non-terminal symbols and are conventionally enclosed in angle brackets  $\langle \rangle$ . The non-terminal symbol on the left-hand side of the first production rule is designated as the start symbol.

■ **Example A.12** The grammar introduced in Example A.10 can be expressed in Backus-Naur form using the following production rule:

```
<string> ::= a <string> b | ba
```

■

## A.3 Counting Methods

*Combinatorics*, a specialized branch of mathematics, is primarily concerned with the study of discrete objects and the relationships among them. Its central themes include the counting, arrangement, and selection of such objects, along with the methods used to carry out these tasks. Combinatorics provides a powerful set of tools for analyzing large collections of objects that satisfy specific properties. In this section, we revisit the most important results in combinatorics, focusing on their interpretation in terms of sets and ordered lists.

The *multiplication rule* is a fundamental principle that determines the number of possible outcomes in the Cartesian product of sets. According to this rule, if there are  $k$  sets  $A_1, A_2, \dots, A_k$ , and each set  $A_i$  contains  $n_i$  elements (for  $i = 1, \dots, k$ ), then the Cartesian product  $A_1 \times A_2 \times \dots \times A_k$  contains exactly  $n_1 n_2 \dots n_k$  elements. In particular, if a set  $A$  has  $n$  elements, then the  $k$ -fold Cartesian product  $A^k$  contains  $n^k$  elements.

The *inclusion-exclusion principle* determines the cardinality of the union of multiple sets based on the sizes of the individual sets and all possible intersections among them. Given  $k$  sets  $A_1, A_2, \dots, A_k$ , the formula is:

$$d\left(\bigcup_{i=1}^k A_i\right) = \sum_{i=1}^k d(A_i) - \sum_{i < j} d(A_i \cap A_j) + \sum_{i < j < l} d(A_i \cap A_j \cap A_l) - \dots - \sum_{i < j < l < m} d(A_i \cap A_j \cap A_l \cap A_m) + \dots + (-1)^{k+1} d(A_1 \cap \dots \cap A_k)$$

*Permutations* refer to the number of distinct ways in which the elements of a set can be arranged. Let  $A$  be a set with  $n$  elements. The number of ordered selections of  $k$  elements from  $n$  distinct elements without replacement, denoted by  $P_{n,k}$ , is given by  $P_{n,k} = n(n-1)\dots(n-k+1)$ . In the case where  $k = n$ , the total number of permutations is  $P_{n,n} = n(n-1)\dots 1 = n!$  where  $n!$  is read as " $n$  factorial".

■ **Example A.13** Consider the set  $\{a, b, c\}$ . There are six distinct permutations of its elements:  $\{a, b, c\}$ ,  $\{a, c, b\}$ ,  $\{b, a, c\}$ ,  $\{b, c, a\}$ ,  $\{c, a, b\}$ , and  $\{c, b, a\}$ . Each permutation represents a unique ordering of the elements in the original set. ■

The *pigeonhole principle* is a simple yet powerful concept. It asserts that if there are more pigeons than pigeonholes, then at least one pigeonhole must contain more than one pigeon. More formally, if  $n$  items are distributed among  $m$  containers and  $n > m$ , then at least one container must hold more than one item.

The logarithmic form of *Stirling's approximation* is particularly effective for estimating large factorials:

$$\log(n!) \approx \frac{1}{2} \log(2\pi) + \left(n + \frac{1}{2}\right) \log(n) - n$$

Numerous counting problems involve determining the number of subsets of a specific size within a given set. For a set with  $n$  elements, the total number of possible subsets is  $2^n$ , including both the empty set and the set itself. The number of subsets of size  $k$ , also known as the number of *combinations* of  $k$  elements from a set of  $n$ , is denoted by  $C_{n,k}$  and computed using the formula  $C_{n,k} = \frac{P_{n,k}}{k!} = \frac{n!}{k!(n-k)!}$ . The symbol  $\binom{n}{k}$  also represents the value  $C_{n,k}$ , which is known as the *binomial coefficient*. It is known that  $\binom{n}{0} = \binom{n}{n} = 1$  for all  $n$ , and  $\binom{n}{k} = \binom{n}{n-k}$  for all  $k = 0, 1, \dots, n$ . Additionally,  $\binom{n}{k} = 0$  whenever  $k > n$ .

■ **Example A.14** Consider the set  $\{a, b, c, d\}$ . There are 4 combinations of size 3:  $[a, b, c]$ ,  $[a, b, d]$ ,  $[a, c, d]$ , and  $[b, c, d]$ . Since combinations disregard order,  $[a, c, d]$  and  $[d, c, a]$  represent the same combination. ■

The *multinomial coefficient*, a generalization of the binomial coefficient to more than two categories, represents the number of ways to partition a set of objects into a fixed number of subsets, each containing a specified number of elements. Suppose we have a set with  $n$  elements that is to be divided into  $k$  subsets of sizes  $n_1, n_2, \dots, n_k$ , respectively. The multinomial coefficient, denoted as  $\binom{n}{n_1, n_2, \dots, n_k}$ , gives the number of such possible partitions and is calculated by the formula:

$$\binom{n}{n_1, n_2, \dots, n_k} = \frac{n!}{n_1! n_2! \dots n_k!}$$

where  $n_1 + n_2 + \dots + n_k = n$ .

The different combinations of replacement and ordering lead to four distinct counting scenarios, as summarized below. These conditions depend on whether the order of selection matters and whether elements can be selected more than once.

	Without replacement	With replacement
Ordered	$\frac{n!}{(n-r)!}$	$n^r$
Unordered	$\binom{n}{r}$	$\binom{n+r-1}{r}$

In the first row, we consider ordered selections: without replacement, the number of ways corresponds to the number of permutations of  $r$  elements from a set of  $n$ ; with replacement, each of the  $r$  positions can independently be filled with any of the  $n$  elements. In the second row, we consider unordered selections: without replacement, the count is given by the standard binomial coefficient; with replacement, the result corresponds to the number of multisets of size  $r$  formed from  $n$  distinct elements.

## A.4 Matrices

A *matrix*, denoted by  $A$ , of order  $m \times n$  is a rectangular array consisting of  $m$  *rows* and  $n$  *columns*, filled with a sequence of  $mn$  scalars. It is typically written as:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

The *entry*  $a_{ij}$  denotes the element located at the  $i$ -th row and  $j$ -th column of the matrix  $A$ . The *set of all matrices* of order  $m \times n$  is denoted by  $\mathcal{M}_{m \times n}$ . A *row matrix* is any matrix in the set  $\mathcal{M}_{1 \times n}$ , while a *column matrix* belongs

to  $\mathcal{M}_{m \times 1}$ . A *square matrix* is an element of  $\mathcal{M}_{n \times n}$ . The entries  $a_{ii}$  of a square matrix form its *main diagonal*. A *diagonal matrix* is a square matrix in which all entries outside the main diagonal are zero. The *identity matrix*, denoted by  $I$ , is a special diagonal matrix with all diagonal entries equal to 1.

The *transpose* of a matrix  $A \in \mathcal{M}_{m \times n}$  is defined as the matrix  $A^T \in \mathcal{M}_{n \times m}$ , whose entry at position  $(i, j)$  is equal to the entry at position  $(j, i)$  in  $A$ . If  $A = A^T$ , then  $A$  is called a *symmetric matrix*. A *submatrix* of a matrix is obtained by deleting one or more rows and/or columns.

■ **Example A.15** Consider the square matrix  $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ . The entry in position  $(2, 3)$  is 6, and the main diagonal consists of the elements 1, 5, and 9. The transpose of  $A$  is  $A^T = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$ . The matrix  $B = \begin{pmatrix} 1 & 3 \\ 4 & 6 \end{pmatrix}$  is a submatrix of  $A$ . ■

The addition of two matrices  $A$  and  $B$  of the same size produces a new matrix  $A + B$ , where each entry at position  $(i, j)$  is defined as  $(A + B)_{ij} = a_{ij} + b_{ij}$ . Matrix addition is associative, i.e.,  $(A + B) + C = A + (B + C)$ , and commutative, i.e.,  $A + B = B + A$ . It has a neutral element, the zero matrix, such that  $A + 0_{m \times n} = A$ , and each matrix has an additive inverse, since  $A + (-A) = 0_{m \times n}$ .

The product of a scalar  $\lambda$  and a matrix  $A$  yields another matrix, denoted by  $\lambda A$ , in which each entry at position  $(i, j)$  is given by  $(\lambda A)_{ij} = \lambda a_{ij}$ . Scalar multiplication is distributive over matrix addition, i.e.,  $\lambda(A + B) = \lambda A + \lambda B$ , and over scalar addition, i.e.,  $(\alpha + \beta)A = \alpha A + \beta A$ . It is also associative with respect to scalar multiplication, i.e.,  $(\alpha\beta)A = \alpha(\beta A)$ , and has a multiplicative identity, since  $1A = A$ .

The product of two matrices  $A_{m \times n}$  and  $B_{n \times p}$  results in a matrix  $AB_{m \times p}$ , where each entry at position  $(i, j)$  is given by  $(AB)_{ij} = \sum_{k=1}^n a_{ik}b_{kj}$ . Matrix multiplication is associative, meaning that  $(AB)D = A(BD)$ . It has a left identity element:  $AI_n = A$ , and a right identity element:  $I_mA = A$ . It is also compatible with scalar multiplication, so that  $\alpha(AB) = (\alpha A)B = A(\alpha B)$ . Furthermore, matrix multiplication is distributive over addition: from the right,  $A(B + C) = AB + AC$ , and from the left,  $(B + C)D = BD + CD$ .

The transpose operation satisfies the following properties:  $(A + B)^T = A^T + B^T$ ,  $(\lambda A)^T = \lambda A^T$ , and  $(AB)^T = B^T A^T$ .

■ **Example A.16** Given the matrices  $A = \begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix}$  and  $B = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$  we have that  $A + B = \begin{pmatrix} 6 & 8 \\ 10 & 13 \end{pmatrix}$ ,  $2A = \begin{pmatrix} 2 & 4 \\ 6 & 10 \end{pmatrix}$  and  $AB = \begin{pmatrix} 19 & 22 \\ 50 & 58 \end{pmatrix}$ . ■

A square matrix  $A$  is said to be *invertible* or *non-singular* if there exists a matrix  $B$  such that  $AB = BA = I$ . If  $A$  is non-singular, the matrix  $B$  is unique and is called the *inverse* of  $A$ , denoted by  $A^{-1}$ . A matrix  $A$  is called *orthogonal* if its transpose is equal to its inverse, that is,  $A^T = A^{-1}$ . The

columns and rows of an orthogonal matrix are referred to as *orthonormal vectors*.

The *determinant* is a special scalar value that can be computed from the elements of a square matrix. For a matrix  $A$ , the determinant is denoted by  $\det(A)$ . It can be calculated using the Leibniz formula:

$$\det(A) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \cdot a_{1,\sigma(1)} \cdot a_{2,\sigma(2)} \cdots a_{n,\sigma(n)}$$

where  $S_n$  denotes the set of all permutations of the integers 1 to  $n$ , and  $\operatorname{sgn}(\sigma)$  is the sign of the permutation  $\sigma$ , equal to +1 for even permutations and -1 for odd permutations. A matrix is invertible if and only if its determinant is nonzero.

■ **Example A.17** The determinant of a  $3 \times 3$  matrix  $A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$  is computed as:

$$\det(A) = aei + bfg + cdh - ceg - bdi - afh$$

■

For a given matrix  $A$ , the *rank*, denoted as  $\operatorname{rank}(A)$ , is the maximum number of linearly independent rows or columns in the matrix.

A number  $\lambda$  and a nonzero vector  $\mathbf{v}$  such that  $A\mathbf{v} = \lambda\mathbf{v}$  are called an *eigenvalue* and an *eigenvector* of  $A$ , respectively.

Matrix decomposition is the process of transforming a matrix into a more tractable form while preserving certain properties, such as the determinant or rank. The *singular value decomposition* (SVD) of a matrix  $A$  of order  $m \times n$  is a factorization of the form  $A = U\Sigma V^T$ , where  $U$  is an  $m \times m$  orthogonal matrix,  $\Sigma$  is an  $m \times n$  diagonal matrix with nonnegative entries, and  $V^T$  is the transpose of an  $n \times n$  orthogonal matrix.

## A.5 Graphs

A *graph*<sup>2</sup>  $G$  is defined as an ordered pair  $(V, E)$ , where  $V$  is a set of *vertices*, and  $E$  is a set of *edges*. Each element of  $E$  is an unordered pair  $\{u, v\}$ , where  $u, v \in V$  are distinct (loops are not allowed). Two vertices  $u$  and  $v$  are said to be *adjacent* if  $\{u, v\} \in E$ ; in that case, they are referred to as the *endpoints* of the edge. If the set  $V$  is infinite, the graph is called an *infinite graph*. In this book, however, we consider only finite graphs. Given a graph  $G = (V, E)$ , its

<sup>2</sup>The definition of a graph given here corresponds to that of a *simple graph*, as commonly found in discrete mathematics literature.

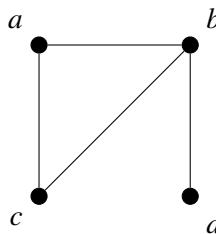


Figure A.2: An Example of Graph

*adjacency matrix* is a square matrix of size  $d(V) \times d(V)$ , denoted  $A$ , where  $A_{uv} = 1$  if  $\{u, v\} \in E$ , and  $A_{uv} = 0$  otherwise.

Graphs are typically depicted as a set of dots (vertices) connected by lines (edges).

■ **Example A.18** Let  $V = a, b, c, d$  and  $E = a, b, a, c, a, d, b, c$ . The graph  $G = (V, E)$  is illustrated in Figure A.2. ■

When a vertex  $v$  is an endpoint of an edge  $e$ , the edge  $e$  is said to be *incident* on  $v$ . The *degree* of a vertex  $v$ , denoted  $\deg(v)$ , is the number of edges incident on  $v$ . A vertex with degree zero is called an *isolated vertex*, while a vertex with degree one is termed a *pendant vertex*. The *neighborhood* of a vertex  $v$ , denoted  $N(v)$ , is the set of all vertices adjacent to  $v$ . For a subset  $A \subset V$ , the neighborhood of  $A$  is defined as  $N(A) = \bigcup_{v \in A} N(v)$ . A *path* in a graph is a sequence of distinct vertices  $v_0, v_1, \dots, v_k$  such that  $v_i$  and  $v_{i+1}$  are adjacent for each  $0 \leq i < k$ . The path is called a *simple path* if no vertex is repeated. A graph is said to be *connected* if there exists a path between every pair of vertices. If  $v_0 = v_k$ , the path is called a *cycle*. A cycle is *simple* if it includes at least three vertices and only the first and last vertices are repeated.

■ **Example A.19** In a graph  $G = (V, E)$ , the *handshaking theorem* states that  $\sum_{v \in V} \deg(v) = 2m$ , where  $m = d(E)$ , since each edge contributes to the degree count of two vertices. ■

If the vertex pairs  $(u, v)$  are arranged as ordered pairs, the graph is called a *directed graph*. In this context,  $u$  is referred to as the *initial vertex* and  $v$  as the *terminal vertex*. For a directed graph  $G$ , the *in-degree* of a vertex  $v$ , denoted by  $\text{indeg}(v)$ , is the number of edges for which  $v$  is the terminal vertex. The *out-degree* of a vertex  $v$ , denoted by  $\text{outdeg}(v)$ , is the number of edges for which  $v$  is the initial vertex. A directed graph is said to be *strongly connected* if there exists a directed path from every vertex to every other vertex. Directed graphs are typically represented using arrows instead of lines to indicate the direction of edges.

A graph  $G$  is classified as *emph bipartite* if its vertex set  $V$  can be partitioned into two subsets  $V_1$  and  $V_2$  such that every edge of  $G$  connects a vertex from  $V_1$  to a vertex from  $V_2$ . Bipartite graphs are commonly denoted as  $G = (V_1, V_2, E)$ . The degrees of the vertices in a bipartite graph satisfy the *degree sum formula*,  $\sum_{u \in V_1} \deg(u) = \sum_{v \in V_2} \deg(v) = d(E)$ .

A graph  $G(V', E')$  is called a *subgraph* of a graph  $G(V, E)$  if  $V' \subseteq V$  and  $E' \subseteq E$ , where every edge in  $E'$  has both endpoints in  $V'$ . A graph  $G$  is said to be a *labeled graph* if its edges and/or vertices are assigned specific data. In particular, if each edge  $e$  in  $G$  is assigned a nonnegative number  $w(e)$ , then  $w(e)$  is referred to as the *weight* of the edge  $e$ .

A specific type of graph that plays a fundamental role in this book is the *tree*. A tree is defined as a non-empty graph in which any pair of vertices is connected by a unique path. A tree typically includes a specially designated vertex known as the *root*, and every edge is conceptually oriented away from this root.

■ **Example A.20** An alternative definition of a tree, grounded in set theory, characterizes it as a partially ordered set  $(T, <)$  such that for every  $t \in T$ , the set  $S = s \in T : s < t$  has a least element, that is, an element smaller than all other elements in  $S$ . ■

Given a tree  $T$ , for any vertex  $v$  other than the root, the *parent* of  $v$  is the unique vertex  $u$  such that there is an edge directly connecting  $u$  to  $v$ . Conversely, if  $u$  is the parent of  $v$ , then  $v$  is called a *child* of  $u$ . Any other vertex in the tree that shares the same parent as  $v$  is referred to as a *sibling* of  $v$ . The *ancestors* of a vertex consist of all vertices along the unique path from the root to that vertex, excluding the vertex itself but including the root. The *descendants* of a vertex  $v$  are all vertices that have  $v$  as an ancestor. A vertex with no children is called a *leaf*, whereas vertices that have one or more children are known as *branches*. The *depth* of a vertex  $v$  is the length of the unique path from the root to  $v$ . The *height* of the tree is the maximum depth among all its vertices.

■ **Example A.21** For the tree illustrated in Figure A.3, the root vertex is  $a$ ;  $c$  is the parent of  $d$ , so  $d$  is a child of  $c$ ;  $d$  and  $g$  are siblings; the ancestors of  $d$  are  $a$  and  $c$ ; the descendants of  $c$  are  $d$ ,  $e$ , and  $f$ ; the leaf vertices are  $b$ ,  $e$ ,  $f$ , and  $g$ ;  $a$  and  $c$  are branches; the depth of  $d$  is 3; the height of the tree is 4. ■

Given a vertex  $v$  in a tree, the *subtree* rooted at  $v$  is the subgraph that consists of  $v$ , all its descendants, and all edges connecting these vertices. A tree is called a  *$k$ -ary tree* if each branch has at most  $k$  children. If every branch has exactly  $k$  children, the tree is referred to as a *full  $k$ -ary tree*. A  $k$ -ary tree with  $k = 2$  is specifically known as a *binary tree*. A  $k$ -ary tree of

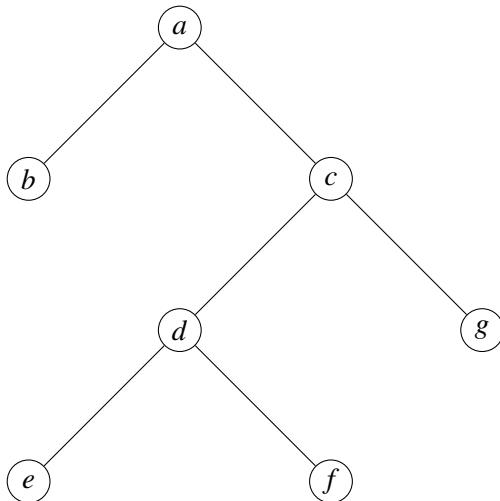


Figure A.3: An Example of a Tree

height  $h$  is said to be *balanced* if all its leaves are located at depth  $h$  or  $h - 1$ .

■ **Example A.22** A tree with  $n$  vertices contains exactly  $n - 1$  edges. A full  $k$ -ary tree with  $i$  branches contains  $m = ki + 1$  vertices. ■

The procedure of visiting each node in a tree exactly once is known as *tree traversal*. Traversal methods are categorized based on the order in which nodes are visited, with the two main types being *depth-first* and *breadth-first*. In a *depth-first traversal*, the algorithm starts at the root node and explores as far as possible along each branch before backtracking to explore siblings. There are three common strategies for visiting nodes in a depth-first manner: *in-order*, *pre-order*, and *post-order*.

The following code snippet, written in a C-like syntax, demonstrates a recursive pre-order depth-first traversal algorithm used to print the nodes of a binary tree:

```

void print_tree(binary_tree *tree) {
    if (!is_empty(tree)) {
        printf("%c\n", tree->node);
        print_tree(tree->left_branch);
        print_tree(tree->right_branch);
    }
}
  
```

Conversely, in a *breadth-first traversal*, the algorithm begins at the root of the tree and explores all nodes at the current depth level before proceeding

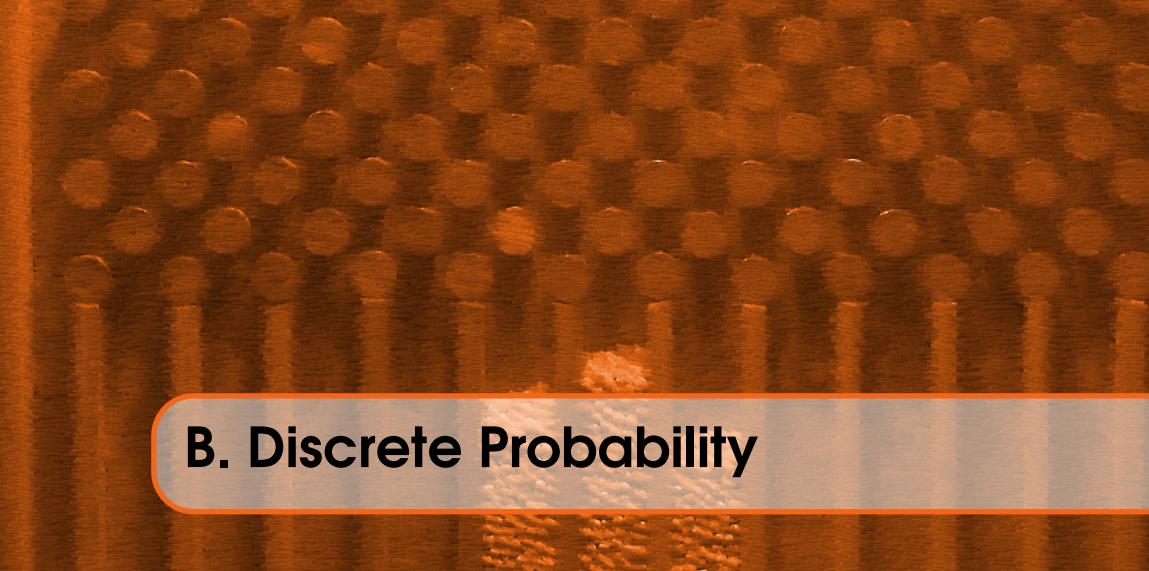
to nodes at the next depth level. Implementing breadth-first traversal typically requires the use of auxiliary data structures, such as queues. For examples and implementation details of such algorithms, please refer to the references section.

■ **Example A.23** For the tree described in Example ??, a pre-order depth-first traversal produces the string "abcdefg". In contrast, a pre-order breadth-first traversal yields the string "abcdgef". ■

## References

The book *Discrete Mathematics* by Johnsonbaugh [Joh09] is designed for undergraduate students enrolled in a one- or two-semester course in discrete mathematics, and it provides thorough coverage of fundamental topics in the field. *Introduction to the Theory of Computation* by Sipser [Sip12] offers a comprehensive and accessible introduction to key topics in computational theory. It is widely praised for its clear presentation and in-depth treatment of automata, formal languages, and complexity theory. *Introduction to Algorithms* by Cormen et al. [Cor+90], often referred to as "CLRS" after the initials of its authors, is a foundational text in the study of algorithms. It covers a broad range of subjects, including graph algorithms, and is known for its depth and rigor. Lastly, *Matrix Computations* by Golub and Van Loan [GV13] focuses on a wide array of topics in matrix theory, with particular emphasis on computational techniques—making it especially valuable for those involved in numerical and computational studies.





## B. Discrete Probability

*The purpose of models is not to fit the data  
but to sharpen the questions.*

Samuel Karlin

Probability theory is the branch of mathematics that studies random experiments and phenomena. It assigns a numerical value to each possible outcome of an experiment, reflecting how likely that outcome is to occur. Even when the outcome of a specific experiment cannot be predicted in advance, probability theory allows us to analyze its properties and derive meaningful insights. For example, while we cannot foresee the next number drawn in a lottery, probability theory helps explain why spending all our savings on lottery tickets is an unwise strategy for becoming wealthy.

The significance of probability theory goes far beyond games of chance. It forms the mathematical foundation of statistical inference, allowing us to draw conclusions from data, and it underpins many machine learning algorithms used for classification and prediction with large datasets. Moreover, probability theory plays a vital role in fields such as finance, risk management, and the natural sciences, where understanding uncertainty and variability is crucial.

In this chapter, we focus on the area of discrete probability. In this version of the theory, the possible outcomes of an event are either finite or, at most, countably infinite. Our interest in discrete probability arises for two main reasons: first, due to its practical applications in learning from data; and second, because it has deep connections to several theoretical concepts explored in this book, such as the length of optimal codes, the probability that a random machine will halt, and the derivation of universal distributions based on Kolmogorov complexity. All of these connections are directly relevant to our theory of nescience.

We will cover only the most important concepts and results of probability theory. The material has been selected based on its relevance to the theory of nescience. For instance, moment-generating functions are not included. For a more comprehensive introduction to probability theory, refer to the references at the end of this chapter.

Our approach to probability theory will be formal and axiomatic. We will begin by stating a basic set of fundamental axioms, from which we will derive the main results and properties. Axioms are essential in mathematical theory, as they provide the foundation for constructing a consistent, universal, and rigorous framework. In the context of probability theory, they allow us to define and manipulate the elusive concept of probability in a way that is both precise and widely applicable.

## B.1 Interpretations of Probability

The concept of *probability* presents a profound intellectual challenge. Consider the case of rolling a die and computing the probability of obtaining an even number. The die has six distinct outcomes, and since half of them are even, we conclude that the probability is  $3/6$ , or equivalently  $1/2$ . This reflects the *classical interpretation* of probability, which states that in an experiment where all finite outcomes are equally likely, the probability of an event is given by the ratio of favorable outcomes to the total number of possible outcomes.

However, this interpretation faces a problem of circularity: the notion of "equally likely" often assumes a symmetry in the outcomes, but in formal contexts this can be criticized for relying implicitly on the very concept of probability it seeks to define. An alternative approach is the *principle of indifference*, which holds that in the absence of any relevant evidence, all outcomes should be assigned equal probability. This principle, however, breaks down when there is evidence suggesting that the outcomes are not equally probable. This principle, however, only applies in the absence of evidence. When information is available, such as knowing the die is loaded,

it cannot be justifiably used, and the assumption of equally likely outcomes breaks down.

The *frequentist interpretation* of probability posits that one should roll the die multiple times and compare the frequency of even outcomes to the total number of rolls. The fundamental idea is to repeat the experiment under similar conditions and assign to each outcome a probability equal to its relative frequency.

This interpretation, however, faces two major limitations. First, the notion of "similar conditions" is vague and lacks a precise definition; after all, if conditions were exactly identical in a deterministic system, the outcomes would also be identical. In practice, the assumption is that conditions are similar enough to allow statistical regularities to emerge. Second, the concept of a "large number of repetitions" is ill-defined—technically, the experiment must be repeated an infinite number of times.

From a practical standpoint, implementing the frequentist interpretation presents significant challenges. Some experiments, such as estimating the probability of a candidate winning an election, cannot be repeated. Moreover, probability is defined only in the context of a sequence of trials, which makes it impossible to compute the probability of a single, non-repeatable event. Finally, the interpretation assumes the existence of a limiting relative frequency, a condition that is not always satisfied, as illustrated by certain financial time series.

The *subjective interpretation*, representing a third approach to the concept of probability, proposes assigning probabilities to events based on our degree of belief: the stronger our conviction that an event will occur, the higher the probability we assign to it. However, not all possible probability assignments are acceptable; certain coherence conditions must be met. For example, assigning probabilities in a way that guarantees a loss in a betting system, a scenario known as a *Dutch book*, violates these conditions.

It turns out that the conditions necessary and sufficient to avoid such inconsistencies align precisely with the axioms of probability that will be introduced later. Thus, we are free to assign probabilities according to our beliefs, as long as these assignments remain consistent with those axioms.

A key limitation of the subjective interpretation is that degrees of belief can vary widely between individuals. The *Bayesian interpretation* offers a refinement: we begin with an initial (prior) assignment of probabilities and update them as new evidence becomes available. As more evidence is accumulated, revised probabilities tend to converge to values that are more consistent with observed data, and under certain conditions, may approximate the long-run frequencies or objective probabilities (if such exist). Nevertheless, assigning probabilities to an infinite number of events is generally

infeasible for humans.

Currently, the notion of probability is defined axiomatically through the *axiomatic interpretation*. This approach abandons the attempt to define probability explicitly, and instead accepts certain fundamental properties as given.

Mathematically, probability is defined as a real number between 0 and 1, where a probability of 0 corresponds to an impossible event, and a probability of 1 corresponds to an event that is certain to occur. Intuitively, however, we often express probabilities as percentages, for example, saying that it will rain tomorrow with a 70% probability, which is also valid interpretation of the concept of probability.

Additional properties are also required. For example, if two events  $A$  and  $B$ , with probabilities  $P(A)$  and  $P(B)$  respectively, are disjoint, then the probability of either  $A$  or  $B$  occurring should be  $P(A) + P(B)$ . If  $A$  and  $B$  can occur simultaneously and are independent (independence being a concept that is mathematically well-defined but often conceptually subtle), then the probability of both occurring together should be  $P(A)P(B)$ . Furthermore, the probability of  $A$  occurring given that  $B$  has already occurred should equal the fraction of the probability of  $A$  that also lies within  $B$ .

## B.2 Foundations of Probability Theory

Probability theory is fundamentally concerned with assigning numerical values to specific events drawn from a sample space <sup>1</sup>.

**Definition B.2.1** Given  $(\Omega, \mathcal{A})$  as a field over a non-empty discrete set,  $\Omega$  is called the *sample space*, its elements are called *outcomes*, and the elements of  $\mathcal{A}$  are referred to as *events*. Specifically,  $\Omega$  is known as the *certain event*, while the empty set  $\emptyset$  is called the *impossible event*.

As discussed in Section A.1, since  $(\Omega, \mathcal{A})$  is a field, it follows that  $\Omega \in \mathcal{A}$  and  $\emptyset \in \mathcal{A}$ . Moreover, the union of a finite collection of events is also an event:  $A_1 \cup A_2 \cup \dots \cup A_n \in \mathcal{A}$ , and likewise, the intersection of a finite collection of events is an event:  $A_1 \cap A_2 \cap \dots \cap A_n \in \mathcal{A}$ .

As mentioned in the introduction to this chapter, our primary focus is

<sup>1</sup>The term "event" in this context may seem somewhat counterintuitive, as it typically suggests that something has happened, an implication not always applicable in the mathematical setting. For example, consider the sample space of all possible outcomes from tossing a fair coin. A subset of this sample space might be the empty set, which represents no outcome at all. From a conventional perspective, this does not correspond to anything "happening," which may confuse readers unfamiliar with the mathematical usage of the term. Nevertheless, for consistency and clarity, we will continue to use the term "events" to refer to subsets of the sample space.

on discrete mathematics. Accordingly, we will concentrate on probabilities defined over discrete sets, whether finite or countably infinite. Extending the concept of probability to continuous sets requires the use of  $\sigma$ -algebras of sets instead of fields, along with the tools of measure theory. On a philosophical level, one might argue that all sample spaces must be countable, since physical measurements cannot be made with infinite precision. In practice, this is indeed the case: any empirical observation is subject to finite resolution.

The standard axiomatization used in probability theory is encapsulated in the framework of the *Kolmogorov axioms*<sup>2</sup>.

**Definition B.2.2** (*Kolmogorov's Axioms*) A *probability* is a real number  $P(A) \in \mathbb{R}$  assigned to each event  $A \in \mathcal{A}$  in the field  $(\Omega, \mathcal{A})$ , subject to the following axioms:

**Axiom 1** Non-negativity: For all events  $A \in \mathcal{A}$ , we have  $P(A) \geq 0$ .

**Axiom 2** Normalization: The probability of the certain event is one, i.e.,  $P(\Omega) = 1$ .

**Axiom 3** Additivity: For any finite sequence of pairwise disjoint events  $A_1, A_2, \dots, A_n \in \mathcal{A}$ , the probability of their union equals the sum of their individual probabilities:

$$P\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n P(A_i).$$

The triplet  $(\Omega, \mathcal{A}, P)$  is called a *probability space*.

Despite their foundational importance, the Kolmogorov axioms present certain limitations. While they establish essential constraints on any probability function (such as non-negativity, normalization, and additivity) they do not prescribe how to assign probabilities to specific events. In other words, the axioms define the formal rules that probabilities must satisfy but remain silent on how those probabilities should be determined in practice. This is a consequence of their high level of generality: any mathematical structure that satisfies these properties can be regarded as a valid probability model. As such, they are abstract enough to encompass not only probability measures but also other normalized physical measures such as mass, volume, or charge.

From an intuitive standpoint, it might seem more natural to assign probabilities directly to the individual elements of the sample space, especially in

---

<sup>2</sup>In discrete probability theory, the sample space consists of a finite or countably infinite set of distinct outcomes. As a result, events are typically made up of individual, separable outcomes. Since probabilities are assigned directly to these discrete events, only finite unions of disjoint events need to be considered in Axiom 3 to account for all practical cases.

the discrete case where each outcome is well-defined and countable. However, in the formal theory of probability, probabilities are assigned to events, which are subsets of the sample space. This approach, though less intuitive at first glance, ensures compatibility with the measure-theoretic framework that underlies modern probability theory. In the continuous case, individual outcomes (such as a specific real number) typically have probability zero, and probability must be defined directly over sets (e.g., intervals) rather than points.

■ **Example B.1** Consider a sample space  $\Omega$  consisting of  $n$  equally probable outcomes. If an event  $A \subset \Omega$  contains  $d(A) = m$  elements, then the probability of event  $A$  is given by  $P(A) = m/n$ . ■

We now proceed to establish some fundamental results in probability theory, beginning with the calculation of the probability of the complement of an event, that is, the probability that the event does not occur.

**Proposition B.2.1** For any event  $A$ , it holds that  $P(A^c) = 1 - P(A)$ .

*Proof.* The sets  $A$  and  $A^c$  are disjoint, and their union satisfies  $A \cup A^c = \Omega$ . By Axiom 3 (additivity), we have  $P(A \cup A^c) = P(A) + P(A^c)$ . By Axiom 2 (normalization), we know that  $P(A \cup A^c) = P(\Omega) = 1$ . Combining both equations, we obtain  $P(A) + P(A^c) = 1$ , which completes the proof. ■

As a direct consequence of the previous proposition, we can deduce the probability of the impossible event.

**Proposition B.2.2** The probability of the impossible event is zero; that is,  $P(\emptyset) = 0$ .

*Proof.* Since  $\emptyset = \Omega^c$ , we apply the complement rule:

$$P(\emptyset) = P(\Omega^c) = 1 - P(\Omega) = 1 - 1 = 0.$$

■

As expected, sub-events (i.e., subsets) are associated with probabilities no greater than those of the events containing them.

**Proposition B.2.3** If  $A \subset B$ , then  $P(A) \leq P(B)$ .

*Proof.* The event  $B$  can be written as the disjoint union of  $A$  and  $B \cap A^c$ . By Axiom 3:  $P(B) = P(A) + P(B \cap A^c)$ . Since  $P(B \cap A^c) \geq 0$ , it follows that  $P(A) \leq P(B)$ . ■

With these basic properties established, we can now confirm that all probabilities lie between zero and one.

**Proposition B.2.4** For any event  $A$ , we have  $0 \leq P(A) \leq 1$ .

*Proof.* By Axiom 1,  $P(A) \geq 0$ . Since  $A \subset \Omega$ , the previous proposition implies  $P(A) \leq P(\Omega) = 1$ . ■

Axiom 3 allows us to compute the probability of the union of disjoint events. However, it does not directly apply to cases involving non-disjoint events. The following proposition provides a formula for computing the probability of the union of two events that may overlap.

**Proposition B.2.5** For any two events  $A$  and  $B$ , we have:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

*Proof.* The union of  $A$  and  $B$  can be expressed as  $A \cup B = (A \setminus B) \cup B$ . Since  $A \setminus B$  and  $B$  are disjoint, by Axiom 3 we obtain:

$$P(A \cup B) = P(B) + P(A \setminus B).$$

Now, observe that  $P(A) = P(A \setminus B) + P(A \cap B)$ , which implies:

$$P(A \setminus B) = P(A) - P(A \cap B).$$

Substituting this into the previous expression yields:

$$P(A \cup B) = P(B) + P(A) - P(A \cap B),$$

as required. ■

This result generalizes to any finite number of events via the *principle of inclusion-exclusion* (see Section A.3):

$$P\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n P(A_i) - \sum_{i < j} P(A_i \cap A_j) + \sum_{i < j < k} P(A_i \cap A_j \cap A_k) - \sum_{i < j < k < l} P(A_i \cap A_j \cap A_k \cap A_l) + \dots + (-1)^{n+1} P(A_1 \cap \dots \cap A_n)$$

To ensure that a probability function satisfies Kolmogorov's axioms, we must define it using methods that are guaranteed to produce valid probability measures. One such method involves assigning probabilities to individual elements of a finite or countable sample space and extending this assignment to subsets via summation.

**Proposition B.2.6** Let  $(\Omega, \mathcal{A})$  be a field over a non-empty discrete set, and let  $p_1, p_2, \dots$  be a sequence of nonnegative real numbers such that  $\sum_{i=1}^{\infty} p_i = 1$ . Define a function  $P : \mathcal{A} \rightarrow [0, 1]$  by  $P(A) = \sum_{\{i : s_i \in A\}} p_i$ , with the convention that the sum over an empty index set is 0. Then  $P$  is a probability on  $(\Omega, \mathcal{A})$ .

*Proof.* We must verify that the function  $P$  satisfies the three Kolmogorov axioms:

Axiom 1 (Nonnegativity): Let  $A \in \mathcal{B}$ . Since each  $p_i \geq 0$ , the sum

$$P(A) = \sum_{\{i : s_i \in A\}} p_i$$

is a (possibly infinite) sum of nonnegative terms. Hence,  $P(A) \geq 0$ .

Axiom 2 (Normalization): We have that

$$P(\Omega) = \sum_{\{i : s_i \in \Omega\}} p_i = \sum_{i=1}^{\infty} p_i = 1,$$

by assumption. Hence,  $P(\Omega) = 1$ .

Axiom 3 (Finite Additivity): Let  $A_1, \dots, A_n \in \mathcal{B}$  be pairwise disjoint events. Define  $A = \bigcup_{j=1}^n A_j$ . Then the sets  $\{i : s_i \in A_j\}$  are disjoint for distinct  $j$ , and we have:

$$P(A) = \sum_{\{i : s_i \in A\}} p_i = \sum_{j=1}^n \sum_{\{i : s_i \in A_j\}} p_i = \sum_{j=1}^n P(A_j),$$

where we used the disjointness of the  $A_j$ 's and associativity of finite sums.

Hence,  $P$  satisfies all three axioms of probability (with finite additivity), and is therefore a valid probability function on  $(\Omega, \mathcal{A})$ . ■

Finally, we introduce the concept of a probability mass function<sup>3</sup>. A probability mass function assigns a probability to each event in a discrete sample space.

**Definition B.2.3** Let  $(\Omega, \mathcal{A}, P)$  be a discrete probability space. A *probability mass function* is a real-valued function  $f : \mathcal{A} \rightarrow [0, 1]$  such that  $f(A) = P(A)$  for every  $A \in \mathcal{A}$ .

In Example B.1, we introduced a discrete probability space  $(\Omega, \mathcal{A}, P)$  consisting of  $n$  equally probable elements. The corresponding probability mass function is defined as  $f : \mathcal{A} \rightarrow [0, 1]$ , with

$$f(A) = \frac{d(A)}{n}, \quad \text{for all } A \in \mathcal{A},$$

---

<sup>3</sup>Most probability textbooks define the probability mass function as a real-valued function of the form  $f : \mathbb{R} \rightarrow [0, 1]$ . However, this definition requires the prior introduction of the concept of a random variable. Since random variables play a more secondary role in the context of discrete probability, we have postponed their introduction to a later subsection. For this reason, we have provided a definition of the probability mass function based directly on events.

where  $d(A)$  denotes the number of elements in the event  $A$ .

In many discrete settings, it is common to treat each individual element  $\omega \in \Omega$  as an elementary event. That is, we assume that every set  $\{\omega\}$  belongs to  $\mathcal{A}$ . In this case, we can define the probability mass function as a function  $f : \Omega \rightarrow [0, 1]$ , where  $f(\omega) = P(\{\omega\})$  for each  $\omega \in \Omega$ . This formulation is often more natural, as it allows us to specify probabilities at the level of individual outcomes.

The following proposition shows that, under this assumption, the probability of any event  $A \in \mathcal{A}$  can be computed by summing the probabilities of the individual outcomes that make up the event.

**Proposition B.2.7** Let  $(\Omega, \mathcal{A}, P)$  be a discrete probability space such that  $\{\omega\} \in \mathcal{A}$  for every  $\omega \in \Omega$ . Define the function  $f : \Omega \rightarrow [0, 1]$  by  $f(\omega) = P(\{\omega\})$ . Then for any event  $A \in \mathcal{A}$ , the probability of  $A$  is given by:

$$P(A) = \sum_{\omega \in A} f(\omega).$$

*Proof.* Since  $A \in \mathcal{A}$ , and  $\{\omega\} \in \mathcal{A}$  for each  $\omega \in \Omega$ , it follows that every singleton  $\{\omega\} \subseteq A$  is a measurable event. Furthermore, the singletons  $\{\omega\}$  for  $\omega \in A$  are pairwise disjoint and their union is exactly  $A$ . Thus, by Axiom 3 (finite additivity),

$$P(A) = P\left(\bigcup_{\omega \in A} \{\omega\}\right) = \sum_{\omega \in A} P(\{\omega\}) = \sum_{\omega \in A} f(\omega).$$

■

### B.3 Conditional Probability

The concept of conditional probability plays a fundamental role in statistical learning. Conditional probability allows us to refine the probability of an event based on new information or imposed conditions. Within the axiomatic framework established by Kolmogorov, conditional probability is introduced as a definition.

**Definition B.3.1** Let  $A$  and  $B$  be two events such that  $P(B) > 0$ . The *conditional probability* of  $A$  given  $B$ , denoted by  $P(A | B)$ , is defined as:

$$P(A | B) = \frac{P(A \cap B)}{P(B)}.$$

Because it satisfies the Kolmogorov axioms, a conditional probability is itself a valid probability. Note that  $P(A | B)$  is undefined when  $P(B) = 0$ .

Some scholars argue that, given its central role in probability theory, conditional probability should be a property logically deduced from the foundational axioms, rather than introduced as a definition. This viewpoint naturally calls for an extension of Definition B.2.2 to include additional properties. Unfortunately, there is no consensus among mathematicians or philosophers on how such an extension should be formalized.

The standard interpretation of conditional probability views it as the updated probability of event  $A$  given that event  $B$  has occurred. This interpretation, however, may implicitly suggest a temporal or even causal relationship between  $B$  and  $A$ , a suggestion that is not necessarily valid.

■ **Example B.2** Suppose we are playing a game with a standard deck of 52 cards and we draw two cards. Let event  $A$  be “drawing at least one heart,” and event  $B$  be “drawing at least one queen.” These two events are dependent, as the occurrence of  $B$  affects the probability of  $A$ . However, the events are not temporally related, since both cards are drawn simultaneously, one does not occur before the other. This example illustrates that dependency in probability theory does not require a temporal sequence between the events involved. ■

The probability of two events occurring together (though not necessarily in a temporal sequence, as discussed in Example B.2), given their conditional relationship, is captured by the formula  $P(A \cap B) = P(A | B)P(B)$ . This equation often provides a more intuitive understanding of the concept of conditional probability. In fact, several authors have argued that this formulation should serve as the primary definition of conditional probability, rather than the quotient-based approach.

This formula generalizes to any finite number of events through what is known as the *multiplication rule*, which is expressed as:

$$P(A_1 \cap A_2 \cap \dots \cap A_n) = P(A_1)P(A_2 | A_1) \dots P(A_n | A_1 \cap A_2 \cap \dots \cap A_{n-1}) \quad (\text{B.1})$$

The concept of event independence plays a central role in probability theory and statistical learning.

■ **Definition B.3.2** Two events  $A$  and  $B$  are declared to be *independent* if  $P(A \cap B) = P(A)P(B)$ .

Intuitively, events  $A$  and  $B$  are considered independent if knowing that event  $B$  has occurred does not affect the probability of event  $A$ . This property follows logically from the definition of independence.

**Proposition B.3.1** Let  $A$  and  $B$  be two events such that  $P(A) > 0$  and  $P(B) > 0$ . Then  $A$  and  $B$  are independent if and only if  $P(A | B) = P(A)$ .

*Proof.* Assume  $A$  and  $B$  are independent, i.e.,  $P(A \cap B) = P(A)P(B)$ . Then,

$$P(A | B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)P(B)}{P(B)} = P(A).$$

Conversely, assume that  $P(A | B) = P(A)$ . Then by the multiplication rule,

$$P(A \cap B) = P(A | B)P(B) = P(A)P(B),$$

which confirms independence. ■

Note that if  $P(A | B) = P(A)$ , then it also follows that  $P(B | A) = P(B)$ . This is a direct consequence of the fact that the joint probability  $P(A \cap B)$  can be expressed as  $P(A)P(B)$ .

As in the case of conditional probability, some scholars have argued that independence should not be treated merely as a definition, but rather as a logical consequence of the foundational axioms of probability. However, incorporating independence as a derived property would require extending or modifying the axiomatic system, a topic for which no universally accepted approach currently exists.

The concept of independence can be generalized to collections of more than two events. The events  $A_1, \dots, A_n$  are said to be *mutually independent* if, for every subset of events  $A_{i_1}, \dots, A_{i_j}$  with  $j \in 2, 3, \dots, n$ , the following condition holds:

$$P(A_{i_1} \cap \dots \cap A_{i_j}) = P(A_{i_1}) \cdots P(A_{i_j}).$$

It is important to emphasize that *pairwise independence*, that is, the condition  $P(A_i \cap A_j) = P(A_i)P(A_j)$  for all  $i \neq j$ , is not sufficient to guarantee mutual independence. Mutual independence requires that the product rule holds for *every* subset of two or more events, not just for pairs.

■ **Example B.3** Confusion often arises when distinguishing between mutually exclusive (or disjoint) events and independent events. For two mutually exclusive events  $A$  and  $B$ , the expression for the conditional probability  $P(A | B)$  becomes problematic: if  $B$  occurs, then  $A$  cannot occur by definition, making  $P(A | B)$  either undefined (if  $P(B) = 0$ ) or equal to zero (if  $P(B) > 0$ ). Similarly, computing the conditional probability of  $A$  given  $B$  when  $P(B) = 0$  is not well-defined.

Although Definition B.3.2 does not explicitly prohibit  $A$  and  $B$  from being mutually exclusive, the two concepts are fundamentally incompatible unless one (or both) of the events has probability zero. In fact, two mutually exclusive events  $A$  and  $B$  are independent if and only if  $P(A) = 0, P(B) = 0$ , or both. ■

An interesting situation occurs when two events  $A$  and  $B$  are not independent in general, but become independent when conditioned on the occurrence of a third event  $C$ .

**Definition B.3.3** Let  $A$ ,  $B$ , and  $C$  be events such that  $P(B \cap C) > 0$ . We say that  $A$  and  $B$  are *conditionally independent* given  $C$  if  $P(A | B \cap C) = P(A | C)$ .

Next example illustrates that two events may be independent but fail to remain conditionally independent given a third event.

■ **Example B.4** Consider the experiment of rolling two fair dice. The outcomes of the two dice are independent, since knowing the result of one die does not provide any information about the result of the other. Now suppose we observe that the first die shows a four, and we are also given that the sum of the two dice is an odd number (event  $C$ ). This additional condition restricts the possible outcomes of the second die: it must now show an odd number to make the total odd. As a result, the outcome of the second die is no longer independent of the first, once we condition on the sum being odd.

The following theorem presents Bayes' rule, a fundamental result that underpins an important statistical learning technique known as Bayesian inference (see Section F.1.2)<sup>4</sup>.

**Theorem B.3.2 — Bayes' Theorem.** Let  $A$  and  $B$  be two events such that  $P(B) \neq 0$ . Then we have that

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$

In this context,  $P(A)$  is called the *prior probability*, while  $P(A | B)$  is known as the *posterior probability*.

*Proof.* From the definition of conditional probability, we have:

$$P(A | B) = \frac{P(A \cap B)}{P(B)} \quad (\text{provided } P(B) \neq 0),$$

<sup>4</sup>As shown in the proof, Bayes' theorem follows directly from the definition of conditional probability. However, despite its practical utility, this derivation raises important philosophical questions regarding its foundational status. If conditional probability is taken as a definition rather than a derived consequence of more fundamental axioms, then Bayes' theorem rests on the same definitional foundation. As such, it functions effectively within this framework, but its theoretical justification depends on accepting conditional probability as a primitive notion rather than as something logically deduced.

and

$$P(B | A) = \frac{P(A \cap B)}{P(A)} \quad (\text{provided } P(A) \neq 0).$$

Solving the second equation for  $P(A \cap B)$  and substituting into the first yields the desired result. ■

Bayesian inference allows us to update our degree of belief in the occurrence of an event  $A$ , initially represented by the prior probability  $P(A)$ , in light of new evidence, represented by the occurrence of another event  $B$ . The result is the posterior probability  $P(A | B)$ , which reflects our revised belief after considering the new information.

■ **Example B.5** Let  $E$  represent a rare disease that affects one in every million people, so that  $P(E) = 1 \times 10^{-6}$ . Let  $+$  denote a diagnostic test designed to detect the disease, with a failure rate of one in a thousand tests; that is,  $P(+ | E) = 999/1000$ . We wish to compute the probability that a person has the disease given a positive test result, i.e.,  $P(E | +)$ . Applying Bayes' theorem, we get:

$$P(E | +) = \frac{P(+ | E)P(E)}{P(+ | E)P(E) + P(+ | E^c)P(E^c)} = \frac{P(+ | E)P(E)}{P(+ | E)P(E) + (1 - P(+ | E))P(E^c)} = \frac{0.999 \times 10^{-6}}{0.999 \times 10^{-6} + (1 - 0.999) \times 10^{-6}} = 0.001$$

This result shows that, despite the test failing only once in a thousand applications, the probability of actually having the disease after a positive result remains extremely low. This seemingly paradoxical outcome arises because the probability of a false positive ( $10^{-3}$ ) is much higher than the prior probability of having the disease ( $10^{-6}$ ).

In practical settings, this problem is often addressed by administering a second, independent test to individuals who test positive. Under the assumption of test independence, the probability of disease given two positive results increases substantially, to approximately 0.5. ■

Bayes' theorem is particularly useful when the events involved are dependent, allowing new information about one event to revise our assessment of the probability of another.

■ **Example B.6** Suppose you draw a single card from a standard 52-card deck. Let event  $A$  be "drawing a red card," and let event  $B$  be "drawing a queen." In this case, applying Bayes' theorem to compute  $P(A | B)$ , the probability of having drawn a red card given that a queen has been drawn, yields little insight. This is because knowing that a queen was drawn does not significantly alter the probability of the card being red: two of the four queens are red and two are black, mirroring the overall color distribution in

the deck. Thus, event  $B$  provides no relevant new information about event  $\$A\$$ . ■

Bayes' theorem can be generalized to accommodate multiple events. Let  $A_1, \dots, A_k$  be a finite collection of events with  $P(A_j) > 0$  for all  $j = 1, \dots, k$ , and assume that these events form a partition of the sample space  $\Omega$ . Let  $B$  be an event such that  $P(B) > 0$ . Then, for each  $i = 1, \dots, k$ , the conditional probability  $P(A_i | B)$  is given by:

$$P(A_i | B) = \frac{P(B | A_i) P(A_i)}{\sum_{j=1}^k P(B | A_j) P(A_j)}$$

This expression demonstrates the applicability of Bayes' theorem to a wider range of inference problems involving multiple events.

## B.4 Random Variables

A random variable<sup>5</sup> is a function that assigns a real number to each possible outcome of an experiment, thus providing a quantitative representation of the results. Although the Kolmogorov framework assigns probabilities to events, random variables offer a more direct approach by associating numerical values with outcomes. This not only simplifies the representation of probabilistic phenomena but also facilitates the analysis of their properties. Owing to their effectiveness, most statisticians conduct their investigations primarily within the framework of random variables rather than working directly with probability spaces.

**Definition B.4.1** Let  $(\Omega, \mathcal{A}, P)$  be a discrete probability space. A *random variable* is a function  $X : \Omega \rightarrow \mathbb{R}$  mapping from the set of the sample space  $\Omega$  to the real numbers  $\mathbb{R}$ . A random variable is discrete if its range  $\{x_1, x_2, \dots, x_i, \dots\}$  is finite or countably infinite.

Our focus here is on discrete random variables defined on discrete probability spaces. Although it is possible to define a discrete random variable on a non-discrete probability space by restricting it to a discrete range, this case is not considered in this book.

Random variables are particularly useful when they represent specific characteristics of the experiment. For example, if the sample space consists of a school's student body, a random variable could assign to each student their respective height. Random variables also allow us to transform the

<sup>5</sup>The term "random variable" can be somewhat misleading. First, a random variable is not a variable in the usual algebraic sense, but rather a function. Second, it is not the random variable itself that is random; the randomness lies in the experiment it represents. Despite this potential for confusion, we follow the standard terminology.

original outcomes of the sample space into new events. For instance, if two dice are rolled, a random variable could represent the sum of the two outcomes.

It is important to note that we are free to define a random variable on any sample space, even if the assignment does not appear intuitively meaningful. For example, one might assign a numerical value to each color in a deck of cards, draw two cards at random, and take the sum of the assigned numbers. While such a construction may have little practical interpretation, it nonetheless defines a valid random variable, and probabilities can be computed accordingly.

**Definition B.4.2** Let  $X : \Omega \rightarrow \mathbb{R}$  be a discrete random variable, and let  $C \subset \mathbb{R}$  be a set such that  $\{\omega \in \Omega : X(\omega) \in C\}$  is an event. The probability of  $X$  taking a value in  $C$ , denoted  $P(X \in C)$ , is defined by  $P(X \in C) = P(\{\omega \in \Omega : X(\omega) \in C\})$ .

The probability associated with a random variable  $X$  effectively defines a probability space on the real line, specifically over the range of  $X$ .

■ **Example B.7** Let  $\Omega = \{1, 2, 3, 4, 5, 6\}$  be the sample space for a single toss of a fair die, and  $P$  a probability that assigns  $1/6$  to each single outcome in  $\Omega$ . Let  $X : \Omega \rightarrow \mathbb{R}$  be a discrete random variable defined as:

$$X(\omega) = \begin{cases} 0 & \text{if } \omega \text{ is even (2, 4, 6),} \\ 1 & \text{if } \omega \text{ is odd (1, 3, 5).} \end{cases}$$

This random variable maps each outcome of the die toss to either 0 (if the result is even) or 1 (if the result is odd). For  $C = \{0\}$ , the probability  $P(X \in C) = P(X = 0) = P(\{2, 4, 6\}) = 1/2$ . For  $C = \{1\}$  the probability  $P(X \in C) = P(X = 1) = P(\{1, 3, 5\}) = 1/2$ . Through this transformation, the original discrete probability space has been mapped to the real numbers using the discrete random variable  $X$ , establishing a new probability over  $X$ 's range. ■

Unless stated otherwise, we will assume that  $\{\omega \in \Omega, : X(\omega) = x_i\}$  is an event for every point  $x_i$  in the range of  $X$ . The following example shows a case where this assumption does not hold.

■ **Example B.8** Let  $(\Omega, \mathcal{A}, P)$  be a discrete probability space, where  $\Omega = \{1, 2\}$ ,  $\mathcal{A} = \{\emptyset, \Omega\}$ ,  $P(\emptyset) = 0$  and  $P(\Omega) = 1$ . Let  $X : \Omega \rightarrow \mathbb{R}$  be a discrete random variable defined as  $X(1) = 1$  and  $X(2) = 2$ . We are interested in the probability  $P(X = 1)$ , however since  $\{\omega \in \Omega : X(\omega) = 1\} = \{1\}$  is not an event (because  $\{1\} \notin \mathcal{A}$ ), such probability cannot be determined. ■

Definition B.2.3 introduced the concept of probability mass function for

probability spaces, based on the probabilities of the events. The following definition extends this concept to random variables.

**Definition B.4.3** Let  $X$  be a discrete random variable over a discrete probability space, and let  $\{x_1, x_2, \dots\}$  be the range of  $X$ . The *probability mass function* of the discrete random variable  $X$ , abbreviated as p.m.f., is the function  $f : \text{range}(X) \rightarrow [0, 1]$  defined by  $f(x_i) = P(X = x_i)$ .

The set of values for which the probability mass function is positive,  $\{x : f(x) > 0\}$ , is called the *support* of the distribution of  $X$ .

It is possible for two random variables to share the same probability mass function while differing in other significant respects.

■ **Example B.9** Let  $\Omega = \{H, T\}$  be the sample space for a single toss of a fair coin, and let  $P$  be a probability that assigns  $1/2$  to each outcome in  $\Omega$ . Define a discrete random variable  $X : \Omega \rightarrow \mathbb{R}$  by setting  $X = 1$  if the coin shows Head and  $X = 0$  if coin shows Tail. The distribution of  $X$  is given by  $P(X = 0) = P(X = 1) = 0.5$ . This distribution is identical to that of the random variable in Example B.7, even though the two random variables are defined on different sample spaces and represent different experiments. ■

Given the probability mass function of a random variable, we can determine the probability of any subset of the real line.

**Proposition B.4.1** Let  $X$  be a discrete random variable with probability mass function  $f$ . For any subset  $C$  of the real line, the probability  $P(X \in C)$  is given by  $\sum_{x_i \in C} f(x_i)$ .

*Proof.* Since each outcome in the sample space corresponds to exactly one value in the range  $x_1, x_2, \dots$  of  $X$ , we have

$$P(X \in C) = P(\{\omega \in \Omega : X(\omega) \in C\}) = \sum_{x_i \in C} P(X = x_i) = \sum_{x_i \in C} f(x_i).$$

■

This result shows that the probability mass function completely determines the probability distribution of a discrete random variable, since the probability of any event involving  $X$  can be computed from it.

The following proposition states a fundamental property: the total probability over all possible values of a random variable is equal to 1.

**Proposition B.4.2** Let  $X$  be a discrete random variable with probability mass function  $f$ . If  $\{x_1, x_2, \dots\}$  is the range of  $X$ , then  $\sum_{i=1}^{\infty} f(x_i) = 1$ .

*Proof.* Since  $X$  is a total function, each outcome in the sample space is associated with exactly one value in the range  $\{x_1, x_2, \dots\}$ . By the axioms of

probability,

$$\sum_{i=1}^{\infty} f(x_i) = \sum_{i=1}^{\infty} P(X = x_i) = \sum_{i=1}^{\infty} P(\{\omega \in \Omega : X(\omega) = x_i\}) = P(\Omega) = 1.$$

■

The values  $\{x_1, x_2, \dots\}$  represent all the possible outcomes of  $X$ . Since these outcomes are mutually exclusive and collectively exhaustive, their probabilities must sum to the total probability of the entire sample space, which is 1.

A fundamental characterization of discrete distributions is that they are completely determined by a nonnegative function on the real line whose total mass is 1. The next proposition states this precisely.

**Proposition B.4.3** A function  $f(x)$  is a probability mass function of a random variable  $X$  if, and only if, i)  $f(x) \geq 0$  for all  $x$ , ii)  $\sum_{i=1}^{\infty} f(x_i) = 1$ .

*Proof.* Suppose  $f$  is the probability mass function of a discrete random variable  $X$  with range  $\{x_1, x_2, \dots\}$ . Then  $f(x_i) = P(X = x_i) \geq 0$  for all  $i$ , and

$$\sum_{i=1}^{\infty} f(x_i) = \sum_{i=1}^{\infty} P(X = x_i) = P(\Omega) = 1,$$

so (i)-(ii) hold.

Conversely, assume  $f(x) \geq 0$  for all  $x$  and  $\sum_{i=1}^{\infty} f(x_i) = 1$  for some countable set  $\{x_1, x_2, \dots\}$  (the points where  $f$  may be positive). Define a discrete sample space  $\Omega = \{x_1, x_2, \dots\}$ , set  $P(x_i) = f(x_i)$ , and define  $X : \Omega \rightarrow \mathbb{R}$  by  $X(\omega) = \omega$ . Then for each  $i$ ,

$$P(X = x_i) = P(\{\omega \in \Omega : X(\omega) = x_i\}) = P(\{x_i\}) = f(x_i),$$

so  $f$  is the probability mass function of  $X$ . ■

The proposition shows that a discrete distribution is fully specified by any function that is everywhere nonnegative and sums to 1 over its (countable) support; values of  $f$  outside the support are irrelevant.

The cumulative distribution function expresses the probability that a random variable takes a value less than or equal to a given point.

**Definition B.4.4** The *cumulative distribution function* (c.d.f.)  $F$  of a discrete random variable  $X$  is defined by  $F(x) = P(X \leq x)$  for all  $-\infty < x < \infty$

If  $X$  has a distribution characterized by the probability mass function  $f(x)$ , its cumulative distribution function  $F(x)$  behaves as follows: at each distinct value  $x_i$  in the range of  $X$ ,  $F(x)$  increases by  $f(x_i)$ ; between these values,  $F(x)$  remains constant.

The cumulative distribution function shows how probabilities accumulate over the range of a random variable, providing insight into the overall distribution of the data. It can be viewed as the running total of the probabilities given by the probability mass function.

■ **Example B.10** Let  $X$  be a discrete random variable representing the grades of students in a class, where each grade is between 0 and 10. The probability that a student receives a grade of  $x$  is given by the probability mass function  $p(x)$ . The cumulative distribution function  $F(x)$  gives the probability that a randomly selected student scores  $x$  or less. For example, if  $F(7) = 0.6$ , this means there is a 60% chance that a randomly chosen student scored 7 or below. ■

The cumulative distribution function of a random variable is always non-decreasing.

**Proposition B.4.4** Let  $F$  be the cumulative distribution function of a discrete random variable  $X$ . If  $x_1 < x_2$ , then  $F(x_1) \leq F(x_2)$ .

*Proof.* For  $x_1 < x_2$ , the event  $X \leq x_1$  is a subset of the event  $X \leq x_2$ . By the monotonicity property of probability,  $P(X \leq x_1) \leq P(X \leq x_2)$ . Since  $F(x) = P(X \leq x)$ , it follows that  $F(x_1) \leq F(x_2)$ . ■

The following proposition states the asymptotic properties of the cumulative distribution function of a random variable, showing its limits as  $x$  approaches negative and positive infinity.

**Proposition B.4.5** Let  $F$  be the cumulative distribution function of a discrete random variable  $X$ . Then  $\lim_{x \rightarrow -\infty} F(x) = 0$  and  $\lim_{x \rightarrow \infty} F(x) = 1$ .

*Proof.* As  $x \rightarrow -\infty$ , the event  $X \leq x$  eventually becomes empty, because there are no values in the range of  $X$  smaller than sufficiently negative  $x$ . The probability of this event is therefore 0, so

$$\lim_{x \rightarrow -\infty} F(x) = \lim_{x \rightarrow -\infty} P(X \leq x) = 0.$$

As  $x \rightarrow \infty$ , the event  $X \leq x$  eventually contains the entire sample space  $\Omega$ , since all possible values of  $X$  are less than or equal to sufficiently large  $x$ . The probability of this event is therefore 1, so

$$\lim_{x \rightarrow \infty} F(x) = \lim_{x \rightarrow \infty} P(X \leq x) = 1.$$



The probability that  $X$  exceeds  $x$  is the complement of the cumulative distribution function at that point.

**Proposition B.4.6** Let  $F$  be the cumulative distribution function of a discrete random variable  $X$ . Then, for every  $x \in \mathbb{R}$ ,  $P(X > x) = 1 - F(x)$ .

*Proof.* The events  $X > x$  and  $X \leq x$  are complementary, so their probabilities sum to 1  $P(X > x) + P(X \leq x) = 1$ . Therefore,  $P(X > x) = 1 - P(X \leq x)$ . By the definition of the cumulative distribution function,  $P(X \leq x) = F(x)$ , and thus  $P(X > x) = 1 - F(x)$ . ■

The following proposition relates the probability of a random variable  $X$  taking a value between two points to the difference in its cumulative distribution function at those points.

**Proposition B.4.7** Let  $F$  be the cumulative distribution function of a discrete random variable  $X$ . Then, for all  $x_1, x_2 \in \mathbb{R}$  with  $x_1 < x_2$ ,  $P(x_1 < X \leq x_2) = F(x_2) - F(x_1)$ .

*Proof.* The probability that  $X \leq x_2$  is  $F(x_2)$ . Subtracting from this the probability that  $X \leq x_1$ , which is  $F(x_1)$ , leaves exactly the probability that  $X$  falls strictly between  $x_1$  and  $x_2$ :

$$P(x_1 < X \leq x_2) = F(x_2) - F(x_1).$$



## B.4.1 Multivariate Distributions

A multivariate probability distribution extends the concept of a probability distribution to multiple random variables, each with its own set of possible outcomes. While univariate distributions describe phenomena involving a single random variable, multivariate distributions capture the relationships and dependencies between two or more variables. This allows for the analysis of complex phenomena in which the outcome of interest is influenced by several factors simultaneously, providing insight into how these variables interact and affect the probabilities of different outcomes.

**Definition B.4.5** Let  $X_1, X_2, \dots, X_n$  be  $n$  discrete random variables on a common discrete probability space  $(\Omega, \mathcal{A}, P)$ , where  $X_i : \Omega \rightarrow \mathbb{R}$  for  $i = 1, \dots, n$ . The *joint probability distribution* of  $X_1, \dots, X_n$  assigns probabilities to events involving all variables simultaneously. Formally, for any

set  $C \subset \mathbb{R}^n$ , the joint probability is

$$P((X_1, X_2, \dots, X_n) \in C),$$

where

$$\{\omega \in \Omega : (X_1(\omega), X_2(\omega), \dots, X_n(\omega)) \in C\}$$

is an event in the sample space.

The joint probability distribution of the random variables  $X_1, X_2, \dots, X_n$  defines a probability distribution on  $\mathbb{R}^n$ . If the random variables  $X_1, X_2, \dots, X_n$  each have a discrete distribution, then the joint distribution is also a discrete distribution.

**Definition B.4.6** Let  $X_1, \dots, X_n$  be  $n$  discrete random variables over a common discrete probability space. The *joint probability mass function* of  $X_1, \dots, X_n$  is the function

$$f : \text{range}(X_1) \times \dots \times \text{range}(X_n) \rightarrow [0, 1]$$

defined by

$$f(x_1, \dots, x_n) = P(X_1 = x_1, \dots, X_n = x_n).$$

■ **Example B.11** A classic example of a bivariate discrete joint distribution involves rolling two six-sided dice. Define two discrete random variables:  $X_1$  is the outcome of the first die, and  $X_2$  is the outcome of the second die. Both  $X_1$  and  $X_2$  follow a discrete uniform distribution over 1, 2, 3, 4, 5, 6. The joint distribution of  $X_1$  and  $X_2$  describes the probability of each possible pair of outcomes when the dice are rolled. The joint probability mass function  $f(x_1, x_2)$  is:

$$f(x_1, x_2) = P(X_1 = x_1, X_2 = x_2) = \frac{1}{36}, \quad x_1, x_2 \in \{1, 2, 3, 4, 5, 6\}.$$

The joint distribution allows us to answer questions such as the probability that the sum of the two dice equals a certain value, or that one die shows a higher number than the other. ■

Before exploring the properties of multivariate random variables, we introduce the concept of a random vector. This simplifies notation and improves clarity by grouping multiple random variables into a single object.

**Definition B.4.7** A *discrete random vector  $\mathbf{X}$*  is an ordered collection of  $n$  discrete random variables  $X_1, X_2, \dots, X_n$ , where  $X_i : \Omega \rightarrow \mathbb{R}$  for all

$i = 1, \dots, n$ . Equivalently,  $\mathbf{X} = (X_1, \dots, X_n) : \Omega \rightarrow \mathbb{R}^n$ .

Given the joint probability mass function of a random vector, we can determine the probability of any subset of the  $n$ -dimensional real space.

**Proposition B.4.8** Let  $\mathbf{X} = (X_1, \dots, X_n)$  be a discrete random vector with joint probability mass function  $f$ . For any subset  $C$  of  $\mathbb{R}^n$ ,

$$P(\mathbf{X} \in C) = \sum_{\mathbf{x} \in C} f(\mathbf{x}),$$

where  $\mathbf{x} = (x_1, \dots, x_n)$ .

*Proof.* Since each outcome in the sample space corresponds to exactly one value in the range of  $\mathbf{X}$ , we have:

$$P(\mathbf{X} \in C) = P(\{\omega \in \Omega : \mathbf{X}(\omega) \in C\}) = \sum_{\mathbf{x} \in C} P(\mathbf{X} = \mathbf{x}) = \sum_{\mathbf{x} \in C} f(\mathbf{x}).$$

■

The following proposition states a fundamental property, that the total probability over all possible outcomes of a discrete random vector is 1.

**Proposition B.4.9** Let  $\mathbf{X} = (X_1, \dots, X_n)$  be a discrete random vector with a joint probability mass function  $f$ . If the range of  $\mathbf{X}$  is  $\mathbf{x} = (x_1, \dots, x_n)$ , then  $\sum_{\mathbf{x}} f(\mathbf{x}) = 1$ .

*Proof.* Since  $\mathbf{X}$  is a total function and each outcome in the sample space corresponds to exactly one value in the range of  $\mathbf{X}$ , the axioms of probability imply:

$$\sum_{\mathbf{x}} f(\mathbf{x}) = \sum_{\mathbf{x}} P(\mathbf{X} = \mathbf{x}) = P(\Omega) = 1.$$

■

A particularly important multivariate construction is the distribution of the sum of  $n$  random variables. The notation  $X_1 + \dots + X_n$  refers to the sum of the random variables (applied pointwise on outcomes), not a sum of  $n$  probability distributions. There are two standard ways to formalize this, depending on how the random variables are defined.

The first one refers to a sum on a common sample space.

**Definition B.4.8** Let  $X_1, X_2, \dots, X_n$  be discrete random variables on a common probability space  $(\Omega, \mathcal{A}, P)$ , with  $X_i : \Omega \rightarrow \mathbb{R}$ . The sum is the discrete random variable

$$S = X_1 + \dots + X_n : \Omega \rightarrow \mathbb{R}, \quad S(\omega) = X_1(\omega) + \dots + X_n(\omega).$$

The *distribution of the sum* is the distribution of  $S$ .

The second one refers to a sum on product space (independent relations).

**Definition B.4.9** Let  $X : \Omega \rightarrow \mathbb{R}$  model a single trial. To model  $n$  repetitions, consider the product sample space  $\Omega^n$  (each  $n$ -tuple of outcomes is a sample point, and in the i.i.d. case all  $n$ -tuples are equally likely when  $\Omega$  is finite). Define

$$X_i(\omega_1, \dots, \omega_n) = X(\omega_i), \quad i = 1, \dots, n,$$

and the sum

$$S(\omega_1, \dots, \omega_n) = X(\omega_1) + \dots + X(\omega_n), \quad S : \Omega^n \rightarrow \mathbb{R}.$$

Again, the *distribution of the sum* is the distribution of  $S$ .

In both setups, the distribution of  $S$  depends on the joint distribution of  $(X_1, \dots, X_n)$ . Independence is not assumed unless explicitly stated.

■ **Example B.12** Let  $X : \Omega \rightarrow \mathbb{R}$  be the outcome of rolling a six-sided die (values  $1, \dots, 6$ , each with probability  $1/6$ ). For three rolls, use the product space  $\Omega^3$  with each triple equally likely. Define

$$X_1(\omega_1, \omega_2, \omega_3) = X(\omega_1), X_2(\omega_1, \omega_2, \omega_3) = X(\omega_2), X_3(\omega_1, \omega_2, \omega_3) = X(\omega_3),$$

and

$$S(\omega_1, \omega_2, \omega_3) = X(\omega_1) + X(\omega_2) + X(\omega_3).$$

Then  $S : \Omega^3 \rightarrow \mathbb{R}$  is the sum of three (independent) die rolls. ■

### B.4.2 Marginal Probability Mass Function

Given a multivariate discrete probability mass function, the marginal probability mass function of a subset of variables is derived by summing the joint probability mass function over all possible values of the remaining variables. This process "marginalizes" out the variables not of interest, allowing us focus on the probability mass function of a single variable or a subset of variables within the multivariate context.

**Definition B.4.10** Let  $\mathbf{X} = (X_1, X_2, \dots, X_n)$  be an  $n$ -dimensional discrete random vector with joint probability mass function  $f_{\mathbf{X}}(x_1, x_2, \dots, x_n)$ , partition  $\mathbf{X}$  into two subvectors:  $\mathbf{Y} = (Y_1, Y_2, \dots, Y_k)$ , a  $k$ -dimensional random vector consisting of  $k$  discrete random variables selected from  $\mathbf{X}$ , and  $\mathbf{Z} = (Z_1, Z_2, \dots, Z_{n-k})$ , the remaining  $(n - k)$  discrete random variables of  $\mathbf{X}$ . The *marginal probability mass function*  $f_{\mathbf{Y}}$  of  $\mathbf{Y}$  is obtained

by summing  $f_{\mathbf{X}}$  over all possible values of the variables in  $\mathbf{Z}$ . That is, for any specific values  $(y_1, y_2, \dots, y_k)$  of  $\mathbf{Y}$ ,

$$f_{\mathbf{Y}}(y_1, y_2, \dots, y_k) = \sum_{z_1} \sum_{z_2} \dots \sum_{z_{n-k}} f_{\mathbf{X}}(x_1, x_2, \dots, x_n),$$

where in each term of the sum,  $x_i = y_i$  for  $i$  corresponding to variables in  $\mathbf{Y}$ , and  $x_j = z_j$  for  $j$  corresponding to variables in  $\mathbf{Z}$ .

This definition captures the marginalization process in the discrete setting, which is key to understanding and analyzing the behavior of specific variables within a larger multivariate framework.

■ **Example B.13** Consider two discrete random variables  $X$  and  $Y$ , each taking values in 0, 1, with joint probability mass function:

$X \setminus Y$	0	1
0	0.1	0.3
1	0.2	0.4

To find the marginal probability mass function of  $X$ , sum over all possible values of  $Y$ :

$$f_X(0) = f_{X,Y}(0,0) + f_{X,Y}(0,1) = 0.1 + 0.3 = 0.4,$$

$$f_X(1) = f_{X,Y}(1,0) + f_{X,Y}(1,1) = 0.2 + 0.4 = 0.6.$$

While the marginal probability mass functions of  $X_1, \dots, X_n$  can be obtained from their joint probability mass function by summing over the range of the other variables, the reverse process is not possible without additional information about the dependence structure among the variables. Marginal probability mass functions describe only the individual behavior of each variable and do not capture interactions between them.

■ **Example B.14** Let  $X$  and  $Y$  be discrete random variables, each taking values in 0, 1, with marginal probability mass functions:

$$f_X(0) = 0.5, \quad f_X(1) = 0.5,$$

$$f_Y(0) = 0.5, \quad f_Y(1) = 0.5.$$

Without additional information about the relationship between  $X$  and  $Y$ , the joint probability mass function cannot be reconstructed from the marginals alone.

A random vector  $\mathbf{X}$  is said to have independent components if knowing the outcome of one component provides no information about the others.

In this case, events defined by different components occur independently, and the joint probability mass function factorizes into the product of the marginals.

**Definition B.4.11** Let  $\mathbf{X} = (X_1, X_2, \dots, X_n)$  be an  $n$ -dimensional discrete random vector with joint probability mass function  $f_{\mathbf{X}}$  and marginal probability mass functions  $f_{X_1}, f_{X_2}, \dots, f_{X_n}$ . The components  $X_1, \dots, X_n$  are *independent* if, for all  $(x_1, \dots, x_n)$  in the support of  $\mathbf{X}$ ,

$$f_{\mathbf{X}}(x_1, \dots, x_n) = \prod_{i=1}^n f_{X_i}(x_i).$$

The concept of independence for a random vector  $\mathbf{X}$  simplifies the computation and understanding of joint probability distributions, particularly in complex problems involving multiple variables. It allows the joint probability distribution of the vector  $\mathbf{X}$  to be expressed as the product of the individual marginal distributions of  $X_1, X_2, \dots, X_n$ .

**Proposition B.4.10** Let  $\mathbf{X} = (X_1, X_2, \dots, X_n)$  be an  $n$ -dimensional random vector with joint probability mass function  $f_{\mathbf{X}}$  and marginal probability mass functions  $f_{X_1}, f_{X_2}, \dots, f_{X_n}$ . The random variables  $X_1, X_2, \dots, X_n$  are independent if and only if for every  $(x_1, x_2, \dots, x_n)$  in the support of  $\mathbf{X}$ , we have:

$$f_{\mathbf{X}}(x_1, x_2, \dots, x_n) = f_{X_1}(x_1) f_{X_2}(x_2) \dots f_{X_n}(x_n).$$

*Proof.* Assume that  $X_1, X_2, \dots, X_n$  are independent. Then, for any values  $x_1, x_2, \dots, x_n$ , we have:

$$\begin{aligned} f_{\mathbf{X}}(x_1, x_2, \dots, x_n) &= P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) \\ &= P(X_1 = x_1) P(X_2 = x_2) \dots P(X_n = x_n) \\ &= f_{X_1}(x_1) f_{X_2}(x_2) \dots f_{X_n}(x_n). \end{aligned}$$

Conversely, assume that for all  $x_1, x_2, \dots, x_n$ ,

$$f_{\mathbf{X}}(x_1, x_2, \dots, x_n) = f_{X_1}(x_1) f_{X_2}(x_2) \dots f_{X_n}(x_n).$$

Then, for any subsets  $A_1, \dots, A_n$  of  $\mathbb{R}$ , we have:

$$\begin{aligned} P(X_1 \in A_1, \dots, X_n \in A_n) &= \sum_{(x_1, \dots, x_n) \in A_1 \times \dots \times A_n} f_{\mathbf{X}}(x_1, \dots, x_n) \\ &= \sum_{x_1 \in A_1} \dots \sum_{x_n \in A_n} f_{X_1}(x_1) \dots f_{X_n}(x_n) \\ &= \left( \sum_{x_1 \in A_1} f_{X_1}(x_1) \right) \dots \left( \sum_{x_n \in A_n} f_{X_n}(x_n) \right) \\ &= P(X_1 \in A_1) \dots P(X_n \in A_n). \end{aligned}$$

This equality holds for all subsets  $A_1, \dots, A_n$ , which implies that  $X_1, \dots, X_n$  are independent. ■

### B.4.3 Conditional Probability Mass Function

The concept of the conditional probability mass function provides a way to quantify the probability of an event given that another event has occurred. In the case of discrete random variables, the conditional probability mass function of  $Y$  given  $X = x$  describes the probability mass function of  $Y$  under the condition that  $X$  takes a specific value  $x$ . This concept is essential for analyzing dependencies between discrete random variables, allowing us to refine probability assessments based on new information.

**Definition B.4.12** Let  $\mathbf{X} = (X_1, X_2, \dots, X_n)$  be an  $n$ -dimensional random vector with joint probability mass function  $f_{\mathbf{X}}(x_1, x_2, \dots, x_n)$ . Partition  $\mathbf{X}$  into two subvectors:  $\mathbf{Y} = (Y_1, Y_2, \dots, Y_k)$ , a  $k$ -dimensional random vector consisting of  $k$  discrete random variables selected from  $\mathbf{X}$ , and  $\mathbf{Z} = (Z_1, Z_2, \dots, Z_{n-k})$ , the remaining  $n - k$  discrete random variables of  $\mathbf{X}$ . Let  $f_{\mathbf{Z}}$  be the marginal probability mass function of  $\mathbf{Z}$  across its  $n - k$  dimensions. Provided that  $f_{\mathbf{Z}}(\mathbf{z}) > 0$  for any vector  $\mathbf{z} \in \mathbb{R}^{n-k}$ , the *conditional probability mass function*  $f_{\mathbf{Y}|\mathbf{Z}}$  of  $\mathbf{Y}$  given  $\mathbf{Z} = \mathbf{z}$  is defined as:

$$f_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y} \mid \mathbf{z}) = \frac{f_{\mathbf{X}}(\mathbf{y}, \mathbf{z})}{f_{\mathbf{Z}}(\mathbf{z})}.$$

The following example illustrates how to compute the conditional probability mass function of  $Y$  given  $X$ .

■ **Example B.15** Consider two discrete random variables  $X$  and  $Y$ , each taking values in  $\{0, 1\}$ , with the joint probability mass function:

$X \setminus Y$	0	1
0	0.3	0.2
1	0.1	0.4

The conditional probability mass function of  $Y$  given  $X = 0$  is:

$$f_{Y|X}(0 \mid 0) = \frac{f_{X,Y}(0, 0)}{f_X(0)} = \frac{0.3}{0.5} = 0.6,$$

$$f_{Y|X}(1 \mid 0) = \frac{f_{X,Y}(0, 1)}{f_X(0)} = \frac{0.2}{0.5} = 0.4.$$

Similarly, for  $X = 1$ :

$$f_{Y|X}(0 | 1) = \frac{f_{X,Y}(1, 0)}{f_X(1)} = \frac{0.1}{0.5} = 0.2,$$

$$f_{Y|X}(1 | 1) = \frac{f_{X,Y}(1, 1)}{f_X(1)} = \frac{0.4}{0.5} = 0.8.$$

■

The next proposition generalizes the multiplication rule (see Equation B.1) by combining marginal and conditional probability mass functions to recover the joint probability mass function for any configuration of discrete random variables within a random vector.

**Proposition B.4.11** Let  $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\mathbf{Z}$ ,  $f_{\mathbf{X}}(\mathbf{x})$ ,  $f_{\mathbf{Z}}(\mathbf{z})$ , and  $f_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y} | \mathbf{z})$  be as in Definition B.4.12. Then, for each  $\mathbf{z}$  such that  $f_{\mathbf{Z}}(\mathbf{z}) > 0$  and each possible value of  $\mathbf{y}$ , the joint probability mass function is:

$$f_{\mathbf{X}}(\mathbf{x}) = f_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y} | \mathbf{z})f_{\mathbf{Z}}(\mathbf{z}),$$

where  $\mathbf{x} = (\mathbf{y}, \mathbf{z})$ .

*Proof.* By the definition of the conditional probability mass function:

$$f_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y} | \mathbf{z}) = \frac{f_{\mathbf{X}}(\mathbf{y}, \mathbf{z})}{f_{\mathbf{Z}}(\mathbf{z})}, \quad \text{for } f_{\mathbf{Z}}(\mathbf{z}) > 0.$$

Rearranging yields:

$$f_{\mathbf{X}}(\mathbf{y}, \mathbf{z}) = f_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y} | \mathbf{z})f_{\mathbf{Z}}(\mathbf{z}),$$

and since  $\mathbf{x} = (\mathbf{y}, \mathbf{z})$ , the result follows. ■

Similarly, the conditional probability mass function of  $\mathbf{Z}$  given  $\mathbf{Y} = \mathbf{y}$ , denoted  $f_{\mathbf{Z}|\mathbf{Y}}(\mathbf{z} | \mathbf{y})$ , satisfies:

$$f_{\mathbf{X}}(\mathbf{x}) = f_{\mathbf{Z}|\mathbf{Y}}(\mathbf{z} | \mathbf{y})f_{\mathbf{Y}}(\mathbf{y}).$$

Bayes' theorem (see Theorem B.3.2) provides a way to update probability estimates for a hypothesis given new evidence. For random vectors, the theorem extends naturally to the multidimensional case.

**Theorem B.4.12 — Bayes' Theorem for Random Vectors.** Let  $\mathbf{X} = (X_1, X_2, \dots, X_n)$  and  $\mathbf{Y} = (Y_1, Y_2, \dots, Y_m)$  be two random vectors representing distinct sets of discrete random variables. The generalized Bayes' theorem states:

$$P(\mathbf{X} = \mathbf{x} | \mathbf{Y} = \mathbf{y}) = \frac{P(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x})P(\mathbf{X} = \mathbf{x})}{P(\mathbf{Y} = \mathbf{y})},$$

where  $P(\mathbf{Y} = \mathbf{y})$  is the marginal probability of  $\mathbf{Y}$ , given by:

$$P(\mathbf{Y} = \mathbf{y}) = \sum_{\mathbf{x}} P(\mathbf{Y} = \mathbf{y} \mid \mathbf{X} = \mathbf{x}) P(\mathbf{X} = \mathbf{x}).$$

*Proof.* From the definition of conditional probability:

$$P(\mathbf{X} = \mathbf{x} \mid \mathbf{Y} = \mathbf{y}) = \frac{P(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})}{P(\mathbf{Y} = \mathbf{y})}.$$

Since

$$P(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}) = P(\mathbf{Y} = \mathbf{y} \mid \mathbf{X} = \mathbf{x}) P(\mathbf{X} = \mathbf{x}),$$

substitution gives the stated result. ■

This generalized Bayes' theorem updates beliefs about  $\mathbf{X}$  based on new evidence from  $\mathbf{Y}$ . It links the prior information we have about  $\mathbf{X}$ , the likelihood of observing  $\mathbf{Y} = \mathbf{y}$  given  $\mathbf{X} = \mathbf{x}$ , and the evidence provided by the actual observation of  $\mathbf{Y} = \mathbf{y}$ .

Finally, building on the concept of independence between discrete random variables (see Definition B.4.11), we define conditional independence. This concept comes into play when the independence of a set of discrete random variables is considered in the context of being conditioned on another set of variables.

**Definition B.4.13** Let  $\mathbf{Z}$  be a random vector with joint probability mass function  $f_{\mathbf{Z}}(\mathbf{z})$ . The variables of the random vector  $\mathbf{X} = (X_1, \dots, X_n)$  are *conditionally independent* given  $\mathbf{Z}$  if, for all  $\mathbf{z}$  such that  $f_{\mathbf{Z}}(\mathbf{z}) > 0$ :

$$f_{\mathbf{X}|\mathbf{Z}}(\mathbf{x} \mid \mathbf{z}) = \prod_{i=1}^n f_{X_i|\mathbf{Z}}(x_i \mid \mathbf{z}),$$

where  $f_{\mathbf{X}|\mathbf{Z}}(\mathbf{x} \mid \mathbf{z})$  is the conditional probability mass function of  $\mathbf{X}$  given  $\mathbf{Z} = \mathbf{z}$ , and  $f_{X_i|\mathbf{Z}}(x_i \mid \mathbf{z})$  is the conditional probability mass function of  $X_i$  given  $\mathbf{Z} = \mathbf{z}$ .

## B.5 Characterizing Distributions

A *measure of central tendency* is a number derived from a probability distribution that summarizes its typical or central value. The two most commonly used measures are the *expected value* and the *median*, each offering a different perspective for characterizing a distribution.

In addition to central tendency, it is often useful to describe how much the distribution varies around its center. For this purpose, *measures of*

*dispersion* are employed, the most common being the *variance* and the *standard deviation*. These quantify the spread of a distribution around its central measure.

In the case of bivariate distributions, analogous measures of dispersion are the *covariance* and the *correlation*, which capture the strength and direction of the statistical relationship between two discrete random variables.

Together, these measures—central tendency and dispersion—allow us to summarize, compare, and analyze probability distributions in a concise and meaningful way.

### B.5.1 Measures of Central Tendency

The most common measures of central tendency used to characterize probability distributions are the expected value and the median.

#### Expected Value

The expected value of a discrete random variable is computed as the weighted average of all its possible values, where the weights are the probabilities of the corresponding outcomes<sup>6</sup>.

**Definition B.5.1** Let  $X$  be a discrete random variable with probability mass function  $f$ . The *expected value* of  $X$ , denoted by  $E(X)$ , is defined as:

$$E(X) = \sum_x x f(x).$$

This definition depends only on the distribution of  $X$ , not on the original sample space. Consequently, two different random variables with the same distribution will have the same expected value, even if their underlying probability spaces are different.

A drawback of the expected value is that it can be heavily influenced by small changes in the probability assigned to large values of  $X$ .

■ **Example B.16** Consider a company with 100 employees, and define a discrete random variable  $X$  based on their salaries. Suppose  $X$  takes the value 300 with probability 99/100 and 6000 with probability 1/100. The expected salary is:

$$E(X) = (300 \times 99/100) + (6000 \times 1/100) = 357.$$

<sup>6</sup>The term “expected value” can be misleading, because it does not necessarily coincide with one of the possible values of the random variable. For example, the expected value of a fair six-sided die is 3.5, which is not itself an attainable outcome. This counterintuitive aspect has often caused confusion in applications of probability.

Now suppose that one additional employee earns 6000 instead of 300. The expected salary becomes:

$$E(X) = (300 \times 98/100) + (6000 \times 2/100) = 414.$$

Thus, changing the salary of a single employee increases the expected salary of the company by more than 13%. ■

The expected value operator is linear: the expected value of a linear combination of discrete random variables equals the same linear combination of their expectations.

**Proposition B.5.1** Let  $X_1, \dots, X_n$  be discrete random variables with expectations  $E(X_i)$ . For constants  $a_1, \dots, a_n$  and  $b$ ,

$$E(a_1X_1 + \dots + a_nX_n + b) = a_1E(X_1) + \dots + a_nE(X_n) + b.$$

*Proof.* By definition,

$$\begin{aligned} E(a_1X_1 + \dots + a_nX_n + b) &= \sum_{x_1, \dots, x_n} (a_1x_1 + \dots + a_nx_n + b)f(x_1, \dots, x_n) = \\ a_1 \sum_{x_1} x_1 f(x_1) + \dots + a_n \sum_{x_n} x_n f(x_n) + b &= a_1E(X_1) + \dots + a_nE(X_n) + b. \end{aligned}$$

■

Note that independence is not required for linearity of expectation. Linearity holds for all discrete random variables as long as the expectations exist.

The expected value of the product of independent random variables equals the product of their expected values.

**Proposition B.5.2** Let  $X_1, \dots, X_n$  be independent discrete random variables with expectations  $E(X_i)$ . Then:

$$E\left(\prod_{i=1}^n X_i\right) = \prod_{i=1}^n E(X_i).$$

*Proof.* By independence of  $X_1, \dots, X_n$ , their joint probability mass function factorizes:

$$f(x_1, \dots, x_n) = f_{X_1}(x_1) \cdots f_{X_n}(x_n).$$

Thus,

$$\begin{aligned} E(X_1 \cdots X_n) &= \sum_{x_1, \dots, x_n} (x_1 \cdots x_n) f(x_1, \dots, x_n) \\ &= \prod_{i=1}^n \left( \sum_{x_i} x_i f_{X_i}(x_i) \right) = \prod_{i=1}^n E(X_i). \end{aligned}$$



If the random variables are not independent, the equality above does not necessarily hold.

### The Median

The median of a discrete random variable is a measure of central tendency that identifies a point dividing the probability distribution so that at least half of the probability mass lies on each side.

**Definition B.5.2** Let  $X$  be a discrete random variable. A *median* of  $X$ , denoted by  $m$ , is any value satisfying:

$$P(X \leq m) \geq \frac{1}{2} \quad \text{and} \quad P(X \geq m) \geq \frac{1}{2}.$$

This definition guarantees that at least half of the probability mass lies at or below  $m$ , and at least half lies at or above  $m$ . In general, the median of a discrete distribution need not be unique; any value satisfying the inequalities is considered a valid median. In applications, it is common to select the smallest such  $m$ .

The median is often more intuitive than the expected value, since it always corresponds to an actual value of the random variable. It is also a robust measure of central tendency, particularly when the distribution is skewed or contains outliers.

■ **Example B.17** Consider the company from Example B.16. The median of the salaries is the value  $m$  for which  $P(X \leq m) \geq \frac{1}{2}$ . In this case,  $m = 300$ . Now suppose that one of the base employees is promoted to the executive level with a salary of 6000. The recalculated median remains  $m = 300$ . This illustrates that the median is more robust than the expected value in the presence of outliers. ■

## B.5.2 Measures of Dispersion

The most common measures of dispersion in use to characterize probability distributions are the variance and its squared root, called standard deviation.

### The Variance

The variance of a discrete random variable is a measure of the spread or dispersion of its possible values around the expected value.

**Definition B.5.3** Let  $X$  be a discrete random variable with expected value  $E(X)$  and probability mass function  $f$ . The *variance* of  $X$ , denoted by

$Var(X)$ , is defined as:

$$Var(X) = E[(X - E(X))^2] = \sum_x (x - E(X))^2 f(x)$$

Variance depends only on the distribution of the random variable and quantifies how much the values typically differ from the expected value. For example, if all possible values of a random variable are identical, the variance is zero.

■ **Example B.18** Suppose the daily commuting time (in minutes) of an employee is modeled by a random variable  $X$  that takes values 20, 30, 40, 50 with probabilities

$$P(X = 20) = 0.1, P(X = 30) = 0.4, P(X = 40) = 0.4, P(X = 50) = 0.1.$$

The expected value is

$$E(X) = 20(0.1) + 30(0.4) + 40(0.4) + 50(0.1) = 35.$$

The variance is

$$Var(X) = (20 - 35)^2(0.1) + (30 - 35)^2(0.4) + (40 - 35)^2(0.4) + (50 - 35)^2(0.1) = 50.$$

Thus, the variance of commuting times is 50 (measured in squared minutes). ■

The following proposition shows how the variance behaves under linear combinations of independent random variables.

**Proposition B.5.3** Let  $X_1, \dots, X_n$  be independent discrete random variables with finite expected values, and let  $a_1, \dots, a_n$  and  $b$  be constants. Then:

$$Var(a_1X_1 + \dots + a_nX_n + b) = a_1^2Var(X_1) + \dots + a_n^2Var(X_n).$$

*Proof.* Let  $Y = a_1X_1 + \dots + a_nX_n + b$ . Since adding a constant does not affect variance,

$$Var(Y) = Var(a_1X_1 + \dots + a_nX_n).$$

For independent random variables, variance is additive:

$$Var(a_1X_1 + \dots + a_nX_n) = Var(a_1X_1) + \dots + Var(a_nX_n).$$

Finally, for any constant  $a_i$ ,  $Var(a_iX_i) = a_i^2Var(X_i)$ , so

$$Var(a_1X_1 + \dots + a_nX_n + b) = a_1^2Var(X_1) + \dots + a_n^2Var(X_n).$$



A particularly useful identity expresses variance in terms of expected values. This formulation often simplifies calculations and highlights the connection between variance and variability.

**Proposition B.5.4** Let  $X$  be a discrete random variable with expected value  $E(X)$ . Then:

$$\text{Var}(X) = E(X^2) - [E(X)]^2.$$

*Proof.* By definition,

$$\text{Var}(X) = E[(X - E(X))^2].$$

Expanding the square gives:

$$\text{Var}(X) = E[X^2 - 2XE(X) + (E(X))^2].$$

By linearity of expectation,

$$\text{Var}(X) = E(X^2) - 2E(X)E(X) + E((E(X))^2).$$

Since  $E(X)$  is a constant,  $E((E(X))^2) = (E(X))^2$ , hence

$$\text{Var}(X) = E(X^2) - (E(X))^2.$$

■

### Standard Deviation

The standard deviation is a measure of the dispersion of a distribution, closely related to the variance. Unlike variance, which squares the deviations from the mean and therefore uses squared units, the standard deviation is defined as the square root of the variance. This restores the measure to the same units as the original variable, making it easier to interpret.

**Definition B.5.4** Let  $X$  be a discrete random variable with expected value  $E(X)$  and finite variance  $\text{Var}(X)$ , and let  $f$  be its probability mass function. The *standard deviation* of  $X$ , denoted by  $\sigma$ , is defined as:

$$\sigma = \sqrt{\text{Var}(X)} = \sqrt{\sum_x (x - E(X))^2 f(x)}.$$

The standard deviation summarizes how far the values of  $X$  typically lie from the mean. A smaller standard deviation indicates that the values are concentrated near the mean, while a larger standard deviation indicates greater spread.

The following proposition describes how the standard deviation transforms under linear changes of a random variable.

**Proposition B.5.5** Let  $X$  be a discrete random variable with standard deviation  $\sigma_X$ , and let  $a$  and  $b$  be constants. Then the standard deviation  $\sigma_Y$  of the random variable  $Y = aX + b$  is

$$\sigma_Y = |a| \sigma_X.$$

*Proof.* The variance of  $Y$  is

$$\text{Var}(Y) = \sum_x (ax + b - (aE(X) + b))^2 f(x) = a^2 \sum_x (x - E(X))^2 f(x) = a^2 \sigma_X^2.$$

Taking square roots gives

$$\sigma_Y = |a| \sigma_X.$$

■

In practice, variance is often more convenient for theoretical derivations because of its algebraic properties, while standard deviation is usually preferred for interpretation, since it is expressed in the same units as the original data.

■ **Example B.19** Continuing from Example B.18 the standard deviation is

$$\sigma = \sqrt{\text{Var}(X)} = \sqrt{50} \approx 7.07.$$

Unlike variance, which is expressed in squared minutes, the standard deviation is measured in minutes, the same units as the commuting times themselves. This means that the typical commuting time deviates from the average of 35 minutes by about 7 minutes. ■

### B.5.3 Measures of Statistical Relationship

Covariance and correlation are measures that describe the relationship between two variables. Covariance quantifies the direction of their linear association, indicating whether the variables tend to increase or decrease together. However, covariance is expressed in the product of the units of the two variables, making its magnitude difficult to interpret. Correlation, in contrast, standardizes this measure to a dimensionless quantity between  $-1$  and  $1$ : a value of  $-1$  indicates a perfect negative linear relationship,  $1$  a perfect positive linear relationship, and  $0$  the absence of a linear relationship.

#### Covariance

Covariance is a measure of the joint variability of two discrete random variables  $X$  and  $Y$ . It indicates the direction of their linear relationship: whether they tend to increase together, decrease together, or vary independently.

**Definition B.5.5** Let  $X$  and  $Y$  be two discrete random variables with finite expected values  $E(X)$  and  $E(Y)$ . The *covariance* of  $X$  and  $Y$ , denoted by  $\text{Cov}(X, Y)$ , is defined as

$$\text{Cov}(X, Y) = E[(X - E(X))(Y - E(Y))].$$

The sign of the covariance reveals the direction of the relationship: a positive value means that as  $X$  increases,  $Y$  tends to increase; a negative value means that as  $X$  increases,  $Y$  tends to decrease. A covariance of zero indicates no linear relationship, though nonlinear dependencies may still exist and are not captured by covariance.

The following result provides a useful formula for computing covariance.

**Proposition B.5.6** For any two discrete random variables  $X$  and  $Y$  with finite expectations,

$$\text{Cov}(X, Y) = E(XY) - E(X)E(Y).$$

*Proof.* By definition,

$$\text{Cov}(X, Y) = E[(X - E(X))(Y - E(Y))].$$

Expanding the product inside the expectation,

$$E[XY - XE(Y) - YE(X) + E(X)E(Y)].$$

Using linearity of expectation and the fact that  $E(X)$  and  $E(Y)$  are constants,

$$E(XY) - E(X)E(Y) - E(X)E(Y) + E(X)E(Y) = E(XY) - E(X)E(Y).$$

■

The following proposition shows the connection between independence and covariance.

**Proposition B.5.7** If  $X$  and  $Y$  are independent discrete random variables, then

$$\text{Cov}(X, Y) = 0.$$

*Proof.* From the identity above,

$$\text{Cov}(X, Y) = E(XY) - E(X)E(Y).$$

If  $X$  and  $Y$  are independent, then  $E(XY) = E(X)E(Y)$ . Substituting,

$$\text{Cov}(X, Y) = E(X)E(Y) - E(X)E(Y) = 0.$$

■

The converse does not hold in general: a covariance of zero does not imply independence. Two random variables may be dependent yet uncorrelated.

## Correlation

The magnitude of covariance depends on the scale of the variables, making it difficult to compare across different contexts. To address this, correlation standardizes covariance by dividing by the product of the standard deviations of the variables. This yields a dimensionless measure of the strength and direction of their linear relationship.

**Definition B.5.6** Let  $X$  and  $Y$  be two discrete random variables with finite, nonzero variances  $\text{Var}(X)$  and  $\text{Var}(Y)$ . The *correlation* of  $X$  and  $Y$ , denoted by  $\text{Cor}(X, Y)$  is defined as

$$\text{Cor}(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)} \sqrt{\text{Var}(Y)}}.$$

Correlation takes values in the interval  $[-1, 1]$ .

**Proposition B.5.8** For any pair of random variables  $X$  and  $Y$  with finite nonzero variances,

$$-1 \leq \text{Cor}(X, Y) \leq 1.$$

*Proof.* Let  $U = X - E(X)$  and  $V = Y - E(Y)$ . Then

$$\text{Cov}(X, Y) = E[UV].$$

By the Cauchy–Schwarz inequality,

$$|E[UV]|^2 \leq E[U^2]E[V^2].$$

Noting that  $E[U^2] = \text{Var}(X)$  and  $E[V^2] = \text{Var}(Y)$ , we obtain

$$|\text{Cov}(X, Y)| \leq \sqrt{\text{Var}(X)} \sqrt{\text{Var}(Y)}.$$

Dividing both sides by the product of the standard deviations gives

$$-1 \leq \text{Cor}(X, Y) \leq 1.$$

■

A value of  $\text{Cor}(X, Y) = 1$  indicates a perfect positive linear relationship,  $\text{Cor}(X, Y) = -1$  indicates a perfect negative linear relationship, and  $\text{Cor}(X, Y) = 0$  indicates no linear relationship.

■ **Example B.20** Consider two variables: weekly hours of exercise ( $X$ ) and monthly weight loss in kilograms ( $Y$ ). Suppose the following data are

observed:

Person	$X$	$Y$
1	5	1
2	10	2
3	15	3
4	20	3.5
5	25	4

Computing the correlation coefficient yields  $\approx 0.98$ , indicating a very strong positive linear relationship. This means that individuals who exercise more hours per week tend to lose more weight. However, it is important to note that correlation measures association, not causation. ■

The following proposition establishes that if two discrete random variables are independent and have finite variances, their correlation must be zero.

**Proposition B.5.9** If  $X$  and  $Y$  are independent discrete random variables with finite, nonzero variances, then

$$\text{Cor}(X, Y) = 0.$$

*Proof.* From the definition,

$$\text{Cor}(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)} \sqrt{\text{Var}(Y)}}.$$

If  $X$  and  $Y$  are independent, then  $E(XY) = E(X)E(Y)$ , so  $\text{Cov}(X, Y) = 0$ . Substituting into the formula above gives  $\text{Cor}(X, Y) = 0$ . ■

Note, however, that the converse is not true: correlation equal to zero does not imply independence, since variables can be dependent in nonlinear ways.

## B.6 Common Distributions

In this section, we define and discuss several important families of distributions that are widely used in applications of probability theory. Specifically, we introduce the following families of discrete distributions: uniform, Bernoulli, binomial, and discrete normal. For each family, we briefly describe how it arises in practical problems and why it provides a suitable probability model for certain types of experiments. We also present the explicit form of the probability mass function and highlight some of the fundamental properties of the distributions in each family.

### B.6.1 Uniform Distribution

The uniform distribution is the simplest probability distribution in probability theory. It models situations in which all outcomes are equally likely.

**Definition B.6.1** A discrete random variable  $X$  is said to follow a *uniform distribution* if all outcomes have the same probability. Specifically, if  $X$  takes values in the set  $x_1, x_2, \dots, x_n$ , the probability mass function is

$$P(X = x_i) = \frac{1}{n}, \quad i = 1, 2, \dots, n.$$

A classic example of the discrete uniform distribution is the outcome of rolling a fair die, where each face has probability  $1/6$ . Unlike other distributions introduced in this section, the uniform distribution is determined solely by its set of possible values, with no additional parameters required.

For a discrete uniform random variable  $X$  taking values  $x_1, x_2, \dots, x_n$ , the expected value and variance are given by

$$E(X) = \frac{x_1 + x_2 + \dots + x_n}{n}, \quad \text{Var}(X) = \frac{1}{n} \sum_{i=1}^n (x_i - E(X))^2.$$

### B.6.2 Bernoulli Distribution

The Bernoulli distribution is one of the simplest and most fundamental in probability theory. It models experiments with only two possible outcomes, commonly referred to as "success" and "failure."

**Definition B.6.2** A discrete random variable  $X$  is said to follow a *Bernoulli distribution* with parameter  $p \in [0, 1]$ , written  $X \sim \text{Bernoulli}(p)$ , if it takes the value 1 (success) with probability  $p$  and the value 0 (failure) with probability  $1 - p$ . Its probability mass function is

$$P(X = x) = \begin{cases} p & \text{if } x = 1, \\ 1 - p & \text{if } x = 0. \end{cases}$$

The Bernoulli distribution arises naturally in many applied settings, especially those involving binary outcomes. For example, in medical studies, the result of a treatment can be modeled as a Bernoulli random variable, with "success" representing recovery and "failure" representing no improvement.

The expected value of  $X \sim \text{Bernoulli}(p)$  is  $E(X) = p$ . Intuitively, this reflects the long-run average outcome of many trials. The variance is

$$\text{Var}(X) = E(X^2) - [E(X)]^2 = p - p^2 = p(1 - p).$$

This variance is maximized when  $p = 0.5$ , corresponding to maximum uncertainty when success and failure are equally likely.

**Definition B.6.3** A sequence of independent and identically distributed random variables  $X_1, X_2, \dots$ , each with  $X_i \sim \text{Bernoulli}(p)$ , is called a sequence of *Bernoulli trials* with parameter  $p$ . An infinite sequence of Bernoulli trials is called a *Bernoulli process*.

The Bernoulli process serves as the foundation for more complex stochastic processes and underlies many important models, including the binomial distribution.

### B.6.3 Binomial Distribution

The binomial distribution is a discrete probability distribution that models the number of successes in a fixed number of independent Bernoulli trials, each with the same probability of success. It is one of the most widely used distributions in probability theory for problems involving repeated binary outcomes.

**Definition B.6.4** A discrete random variable  $X$  is said to follow a *binomial distribution* with parameters  $n$  and  $p$  if it represents the number of successes in  $n$  independent Bernoulli trials, each with success probability  $p$ . The probability mass function is given by:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k} \quad k = 0, 1, 2, \dots, n.$$

Here,  $n$  is the number of trials, and  $p$  is the probability of success on each trial. The coefficient  $\binom{n}{k}$  counts the number of ways to obtain exactly  $k$  successes in  $n$  trials.

The binomial distribution arises naturally in many applications where outcomes can be classified as either "success" or "failure."

■ **Example B.21** Suppose a company produces light bulbs, and historically, 95% of them are functional (success) while 5% are defective (failure). The company tests a random sample of 20 bulbs. Let  $X$  be the number of functional bulbs. Then  $X \sim \text{Binomial}(n = 20, p = 0.95)$ . For example, the probability that exactly 18 bulbs are functional is:

$$P(X = 18) = \binom{20}{18} (0.95)^{18} (0.05)^2.$$

The expected value of a binomial random variable  $X$  with parameters  $n$

and  $p$  is

$$E(X) = np,$$

representing the average number of successes in  $n$  trials. Its variance is

$$\text{Var}(X) = np(1 - p),$$

which measures the variability of the number of successes around the expected value.

#### B.6.4 Discrete Normal Distribution

The normal distribution is a continuous probability distribution that frequently appears in natural and social sciences, primarily due to the Central Limit Theorem (see Theorem B.7.5). However, many real-world situations involve discrete outcomes, such as counts of events or integer-valued random variables. In these cases, it is desirable to have a discrete analog of the normal distribution. This is not a limitation, but rather a reflection of the fact that continuous distributions often arise as limiting abstractions of underlying discrete processes.

In this section, we introduce the *discrete normal distribution*, which serves as a discrete counterpart to the normal distribution for integer-valued random variables. Although not as standardized as other families such as the Bernoulli or binomial distributions, the discrete normal can provide a useful tool in applications where outcomes must remain integer-valued. propositions.

To define the discrete normal distribution, we first introduce the concept of a standardized discrete random variable.

**Definition B.6.5** Let  $X$  be a discrete random variable with expected value  $E(X)$  and variance  $\text{Var}(X)$ . The *standardized discrete random variable* of  $X$ , denoted by  $Z$ , is defined as:

$$Z = \frac{X - E(X)}{\sqrt{\text{Var}(X)}}.$$

This transformation centers the distribution around 0 and rescales it so that the variance is 1.

■ **Example B.22** If  $X \sim \text{Bin}(n, p)$ , then its standardized version is

$$Z = \frac{X - np}{\sqrt{np(1 - p)}}.$$

The random variable  $Z$  takes values in the set

$$\left\{ \frac{k-np}{\sqrt{np(1-p)}} : k = 0, 1, \dots, n \right\},$$

with probability mass function

$$P\left(Z = \frac{k-np}{\sqrt{np(1-p)}}\right) = \binom{n}{k} p^k (1-p)^{n-k}.$$

■

The normal distribution arises as the limit of the standardized binomial distribution as  $n \rightarrow \infty$ . However, because the binomial is discrete and the normal is continuous, it is natural to consider a discrete distribution whose probabilities are proportional to the normal density evaluated at integer points.

**Definition B.6.6** A discrete random variable  $X$  is said to follow a *discrete normal distribution* with parameters  $\mu$  and  $\sigma^2$  if it has probability mass function:

$$P(X = k) = \frac{1}{Z} \exp\left(-\frac{(k-\mu)^2}{2\sigma^2}\right), \quad k \in \mathbb{Z},$$

where  $Z$  is the normalization constant:

$$Z = \sum_{k=-\infty}^{\infty} \exp\left(-\frac{(k-\mu)^2}{2\sigma^2}\right).$$

This distribution retains the discrete nature of integer-valued random variables while exhibiting the bell-shaped curve characteristic of the continuous normal distribution.

**Proposition B.6.1** The normalization constant  $Z$  in the discrete normal distribution satisfies:

$$Z \approx \sqrt{2\pi\sigma^2},$$

for large  $\sigma$ , and thus the probability mass function can be approximated by:

$$P(X = k) \approx \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(k-\mu)^2}{2\sigma^2}\right).$$

*Proof.* For large  $\sigma$ , the sum in  $Z$  can be approximated by an integral:

$$Z = \sum_{k=-\infty}^{\infty} \exp\left(-\frac{(k-\mu)^2}{2\sigma^2}\right) \approx \int_{-\infty}^{\infty} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx = \sqrt{2\pi\sigma^2}.$$

■

For large  $\sigma$ , the discrete normal approaches the continuous normal probability den evaluated at integer points.

■ **Example B.23** Suppose we model the number of defective items in a production run of 100 units, where each item is defective with probability  $p = 0.1$  independently. Then  $X \sim \text{Bin}(100, 0.1)$ , with

$$E(X) = 100 \times 0.1 = 10, \quad \text{Var}(X) = 100 \times 0.1 \times 0.9 = 9.$$

For large  $n$ ,  $X$  may be approximated by the discrete normal distribution with parameters  $\mu = 10$  and  $\sigma^2 = 9$ . The probability of exactly 12 defective items is approximated by

$$P(X = 12) \approx \frac{1}{Z} \exp\left(-\frac{(12 - 10)^2}{2 \times 9}\right).$$

This provides an alternative to the usual continuous normal approximation, avoiding the need for continuity corrections and keeping the support discrete.

■

The following proposition highlights an important property of the discrete normal distribution, specifically its symmetry around the mean.

**Proposition B.6.2** The discrete normal distribution is symmetric about its mean  $\mu$ . Formally, for any integer  $k$ ,

$$P(X = \mu + k) = P(X = \mu - k).$$

*Proof.* From the definition,

$$P(X = \mu + k) = \frac{1}{Z} \exp\left(-\frac{k^2}{2\sigma^2}\right),$$

and similarly,

$$P(X = \mu - k) = \frac{1}{Z} \exp\left(-\frac{(-k)^2}{2\sigma^2}\right) = \frac{1}{Z} \exp\left(-\frac{k^2}{2\sigma^2}\right).$$

Thus the distribution is symmetric about  $\mu$ . ■

The discrete normal distribution serves as a bridge between discrete and continuous probability models. It inherits key properties of the normal distribution—such as symmetry and centrality—while remaining defined on the integers. This makes it particularly useful in contexts such as manufacturing, epidemiology, or educational testing, where outcomes are integer-valued but approximately normal in shape.

## B.7 Large Random Samples

A random sample is a collection of independent and identically distributed random variables. Random samples are fundamental in probability theory, providing the foundation for making inferences about an unknown probability distribution. Two key results that arise from random samples are the law of large numbers, which ensures that the sample mean converges to the expected value of the distribution as the sample size increases, and the central limit theorem, which states that the distribution of the sample mean approaches a normal distribution as the sample size grows, regardless of the original population's distribution.

Random sampling assumes that every member of the population has an equal probability of being selected and that selections are independent of one another. In this sense, the sample is an idealized representation of how real-world data might be generated. Perfect random sampling is rarely achievable in practice, but it provides a baseline against which deviations can be evaluated. This framework allows us to construct probability distributions describing the expected behavior of sample statistics (e.g., sample mean, variance), making it possible to quantify uncertainty and to draw principled conclusions about the population.

The random sampling model is therefore a powerful representation of how data might be collected, though it is not identical to reality. Like all models, it works well in specific contexts, but its assumptions do not universally hold. The practical usefulness of random sampling depends on how closely the model aligns with the real-world situation under study.

### B.7.1 Random Sample

The concept of a random sample is pivotal in the field of statistical learning, since assuming that a set of discrete random variables forms a random sample greatly simplifies the mathematical foundations of inference. However, the criteria required for a collection of variables to be considered a random sample are not always satisfied in real-world scenarios.

**Definition B.7.1** Let  $f$  be a probability mass function, and let  $X_1, X_2, \dots, X_n$  be a collection of discrete random variables. The variables  $X_1, X_2, \dots, X_n$  form a *random sample* from the distribution  $f$  if each  $X_i$  follows the probability mass function  $f$  and the variables  $X_1, X_2, \dots, X_n$  are mutually independent. Such variables are called *independent and identically distributed* (*i.i.d.*). The number  $n$  of variables is called the *sample size*.

The joint probability mass function  $g$  of a random sample  $X_1, X_2, \dots, X_n$

is

$$g(x_1, x_2, \dots, x_n) = f(x_1)f(x_2) \dots f(x_n).$$

■ **Example B.24** A factory produces light bulbs, and the quality control department is interested in estimating the proportion of defective bulbs. The number of defectives in a batch of 100 bulbs is modeled by a binomial distribution with parameter  $p$ , the probability of a bulb being defective. Suppose the department randomly selects 5 batches, each of 100 bulbs. Let  $X_1, \dots, X_5$  denote the number of defectives in each batch. Each  $X_i \sim \text{Binomial}(100, p)$ , and assuming the batches are independent, the variables  $X_1, \dots, X_5$  form a random sample. ■

In statistical inference, it is convenient to compute the sample mean, as the average of  $n$  random variables.

**Definition B.7.2** Let  $X_1, X_2, \dots, X_n$  be  $n$  discrete random variables. The *sample mean*, denoted by  $\bar{X}_n$ , is:

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$$

Do not confuse the random variable  $\bar{X}_n = \frac{1}{n}(X_1 + \dots + X_n)$  with the population mean  $E(X_i)$ , which is a fixed real number. Recall that  $X_1 + \dots + X_n$  defines a new random variable on the Cartesian product of the original sample spaces, not a sum of probability distributions.

The following proposition shows that the sample mean provides an unbiased estimate of the population mean, and that its variance decreases as the sample size increases.

**Proposition B.7.1** Let  $X_1, \dots, X_n$  be a random sample with finite mean  $E(X_i) = \mu$  and variance  $\text{Var}(X_i) = \sigma^2$ . Then

$$E(\bar{X}_n) = \mu, \quad \text{Var}(\bar{X}_n) = \frac{\sigma^2}{n}.$$

*Proof.* By linearity of expectation,

$$E(\bar{X}_n) = E\left(\frac{1}{n} \sum_{i=1}^n X_i\right) = \frac{1}{n} \sum_{i=1}^n E(X_i) = \mu.$$

For the variance,

$$\text{Var}(\bar{X}_n) = \text{Var}\left(\frac{1}{n} \sum_{i=1}^n X_i\right) = \frac{1}{n^2} \text{Var}\left(\sum_{i=1}^n X_i\right).$$

Since the  $X_i$  are independent,

$$\text{Var}\left(\sum_{i=1}^n X_i\right) = \sum_{i=1}^n \text{Var}(X_i) = n\sigma^2,$$

so

$$\text{Var}(\bar{X}_n) = \frac{1}{n^2} \cdot n\sigma^2 = \frac{\sigma^2}{n}.$$

■

Thus, the sample mean is unbiased, and its variance shrinks at rate  $1/n$ . As  $n$  grows, the distribution of  $\bar{X}_n$  becomes more concentrated around the population mean  $\mu$ .

In statistical inference, it is also convenient to compute the sample variance as a measure of the dispersion of the sample values.

**Definition B.7.3** Let  $X_1, X_2, \dots, X_n$  be a set of discrete random variables with sample mean  $\bar{X}_n$ . The *sample variance*, denoted by  $S^2$ , is:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2$$

The factor  $1/(n-1)$  ensures that  $S^2$  is an unbiased estimator of the population variance when the  $X_i$  are i.i.d. Do not confuse the sample variance, which is computed from observed data, with the theoretical variance of the underlying distribution.

■ **Example B.25** Suppose we observe  $X_1 = 6, X_2 = 4, X_3 = 5, X_4 = 7, X_5 = 3$ . Then

$$\bar{X}_5 = \frac{6+4+5+7+3}{5} = 5,$$

and

$$S^2 = \frac{1}{4} [(6-5)^2 + (4-5)^2 + (5-5)^2 + (7-5)^2 + (3-5)^2] = \frac{10}{4} = 2.5.$$

■

The sample standard deviation is used to estimate the standard deviation of a population based on a sample taken from it. Unlike the population standard deviation, which uses the true mean of the entire population, the sample standard deviation uses the mean of the sample as an estimate of the true mean.

**Definition B.7.4** Let  $X_1, X_2, \dots, X_n$  be  $n$  discrete random variables with sample mean  $\bar{X}_n$  and sample variance  $S^2$ . The *sample standard deviation*, denoted by  $S$ , is:

$$S = \sqrt{S^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2}$$

■ **Example B.26** Using the variance from the previous example, the sample standard deviation is

$$S = \sqrt{2.5} \approx 1.58.$$

■

Finally, when two variables are sampled together, the sample covariance estimates the covariance between their populations.

**Definition B.7.5** Let  $X_1, X_2, \dots, X_n$  and  $Y_1, Y_2, \dots, Y_n$  be two sets of discrete random variables with sample means  $\bar{X}$  and  $\bar{Y}$  respectively. The *sample covariance* is:

$$s_{XY} = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$$

■ **Example B.27** Suppose we observe the paired data

$i$	1	2	3	4	5
$X_i$	2	4	6	8	10
$Y_i$	1	3	5	7	9

Then

$$\bar{X} = 6, \quad \bar{Y} = 5,$$

and

$$s_{XY} = \frac{1}{4} [(-4)(-4) + (-2)(-2) + (0)(0) + (2)(2) + (4)(4)] = \frac{40}{4} = 10.$$

■

## B.7.2 Law of Large Numbers

The law of large numbers states that the sample mean of a large random sample, i.e., a large number of independent and identically distributed (i.i.d.) discrete random variables, will be close to the expected value of the distribution, and that as we increase the sample size, the sample mean gets closer

(in a precise probabilistic sense) to that value. Before proving this theorem, we establish two related propositions: Markov's inequality and Chebyshev's inequality.

Markov's inequality provides a simple bound on how unlikely it is for a nonnegative discrete random variable to take a large value, given its expected value. Intuitively, if the average value of a nonnegative random variable  $X$  is small, then  $X$  is unlikely to be very large (recall Example B.16). Markov's inequality is particularly useful because it requires no assumptions on the distribution of  $X$  beyond nonnegativity and finiteness of the mean.

**Proposition B.7.2 — Markov's Inequality.** Let  $X$  be a nonnegative discrete random variable, i.e.  $P(X \geq 0) = 1$ , with expected value  $E(X)$ . Then for every real number  $t > 0$  we have that

$$P(X \geq t) \leq \frac{E(X)}{t}$$

*Proof.* Let  $f$  be the probability mass function of  $X$ . Since  $X$  is non-negative we have that  $E(X) = \sum_{x>0} xf(x)$ . Then

$$\begin{aligned} E(X) &= \sum_{x>0} xf(x) = \sum_{x=0}^t xf(x) + \sum_{x=t}^{\infty} xf(x) \geq \\ &\sum_{x=t}^{\infty} xf(x) \geq \sum_{x=t}^{\infty} tf(x) = t \sum_{x=t}^{\infty} f(x) = tP(X \geq t) \end{aligned}$$

that is,  $E(X) \geq tP(X \geq t)$ . ■

For example, if  $E(X) = 10$  and we ask for  $P(X \geq 50)$ , Markov's inequality gives  $P(X \geq 50) \leq 10/50 = 0.2$ . While Markov's inequality is very general and easy to use, its bound is often not tight because it uses only the mean of  $X$ .

Chebyshev's inequality bounds the probability that a random variable deviates from its expected value by more than a specified number of standard deviations. Unlike Markov's inequality, which only requires non-negativity, it does not require  $X$  to be nonnegative, but it requires knowledge of both the mean and variance.

**Corollary B.7.3 — Chebyshev's Inequality.** Let  $X$  be a discrete random variable with finite mean  $E(X)$  and finite variance  $\text{Var}(X)$ . Then for every  $t > 0$ ,

$$P(|X - E(X)| \geq t) \leq \frac{\text{Var}(X)}{t^2}.$$

*Proof.* Apply Markov's inequality to the nonnegative random variable  $(X - E(X))^2$ :

$$P(|X - E(X)| \geq t) = P((X - E(X))^2 \geq t^2) \leq \frac{E[(X - E(X))^2]}{t^2} = \frac{\text{Var}(X)}{t^2}.$$

■

Chebyshev's inequality formalizes the idea that most of the probability mass of a distribution lies near its mean: the probability of being several standard deviations away decays at least on the order of  $1/t^2$ .

The last notion we need for the law of large numbers is convergence in probability. Intuitively, a sequence of random variables converges in probability to  $b$  if the values  $X_n$  concentrate around  $b$  as  $n$  increases.

**Definition B.7.6** Let  $X_1, X_2, \dots$  be a sequence of discrete random variables. It is said that the sequence  $X_1, X_2, \dots$  converges in probability to  $b$ , denoted by  $X_n \xrightarrow{p} b$ , if for every positive real number  $\varepsilon > 0$ ,

$$\lim_{n \rightarrow \infty} P(|X_n - b| < \varepsilon) = 1$$

Given these elements, we can state and prove the (weak) law of large numbers.

**Theorem B.7.4 — Law of Large Numbers.** Let  $X_1, X_2, \dots$  be i.i.d. random variables with finite mean  $E(X_i) = \mu$  and finite variance  $\text{Var}(X_i) = \sigma^2$ . Let  $\bar{X}_n = \frac{1}{n}(X_1 + \dots + X_n)$  be the sample mean. Then

$$\bar{X}_n \xrightarrow{p} \mu.$$

*Proof.* By Proposition B.7.1,  $E(\bar{X}_n) = \mu$  and  $\text{Var}(\bar{X}_n) = \sigma^2/n$ . Applying Chebyshev's inequality,

$$P(|\bar{X}_n - \mu| \geq \varepsilon) \leq \frac{\sigma^2}{n\varepsilon^2}.$$

From there we can obtain

$$P(|\bar{X}_n - \mu| < \varepsilon) = 1 - P(|\bar{X}_n - \mu| \geq \varepsilon) \geq 1 - \frac{\sigma^2}{n\varepsilon^2}.$$

As  $n$  approaches infinity, the above expression approaches 1. Thus,

$$\bar{X}_n \xrightarrow{p} \mu$$

■

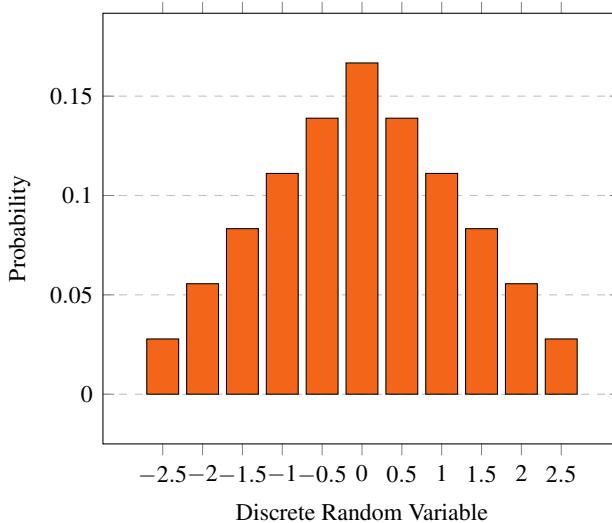


Figure B.1: Probability mass function of the discrete random variable  $\frac{X_1 + X_2}{2} - E(X)$ .

It is important to note that the assumptions (independence and identical distribution, with finite variance) are essential. The theorem is an asymptotic statement: for a finite sample, the sample mean is typically close to the population mean, but not necessarily equal to it.

■ **Example B.28** Consider rolling a fair six-sided die with sample space  $\Omega = \{1, 2, 3, 4, 5, 6\}$  and  $P(\omega) = 1/6$  for each outcome. Let  $X : \Omega \rightarrow \mathbb{R}$  map each side to its face value, so  $E(X) = 3.5$ .

First take a random sample of size two: roll the die twice (in sequence, so the rolls are distinguishable) and let  $X_1$  and  $X_2$  be the outcomes. The sample mean is the random variable

$$\frac{X_1 + X_2}{2} : \Omega \times \Omega \rightarrow \mathbb{R}, \quad (\omega_1, \omega_2) \mapsto \frac{X_1(\omega_1) + X_2(\omega_2)}{2}.$$

whose probability mass function is depicted in Figure B.1. A direct calculation shows

$$\begin{aligned} P\left(\left|\frac{X_1 + X_2}{2} - E(X)\right| < 1\right) &= P(|X_1 + X_2 - 7| < 2) \\ &= P(X_1 + X_2 \in \{6, 7, 8\}) = \frac{16}{36} \approx 0.444. \end{aligned}$$

Now consider a random sample of size ten. The distribution of

$$\frac{X_1 + \dots + X_{10}}{10} - E(X)$$

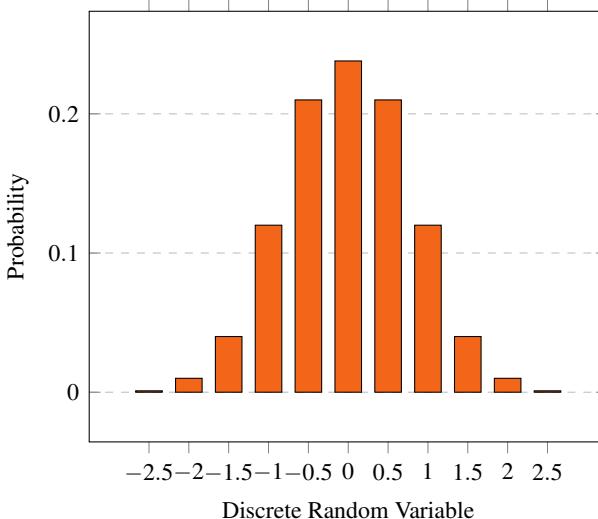


Figure B.2: Probability mass function of the discrete random variable  $\frac{X_1 + X_2 + \dots + X_{10}}{10} - E(X)$ .

is much more concentrated around 0 (see Figure B.2); numerically,

$$P\left(\left|\frac{X_1 + \dots + X_{10}}{10} - 3.5\right| < 1\right) \approx 0.97,$$

illustrating the sharpening concentration predicted by the law of large numbers. ■

Finally, the law of large numbers concerns the *sample mean*  $\frac{1}{n} \sum_{i=1}^n X_i$ , not other aggregates. For instance, the deviation from the expected total,

$$\sum_{i=1}^n X_i - nE(X),$$

does not converge to 0; in fact, its typical size grows on the order of  $\sqrt{n}$  (see Example B.29). Confusing these statements underlies a common misconception.

■ **Example B.29** If we toss a fair coin, the probability of heads is  $1/2$ . By the law of large numbers, the *proportion* of heads in many tosses is close to  $1/2$ . However, the *difference* between the number of heads and tails need not be close to 0; in fact, its typical magnitude grows like a constant multiple of  $\sqrt{n}$  as the number of tosses  $n$  increases. The belief that the counts of heads and tails should stay nearly equal as  $n$  grows is a form of the gambler's fallacy. ■

### B.7.3 Central Limit Theorem

Let  $X_1, \dots, X_n$  be a sample of  $n$  independent and identically distributed random variables with mean  $\mu$  and variance  $\sigma^2$ . As we saw in the previous section, the law of large numbers states that the sample average  $\bar{X}_n$  converges in probability to  $\mu$  as  $n$  increases. The central limit theorem strengthens this by describing the distribution of the centered and scaled sample mean: the quantity  $\sqrt{n}(\bar{X}_n - \mu)$  converges in distribution to a normal random variable with mean 0 and variance  $\sigma^2$ . Equivalently, the standardized statistic

$$Z_n = \frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}}$$

converges in distribution to the standard normal distribution  $N(0, 1)$ . This result holds regardless of the shape of the underlying distribution, provided it has finite variance.

**Theorem B.7.5 — Central Limit Theorem.** Let  $X_1, \dots, X_n$  be i.i.d. random variables with mean  $\mu$  and finite variance  $\sigma^2$ . Define

$$Z_n = \frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}}, \quad \text{where} \quad \bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i.$$

Then, as  $n \rightarrow \infty$ ,

$$Z_n \Rightarrow N(0, 1),$$

i.e.,  $Z_n$  converges in distribution to the standard normal law with density  $\phi(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$ . Equivalently,

$$\sqrt{n}(\bar{X}_n - \mu) \Rightarrow N(0, \sigma^2).$$

*Proof sketch.* We first consider the special case of Bernoulli random variables. Let  $X_i \sim \text{Bernoulli}(p)$  and  $S_n = \sum_{i=1}^n X_i$ . The standardized sum is

$$Z_n = \frac{S_n - np}{\sqrt{np(1-p)}}.$$

Using Stirling's approximation for factorials in the binomial formula, one shows that for integers  $k$  close to  $np$  (within order  $\sqrt{n}$ ),

$$P(S_n = k) \approx \frac{1}{\sqrt{2\pi np(1-p)}} \exp\left(-\frac{(k-np)^2}{2np(1-p)}\right).$$

This is exactly the probability mass function of a discretized normal distribution. Summing over such  $k$  produces a Riemann sum converging to the normal distribution function, proving the result for Bernoulli variables.

For general i.i.d. random variables with finite mean and variance, one reduces to the Bernoulli case by truncating the variables to control large deviations and matching means and variances. The same approximation then applies, and the limiting distribution is again standard normal. ■

The central limit theorem is a cornerstone of probability and statistics. It justifies normal approximations for sampling distributions of means and enables approximate inference. The basic conditions used here are i.i.d. observations with finite variance.

■ **Example B.30** Suppose a factory produces light bulbs and the lifetime of each bulb (measured, say, in hours) has mean  $\mu = 1000$  and standard deviation  $\sigma = 50$ . A random sample of  $n = 36$  bulbs is tested. What is the probability that the sample mean lifetime lies between 990 and 1010 hours?

By the CLT, for  $n = 36$  the sample mean  $\bar{X}_n$  is approximately normal with mean  $\mu$  and standard deviation  $\sigma/\sqrt{n} = 50/6 \approx 8.33$ :

$$\bar{X}_n \approx N\left(1000, \left(\frac{50}{\sqrt{36}}\right)^2\right).$$

Standardizing,

$$z_{990} = \frac{990 - 1000}{8.33} \approx -1.20, \quad z_{1010} = \frac{1010 - 1000}{8.33} \approx 1.20.$$

Thus

$$\begin{aligned} P(990 \leq \bar{X}_n \leq 1010) &\approx P(-1.20 \leq Z \leq 1.20) = \Phi(1.20) - \Phi(-1.20) \\ &\approx 0.8849 - 0.1151 = 0.7698. \end{aligned}$$

So the probability is about 76.98%. ■

## References

[DeG+86] is a widely respected textbook in the fields of statistics and probability theory. First published in 1975, this book is known for its clear exposition of the fundamental concepts of probability and statistics, making it suitable for both beginners and those with some background in the subject. The book's approach balances theory and application, making it useful both for learning theoretical underpinnings and for applying probability and statistics to real-world problems.

[Chi13] offers a comprehensive introduction to the foundational aspects of probability, with a focus on the philosophical questions it raises. Childers

explores various interpretations of probability, including frequentist, propensity, classical, Bayesian, and objective Bayesian, and presents these complex ideas in a way that is accessible even to those without a strong background in probability or mathematics.

An example of the problems associated with the misinterpretation of expected value is the St. Petersburg Paradox. Introduced by Nicholas Bernoulli in 1713, this paradox involves a gambling game with an infinite expected payoff, yet no reasonable person would pay more than \$25 to play it. Despite being three centuries old, the paradox continues to inspire new arguments and solutions in recent years (see [Hua13] for a historical review of the main proposed solutions).



## C. Computability

*Wanderer, there is no road,  
the road is made by walking.*

Antonio Machado

We continue our review of the background required to understand the theory of nescience by providing a mathematical formalization of the concept of a *computable procedure*. Intuitively, a computable procedure is a method consisting of a finite sequence of instructions that, when applied to a problem, produces the correct answer after a finite number of steps. This informal notion rests on the requirement that the instructions be clear and precise enough for any human to follow without additional guidance. We may strengthen this requirement further by demanding that the instructions be so unambiguous that a machine could execute them.

In 1936, the British mathematician Alan Turing introduced a formal model of a family of hypothetical machines and argued that, for every intuitively computable procedure, there exists a *Turing machine* capable of carrying it out. The model is not only simple enough to allow precise mathematical analysis but also sufficiently general to capture the intuitive concept of effective computation.

Over the years, many alternative approaches have been proposed to formalize the notion of computability. Some of these are technically intricate, yet all have been shown to be equivalent in expressive power to Turing machines; in other words, they characterize the same class of effectively computable functions. Two notable examples are Alonzo Church's *lambda calculus* and the *theory of recursive functions* developed by Kurt Gödel and Stephen Kleene. The *Church-Turing thesis* asserts that any reasonable formalization of a computable procedure (subject to minimal requirements, such as performing only a finite amount of work in a single step) coincides with the Turing machine model. While not a theorem, this thesis has become a widely accepted working principle, providing a stable and shared notion of computability that is independent of any specific formalism.

The concept of the Turing machine, originally conceived as a model of a mechanical device designed to solve a particular problem, was later extended and generalized. A *universal Turing machine* can simulate the behavior of any other Turing machine and thereby compute any function that is computable in the intuitive sense. This is analogous to modern digital computers, which can execute algorithms expressed in diverse programming languages. This universality raises a natural question: Are there problems that no Turing machine can solve? The answer is yes. Certain well-defined problems, such as the Halting problem, lie beyond the reach of computation, and such problems are more common than one might initially expect. The existence of uncomputable functions will play a central role in our theory of science.

Given the abstract character of many entities studied in science, we employ the concept of the *oracle Turing machine* to formalize our framework. An oracle Turing machine resembles an ordinary Turing machine but is augmented with the ability to query an external oracle. The oracle, whose internal mechanism is unspecified, provides answers to questions that may be uncomputable for standard machines. Different oracles yield different computational powers, giving rise to a hierarchy of relative computability. For our purposes, the oracle can be viewed as a theoretical construct that models access to sources of information beyond the limits of mechanical computation. It should be emphasized that the oracle is not a physical device but an abstract tool for reasoning about the boundaries of computation.

Turing machines reveal the intrinsic limitations of computation. Exploring these limitations is not merely a philosophical exercise; it has profound implications for the field of *computational complexity*. Situated at the intersection of computer science and mathematics, computational complexity studies the resources (most notably time and space) required to solve problems. Problems are classified into *complexity classes* according to the

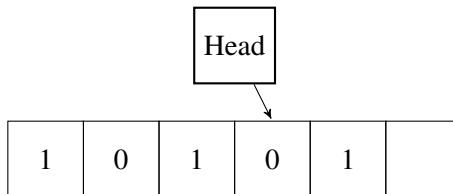


Figure C.1: Turing Machine

asymptotic resources needed by the best algorithms known for their solution. One of the most famous open problems in this field is the  $P \stackrel{?}{=} NP$  question, which asks whether the class  $P$  of problems solvable in polynomial time coincides with the class  $NP$  of problems for which proposed solutions can be verified in polynomial time. In this book, our interest extends beyond the epistemological issue of determining which problems can be solved in principle, given unlimited resources, to the more practical question of which problems can be solved efficiently with respect to time.

## C.1 Turing Machines

A Turing machine is an extremely simplified model of a general-purpose computer, yet it is powerful enough to capture the notion of effective computation. Intuitively, the machine consists of a head that operates on a two-way infinite tape divided into cells, each containing a symbol. At each time step, the machine reads the symbol under the head and, according to a fixed set of rules, either writes a new symbol, moves the head one cell to the left or right, or performs both actions. The rules are encoded in a finite control table. The input to the computation is written on the tape at the start, and once the machine enters a designated halting state, the output can be read from the tape. Figure C.1 illustrates a machine in its initial configuration, with the head positioned at the beginning of the input string.

**Definition C.1.1 — Turing Machine.** A *Turing machine* is a 7-tuple

$(Q, \Gamma, \sqcup, \Sigma, q_0, q_f, \tau)$  where:

- $Q$  is a finite, non-empty set of *states*,
- $\Gamma$  is a finite, non-empty set of *tape symbols*,
- $\sqcup \in \Gamma$  is the *blank symbol*,
- $\Sigma \subseteq \Gamma \setminus \{\sqcup\}$  is the set of *input symbols*,
- $q_0 \in Q$  is the *initial state*,
- $q_f \in Q \setminus \{q_0\}$  is the designated *halting state*,
- $\tau : (Q \setminus \{q_f\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$  is the *transition*.

The algorithm executed by the machine is defined by the (partial) transition function  $\tau$ . This function dictates the machine's actions based on its current state and the tape symbol currently under the head. According to  $\tau$ , the machine transitions to a new state, writes a new symbol on the tape (or retains the existing one), and moves the head left, right, or keeps it stationary ( $L$ ,  $R$ , or  $S$  respectively). The machine follows a finite, uniquely determined sequence of steps until it reaches the final state  $q_f$ <sup>1</sup> and *halts*, making no subsequent moves. The algorithm's output is the string of symbols  $s \in \Gamma^*$  remaining on the tape after halting. Some machines, however, may enter an infinite loop, never reaching a halting state. If a machine encounters an undefined transition, it will enter an infinite loop, never halting. We assume that the head of the machine is initially positioned at the first symbol of the input string.

■ **Example C.1** The following Turing machine is designed to solve the problem of adding two natural numbers. It consists of the set of states  $Q = \{q_0, q_1, q_f\}$ , the set of tape symbols  $\Gamma = \{0, 1, \sqcup\}$ , and the set of input symbols  $\Sigma = \{0, 1\}$ . The transition function is defined in the table below, where rows are indexed by machine states, and columns by tape symbols:

	0	1	$\sqcup$
$q_0$	$(q_f, \sqcup, R)$	$(q_1, \sqcup, R)$	$\uparrow$
$q_1$	$(q_f, 1, S)$	$(q_1, 1, R)$	$\uparrow$

Table C.1: Transition Rules

For natural numbers  $n$  and  $m$ , the input string is composed of  $n$  occurrences of the symbol '1', followed by a '0', and then followed by  $m$  occurrences of '1'. The machine's output will be a string of  $n+m$  consecu-

<sup>1</sup>Some authors allow multiple halting states in the definition of Turing machines.

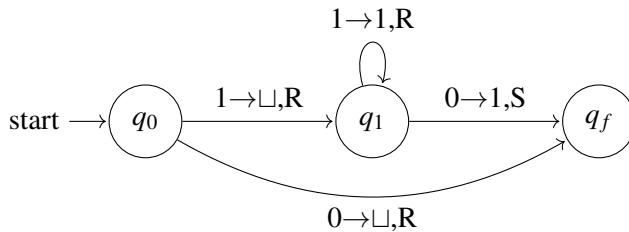


Figure C.2: State diagram of the addition machine

tive '1's. For instance, to add the numbers 2 and 3, the input string should be  $\sqcup 110111\sqcup$ , resulting in the output string  $\sqcup 11111\sqcup$ . ■

A Turing machine can also be represented by a *state diagram*. A state diagram is similar to a labeled directed graph<sup>2</sup> where the vertices represent the states of the machine. The edges signify transitions from one state to another, and the edge labels indicate the symbol under the head that leads to the new state, the symbol that gets written on the tape, and the direction in which the head moves. Following these conventions, the state diagram for the Turing machine in Example C.1 is depicted in Figure C.2.

It is a remarkable fact that minor alterations to the definition of a Turing machine do not change its computational power. In other words, the definition is highly robust. In Example C.2, it's demonstrated that adding more tapes to the machine doesn't expand the range of problems it can solve. Similar arguments can be made when adding finite storage to the control, allowing for parallel processing with multiple control heads, and so on.

■ **Example C.2** A *multitape Turing machine* is a Turing machine equipped with multiple heads and their respective tapes. In the initial configuration, the input string resides in tape 1, while the other tapes are blank. The transition function for a multitape Turing machine is:

$$\tau : (Q \setminus q_f) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, S\}^k,$$

where  $k$  denotes the number of tapes. Multitape Turing machines are equivalent in power to standard Turing machines. We can validate this claim by devising a method for a standard Turing machine to mimic a multitape machine's behavior. This requires encoding the content of multiple tapes onto a single tape, introducing a new symbol as a tape separator, and encoding the positions of the heads across the tapes with a distinct head location symbol. If we designate the tape separation symbol as  $|$  and the head location symbol as  $h$ , a simulation tape for a machine with 3 tapes might appear as

<sup>2</sup>In this particular case, we allow loops and multiple edges originating from vertices.

$\sqcup 01h00|000h1|h0101\sqcup$ . The standard machine's operation would involve scanning the subtapes one by one, pinpointing the head's location, and executing the necessary transition. If the computation on one subtape necessitates writing a new symbol beyond its boundary, we'd need to shift subsequent symbols to accommodate the new one. While the simulation might operate at a slower pace than the original multitape machine, both machine types can solve an identical set of problems. ■

For the remainder of this book, without any loss of generality, we'll assume that the set of input symbols is  $\Sigma = \mathcal{B}$  and the set of tape symbols is  $\Gamma = \{0, 1, \sqcup\}$ .

In addition to providing a formal definition of a Turing machine, it's essential to formally outline its computational process. This entails detailing how the machine reads the input string, produces the output string, and transitions between states during computation. We will start by defining the concept of the machine's internal configuration. This configuration captures the machine's current state and position, as well as the present state of the tape.

**Definition C.1.2** A *configuration* of a Turing machine  $T$  is the 3-tuple  $(q, s, i)$ , where  $q \in Q$  represents a state of the machine,  $s \in \Gamma^+$  denotes a string containing the tape's content (excluding the blank symbols), and  $1 \leq i \leq n$  is the index of the symbol  $s_i$  beneath the head. Here,  $s_1$  is the first symbol on the tape, and  $n = l(s)$ .

Configurations enable us to describe the current state of a Turing machine without any loss of information. At any stage of computation, one could halt the machine, record its configuration, and later resume the computation from the exact point of interruption using this configuration.

The following definition explains how we transition from one configuration to the next during computation.

**Definition C.1.3** A configuration  $C = (q, s, i)$  yields another configuration  $C' = (r, s', j)$  if there exists a transition  $\tau : (q, s_i) = (r, s'_i, a)$ , where  $s = s_1 \dots s_{i-1} s_i s_{i+1} \dots s_n$ ,  $s' = s_1 \dots s_{i-1} s'_i s_{i+1} \dots s_n$ , and

$$j = \begin{cases} i + 1 & \text{if } a = R \\ i - 1 & \text{if } a = L \\ i & \text{if } a = S \end{cases} \quad (\text{C.1})$$

If  $a = L$  and  $i = 1$ , we extend  $s$  on the left by a blank and keep the head at position 1 (now on  $\sqcup$ ); if  $a = R$  and  $i = n$ , we extend  $s$  on the right by a blank and set the head to position  $n + 1$ .

Building on the concepts of configuration and one configuration yielding another, we can now formally articulate the notion of computation.

**Definition C.1.4 — Computation.** Let  $T$  be a Turing machine,  $C_0$  its initial configuration, and  $C_n$  a configuration encompassing the final state  $q_f$ . A *computation* under machine  $T$  refers to a finite sequence of  $n + 1$  configurations  $(C_0, C_1, \dots, C_n)$  wherein each configuration  $C_k$  yields the subsequent configuration  $C_{k+1}$ , for all  $0 \leq k < n$ .

Computations are deterministic; meaning, for a given Turing machine  $T$  and an input string  $s$ , the configuration sequence is uniquely determined.. If machine  $T$  neither halts nor progresses with input  $s$ , we deduce the absence of computation.

■ **Example C.3** The computation of the Turing machine described in Example C.1 using the input string 110111 results in the following sequence of configurations:

- 1  $(q_0, 110111, 1)$
- 2  $(q_1, 10111, 1)$
- 3  $(q_1, 10111, 2)$
- 4  $(q_f, 11111, 2)$

■

Intuitively, a procedure is deemed computable by a human if it can be delineated through specific steps, executed systematically, without relying on intuition or ingenuity. This intuitive grasp aligns with the formalized concept of a Turing machine, bridging informal comprehension and the machine's rigorous definition—a cornerstone in the theory of computation. However, this alignment presents an intriguing challenge. Affirming that our grasp of computability mirrors a Turing machine's capabilities cannot be proven traditionally, as 'computability' lacks a well-defined interpretation. Consequently, some researchers categorize this as a *thesis*, avoiding the formal 'theorem' label. Turing himself opted to term it a *definition*, steering clear of denoting it as a theorem.

**Theorem C.1.1 — Turing's Thesis.** A procedure is computable if, and only if, it can be executed by a Turing machine.

To further underscore the significance and robustness of the Turing machine as a model of computation, it's worth noting, as mentioned earlier in this chapter, that all alternative formalizations of computability proposed to date align in terms of their computational capabilities with that of the Turing machine. This universality underscores the Turing machine's central position in the realm of theoretical computer science.

## C.2 Universal Turing Machines

In Section C.1, we explored storing the current state of a Turing machine, its configuration, to pause and later resume computation. In Example C.4, we will delve into a similar procedure, not for storing the machine's current state, but for saving a comprehensive description of the machine itself. This methodology facilitates the enumeration, or listing, of all possible Turing machines. Such enumeration is instrumental in demonstrating the existence of problems that cannot be solved by any Turing machine (refer to Section C.3) and unveiling the pivotal concept of the *Universal Turing Machine*.

■ **Example C.4** To describe a Turing machine concisely, we need to encode the transition function  $\tau : (Q \setminus \{q_f\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$ . This function can be represented as a collection of quintuples  $(q, s, r, t, a)$ , where  $q \in (Q \setminus \{q_f\})$ ,  $r \in Q$ ,  $s, t \in \Gamma$ , and  $a \in \{L, R, S\}$ . In this manner, any Turing machine  $T$  is fully described by a collection of quintuples:

$$(q_1, s_1, r_1, t_1, a_1), (q_2, s_2, r_2, t_2, a_2), \dots, (q_m, s_m, r_m, t_m, a_m)$$

where  $m \leq d(Q \setminus \{q_f\}) \times d(\Gamma)$ , with the stipulation that the first quintuple refers to the initial state and the second one to the final state; i.e.,  $q_1 = q_o$  and  $r_2 = q_f$ . A possible approach to describe these quintuples is to encode the elements of the set  $Q \cup \Gamma \cup \{L, R, S\}$  using a fixed-length binary code (refer to Definition D.1.6 for more details), encoding the quintuple  $(q, s, r, t, a)$  as  $\langle q, s, r, t, a \rangle$ . The length of an encoded quintuple is  $5l$ , where  $l = \lceil \log(d(Q \cup \Gamma \cup \{L, R, S\})) \rceil$ . Following this convention, machine  $T$  is encoded as the binary string:

$$\langle T \rangle = \langle \bar{l}, \langle q_1, s_1, r_1, t_1, a_1 \rangle, \dots, \langle q_r, r_r, s_r, t_r, a_r \rangle \rangle$$

The length of the encoded machine, following this schema, would be  $l(\langle T \rangle) \leq 5lm + \log l + 1$ . ■

Since each Turing machine is composed by a finite set of quintuples, we can encode and list all the machines using a shortlex ordering. We associate each machine  $T$  with the index  $i$  corresponding to its position in this list, and we denote by  $T_i$  the  $i$ -th Turing machine. Each positive integer  $i$  encodes one, and only one, Turing machine. However, as Proposition C.2.1 shows, all Turing machines have an infinite number of indexes. We associate each Turing machine with its smallest index.

**Proposition C.2.1 — Padding Lemma.** Each Turing machine has infinitely many indexes.

*Proof.* Consider a Turing machine  $T_i$  encoded by the string  $\langle T_i \rangle$ . We can create a new encoding  $\langle T_j \rangle$  by appending a finite number of 0's to  $\langle T_i \rangle$ ,

such that  $\langle T_j \rangle = \langle T_i \rangle 0^n$  for some positive integer  $n$ . Since  $n$  can take on any positive integer value, there are infinitely many possible encodings  $\langle T_j \rangle$  for the same Turing machine  $T_i$ . ■

A universal Turing machine is a machine that can simulate the behavior of any other Turing machine on arbitrary input. The universal machine achieves this by reading both the description of the machine to be simulated (for instance, using the coding schema described in Example C.4) and the input string for the computation from its own tape.

**Definition C.2.1 — Universal Turing Machine.** A *Universal Turing Machine* is a Turing machine  $U$  such that  $U(\langle\langle T_i \rangle, s\rangle) = T_i(s)$  for all Turing machines  $T_i$  and all input strings  $s \in \mathcal{B}$ .

Naturally, we must prove that such a machine exists before we can utilize it. One could argue that a human being could decode the machine  $T_i$  and simulate its behavior with the input string  $s$ , and then refer to Theorem C.1.1. A more rigorous approach would be to explicitly construct a universal Turing machine. However, providing a detailed description of one of these machines is beyond the scope of this book. Instead, we direct the reader to the references included at the end of the chapter for further exploration.

### C.3 Non-Computable Problems

Turing machines enable us to delineate the set of problems that can be resolved through effective procedures or, in other words, by computers. It may be surprising to learn that numerous problems cannot be addressed using algorithms; such challenges lie beyond the computational capabilities of machines. We are not alluding to speculative queries like whether a computer can be intelligent or self-aware but to concrete, well-defined mathematical problems. We are also not referring to complex problems that demand a substantial amount of time to solve, as those, irrespective of their time consumption, remain computable.

One classic exemplar of non-computability is the *halting problem*. As illustrated in Algorithm C.1, it involves a program or algorithm tasked with determining whether any given program (including itself) and input will eventually halt or continue to run indefinitely. Alan Turing proved that no algorithm can exist to solve this problem for all possible program-input pairs. This revelation wasn't a reflection on the limitations of technology or processing power but highlighted a profound theoretical limit intrinsic to computation.

The proposition below proves that the halting problem is non-computable.

**Algorithm C.1** HALT function

---

```

procedure HALT( $A, I$ )
  if  $A(I)$  halts then
    return 1
  else
    return 0
  end if
end procedure

```

---

**Proposition C.3.1 — Halting Problem.** Define HALT as in Algorithm C.1. There does not exist a Turing machine that computes the HALT function for all possible pairs  $(A, I)$ , where  $A$  is a Turing machine and  $I$  is the input string to that machine.

*Proof.* The proof is by contradiction. Assume that the machine  $HALT$  exists, and define a new Turing machine  $TC$  such that  $TC(A) = 1$  if  $HALT(A, A) = 0$ , and  $TC(A)$  will never stop if  $HALT(A, A) = 1$ . Then the contradiction arises when we ask about the result of  $TC(TC)$ : if  $TC(TC)$  stops we have that  $HALT(TC, TC) = 0$  and that  $TC(TC)$  should not stop, and if  $TC(TC)$  does not stop then we have that  $TC(TC) = 1$  and thus  $TC(TC)$  should stop. ■

The existence of such non-computable problems underscores the boundaries of mechanical computation. It illustrates that while Turing machines, and by extension, computers are profoundly powerful tools capable of solving an extensive array of problems, they are not omnipotent. A frontier of unsolvable problems exists, necessitating deeper exploration into the realms of mathematics, logic, and perhaps even philosophy to understand the inherent limits of computation.

The Halting Problem also has significant practical consequences in computer programming. For instance, it is impossible to write a program that can guarantee any other arbitrary program is bug-free or that all infinite loops with conditional exits will eventually halt for all possible inputs.

The next example introduces a well-defined, practical problem involving simple string manipulation that cannot be solved using computers.

■ **Example C.5** Given two finite lists  $(\alpha_1, \dots, \alpha_n)$  and  $(\beta_1, \dots, \beta_n)$  of strings over some alphabet  $\Sigma$ , where  $d(\Sigma) \geq 2$ , the *Post Correspondence Problem* (PCP) asks to determine if there exists a sequence of  $K \geq 1$  indices  $(i_k)$ , with  $1 \leq i_k \leq n$  for all  $1 \leq k \leq K$ , such that  $\alpha_{i_1} \dots \alpha_{i_K} = \beta_{i_1} \dots \beta_{i_K}$ . For instance, given the sequences  $(a, ab, bba)$  and  $(baa, aa, bb)$ , a solution would be  $\alpha_3 \alpha_2 \alpha_3 \alpha_1 = \beta_3 \beta_2 \beta_3 \beta_1$ . No algorithm exists to solve PCP. Like many

proofs of incomputability, the proof proceeds by showing that HALT can be reduced to PCP, meaning if PCP is decidable, then the Halting Problem should be decidable as well. We will not detail the proof in this section; for interested readers, we refer to the references at the end of this chapter. ■

Non-computable problems are generally not derived directly from natural phenomena but from logical and mathematical constructs. To date, there are no known examples of non-computable problems manifesting plainly in natural phenomena. It's essential to distinguish between non-computability and unpredictability. Non-computable problems are those for which no algorithm can ever be created to solve them. In contrast, unpredictable systems (such as chaotic or complex systems) are theoretically computable but are unpredictable in practice due to factors like sensitivity to initial conditions or measurement precision.

## C.4 Computable Functions and Sets

Each Turing machine  $T$  defines a function  $f_T : \mathcal{B}^* \rightarrow \mathcal{B}^*$  that assigns to each input string  $s \in \mathcal{B}^*$  an output string  $T(s) \in \mathcal{B}^*$ . The relationship between Turing machines and functions forms the basis for introducing the concept of a *computable function*.

**Definition C.4.1** A function  $f : \mathcal{B}^* \rightarrow \mathcal{B}^*$  is *computable* if there exists a Turing machine  $T$  that defines the function  $f$  and halts for all the values of  $\mathcal{B}^*$ .

The terminology in computational theory can vary. While computable functions are occasionally referred to as *recursive functions*, this book opts for the term computable functions for consistency.

**Example C.6** The function that assigns to each pair of natural numbers  $x$  and  $y$  their sum  $x + y$  is computable, as demonstrated in Example C.1. ■

In real-world scenarios, certain functions don't provide a defined output for all possible inputs. Partial computable functions, characterized by Turing machines that don't halt for specific inputs, model these cases.

**Definition C.4.2** A partial function  $f : \mathcal{B}^* \rightarrow \mathcal{B}^*$  is *partial computable* if there exists a Turing machine  $T$  that defines  $f$  for defined values and does not halt for undefined values.

The distinction between total computable functions and partial computable functions is significant in computability theory because it reflects the difference between problems that are always solvable by an algorithm (total) and those that are only solvable in some cases (partial).

■ **Example C.7** The function  $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  that assigns to each pair of natural numbers  $x$  and  $y$  the number  $x - y$  is a partial computable function, since it is not defined in the case that  $x < y$ . Recall that according to our definition of Turing machine (see Definition C.1.1), when the machine reaches an undefined configuration enters an infinite loop without ever halting. ■

We can expand the application of the principles of computability and partial computability to the domain of sets. We characterize sets through the lens of their characteristic functions that discern the membership of elements within the sets.

**Definition C.4.3** A set  $A \in \mathcal{B}^*$  is *computable* if its characteristic function  $\chi_A$  is a total computable function. A set  $A \in \mathcal{B}^*$  is *computably enumerable* if its characteristic function  $\chi_A$  is a partial computable function, that is,  $\chi_A(a) = 1$  if  $a \in A$ , but  $\chi_A(a)$  is undefined if  $a \notin A$ .

The application of these concepts is illustrated through the example of the set of all Turing machines that halt for all inputs.

■ **Example C.8** The set of all Turing machines that halt on all inputs, as demonstrated in C.3.1, is not computable but is computably enumerable. ■

Next proposition provides an alternative characterization of computable sets.

**Proposition C.4.1** A set  $A \in \mathcal{B}^*$  is computable if and only if  $A$  and its complement  $A^c$  are computably enumerable.

*Proof.* If  $A$  is computable, by definition, there exists a Turing machine that decides for any input  $x \in \mathcal{B}^*$  whether  $x \in A$  or  $x \notin A$ , halting in both cases. This implies that both  $A$  and its complement  $A^c$  can be enumerated by Turing machines. Thus, both  $A$  and  $A^c$  are computably enumerable.

Conversely, suppose  $A$  and  $A^c$  are computably enumerable. This means there exist two Turing machines,  $T_A$  and  $T_{A^c}$ , that enumerate the elements of  $A$  and  $A^c$ , respectively. To show that  $A$  is computable, construct a Turing machine  $T$  that, given an input  $x \in \mathcal{B}^*$ , simulates  $T_A$  and  $T_{A^c}$  in parallel to search for  $x$ . If  $x$  appears in the enumeration produced by  $T_A$ ,  $T$  halts and accepts  $x$  as an element of  $A$ . If  $x$  appears in the enumeration produced by  $T_{A^c}$ ,  $T$  halts and accepts  $x$ , indicating  $x \notin A$ . Since every element of  $\mathcal{B}^*$  must be in either  $A$  or  $A^c$  and both sets are computably enumerable,  $T$  will eventually halt for every input  $x$ , proving that  $A$  is computable. ■

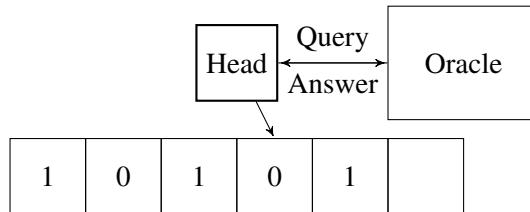


Figure C.3: Oracle Turing Machine

## C.5 Oracle Turing Machine

An oracle Turing machine (see Figure C.3) is a theoretical model of computation that extends the capabilities of a standard Turing machine by providing it with an oracle. The oracle is a black box that can instantly compute certain answers, even for problems that are unsolvable or would take an impractical amount of time for a standard Turing machine to process. This model helps computer scientists and mathematicians explore the implications and boundaries of computational theory, including questions about complexity classes and the limits of what is computationally possible. The oracle Turing machine isn't a physical or implementable machine but rather a conceptual tool used in theoretical studies.

**Definition C.5.1 — Oracle Turing Machine.** An *oracle Turing machine* with oracle set  $\mathcal{O}$  is a 8-tuple  $(Q, \Gamma, \sqcup, \Sigma, q_i, q_f, \tau, \mathcal{O})$  where:

- $Q$  is a finite, non-empty, set of *states*,
- $\Gamma$  is a finite, non-empty, set of *tape symbols*,
- $\sqcup \in \Gamma$  is the *blank symbol*,
- $\Sigma \subseteq \Gamma \setminus \sqcup$  is the set of *input symbols*,
- $q_0 \in Q$  is the *initial state*,
- $q_f \in Q \setminus \{q_0\}$  is the *final state*,
- $\tau : (Q \setminus \{q_f\}) \times \Gamma \times \{0, 1\} \rightarrow Q \times \Gamma \times \{L, R, S\}$  is the *transition*,
- $\mathcal{O} \subseteq \Sigma^*$  is the *oracle set*.

Building on the concept of a regular Turing machine (refer to Definition C.1.1), an oracle Turing machine introduces the unique feature of an oracle set  $\mathcal{O}$ . This set comprises a subset of strings for which the oracle can instantly provide answers. The true strength of an oracle machine emerges when the set  $\mathcal{O}$  is non-computable; in cases where  $\mathcal{O}$  is computable, a regular Turing machine would be sufficient.

The transition function  $\tau$  for the oracle Turing machine is nuanced. At

each step the machine will query the oracle. Specifically, it sends the current string  $w$ —starting under the head and extending to the rightmost non-empty cell of the input tape—to the oracle. The oracle then responds with a binary answer: '1' if  $w$  is in  $\mathcal{O}$  or '0' if it isn't. The machine doesn't always utilize this response. There are steps where, despite receiving an answer, the computation proceeds unaffected by the oracle's response—essentially ignoring it. However, when the oracle's answer is pivotal, the machine makes a decision based on it. This decision could affect the next state, the next symbol to be written, or the next move (left, right, or stay). In essence, while the machine has the capability to continuously consult the oracle, it strategically chooses when to act on the information received. We could have modified the  $\tau$  function in such a way that the oracle is queried only when the answer is relevant for the computation, but that would require making important changes to the behaviour of the function, such as the introduction of new control states.

■ **Example C.9** In the realm of theoretical computer science, an oracle can be invoked to "solve" the halting problem (refer to Theorem C.3.1). The oracle is a hypothetical device or black box that, as if by magic, provides an instantaneous answer to a specific problem instance. The oracle set  $\mathcal{O}$  would consist of a collection of strings in the form  $\langle P, I \rangle$ , where  $P$  represents a program encoded as a string and  $I$  denotes its input. Given a program and its input, this oracle would instantly inform us whether the program halts on that input. Naturally, this concept is purely theoretical. No such oracle exists in reality, and the halting problem remains unsolvable in practical terms. ■

A standard Turing machine can be viewed as an oracle machine that simply *ignores* its oracle or, equivalently, uses any fixed oracle set and never branches on the oracle bit. Thus ordinary Turing machines embed as a subclass of oracle Turing machines.

**Proposition C.5.1** For every (ordinary) Turing machine  $T = (Q, \Gamma, \sqcup, \Sigma, q_0, q_f, \tau)$  and every oracle set  $\mathcal{O} \subseteq \Sigma^*$ , there exists an oracle Turing machine  $T^{\mathcal{O}}$  such that for all inputs  $x \in \Sigma^*$ , the computation of  $T$  on  $x$  and the computation of  $T^{\mathcal{O}}$  on  $x$  (with oracle  $\mathcal{O}$ ) produce the same outcome.

*Proof.* Define  $T^{\mathcal{O}} = (Q, \Gamma, \sqcup, \Sigma, q_0, q_f, \tau', \mathcal{O})$  with the same components as  $T$  and transition function  $\tau'$  given by

$$\tau'(q, s, b) = \tau(q, s) \quad \text{for all } (q, s) \in (Q \setminus \{q_f\}) \times \Gamma \text{ and } b \in \{0, 1\}.$$

That is,  $T^{\mathcal{O}}$  ignores the oracle bit. A step-by-step induction on the unique computation of  $T$  on  $x$  shows that  $T$  and  $T^{\mathcal{O}}$  visit identical configurations

(up to the presence of the unused oracle bit) and therefore have the same outcome and output. ■

An oracle machine is specified by two pieces of data: the *machine component*  $(Q, \Gamma, \sqcup, \Sigma, q_0, q_f, \tau)$  and the *oracle set*  $\mathcal{O}$ . The former is defined syntactically and does not depend on which  $\mathcal{O}$  is chosen. Semantically, the behavior of a fixed machine component on a given input depends only on the answers to the queries it actually makes (i.e., on the restriction of  $\mathcal{O}$  to those queries).

**Proposition C.5.2** Let  $M = (Q, \Gamma, \sqcup, \Sigma, q_0, q_f, \tau)$  be a fixed machine component. For any two oracle sets  $\mathcal{O}_1, \mathcal{O}_2 \subseteq \Sigma^*$  and any input  $x$ , if  $\mathcal{O}_1$  and  $\mathcal{O}_2$  give the same answers to all oracle queries made during the run of  $M$  on input  $x$  (with either oracle), then the computations  $M^{\mathcal{O}_1}(x)$  and  $M^{\mathcal{O}_2}(x)$  are identical (same sequence of configurations, hence same outcome and output).

*Proof.* Consider the (unique) run of  $M^{\mathcal{O}_1}$  on input  $x$ . By hypothesis,  $\mathcal{O}_2$  agrees with  $\mathcal{O}_1$  on precisely the queried strings that arise along this run. Since  $\tau$  is deterministic given the current state, scanned symbol, and oracle bit, and the oracle bit provided by  $\mathcal{O}_2$  matches that from  $\mathcal{O}_1$  at every step, both computations take the same transition at each step. Thus the configuration sequences coincide, yielding identical outcomes. ■

A (partial) function is *computable relative to*  $\mathcal{O}$  if some oracle machine with oracle  $\mathcal{O}$  computes it. Two useful facts: (i) ordinary computable functions are  $\mathcal{O}$ -computable for any oracle (by ignoring the oracle); (ii) if  $\mathcal{O}$  is itself computable, then  $\mathcal{O}$ -computability coincides with ordinary computability.

**Definition C.5.2 — Oracle computability.** Given  $\mathcal{O} \subseteq \Sigma^*$ , a partial function  $f : \Sigma^* \rightharpoonup \Sigma^*$  is  $\mathcal{O}$ -computable if there exists an oracle Turing machine  $M^{\mathcal{O}}$  such that, for every  $x \in \Sigma^*$ :

- if  $f(x)$  is defined, then  $M^{\mathcal{O}}$  halts on input  $x$  and outputs  $f(x)$  (under the chosen output convention);
- if  $f(x)$  is undefined, then  $M^{\mathcal{O}}$  does not halt on input  $x$ .

**Proposition C.5.3** If  $\mathcal{O}$  is (ordinarily) computable, then a partial function  $f$  is  $\mathcal{O}$ -computable if and only if  $f$  is computable by an ordinary Turing machine.

*Proof.* Suppose  $f$  is  $\mathcal{O}$ -computable via  $M^{\mathcal{O}}$ . Since  $\mathcal{O}$  is computable, there exists an ordinary machine  $D$  deciding membership in  $\mathcal{O}$ . Simulate  $M^{\mathcal{O}}$  step by step on an ordinary machine; whenever  $M^{\mathcal{O}}$  needs the oracle bit, call  $D$

on the current query string to obtain the bit and continue the simulation. This yields an ordinary machine computing  $f$ .

If  $f$  is computable by an ordinary machine  $T$ , then by Proposition C.5.1 there is an oracle machine  $T^{\mathcal{O}}$  that ignores the oracle and computes  $f$  for any  $\mathcal{O}$ . Hence  $f$  is  $\mathcal{O}$ -computable. ■

More generally, if  $\mathcal{O}$  is *Turing reducible* to  $\mathcal{O}'$  (i.e., decidable by some  $\mathcal{O}'$ -oracle machine), then every  $\mathcal{O}$ -computable function is  $\mathcal{O}'$ -computable. This captures the standard notion of *relative computability*.

■ **Example C.10 — Toy oracle machine that outputs the oracle's answer.**

Let  $\Sigma = 0, 1$  and  $\Gamma = 0, 1, \sqcup$ . Fix the oracle set  $\mathcal{O} = 11, 101$  (finite for concreteness). Consider the oracle machine  $M^{\mathcal{O}}$  that, on any input  $w \in \Sigma^*$ , writes a single symbol indicating the oracle's answer and halts:

- If the queried string (as per your convention) is in  $\mathcal{O}$ , write 1 and halt.
- Otherwise, write 0 and halt.

Use states  $Q = q_0, q_f$ , with  $q_0$  initial and  $q_f$  halting. Define  $\tau$  independently of the scanned symbol  $s \in \Gamma$  by the sextuples

$$(q_0, 0, 1, q_f, 1, S), \quad (q_0, 1, 1, q_f, 1, S), \quad (q_0, \sqcup, 1, q_f, 1, S),$$

$$(q_0, 0, 0, q_f, 0, S), \quad (q_0, 1, 0, q_f, 0, S), \quad (q_0, \sqcup, 0, q_f, 0, S),$$

and leave all other transitions undefined (so the machine halts after one step). Then  $\langle M \rangle$  is obtained by concatenating the fixed-length codes of these six sextuples after the header. The oracle's finite code is, for example,

$$\langle \mathcal{O} \rangle = \langle 2, \langle 11 \rangle, \langle 101 \rangle \rangle,$$

using any self-delimiting code for strings. The full encoding is the pairing  $\langle \langle M \rangle, \langle \mathcal{O} \rangle \rangle$ . ■

These additions make explicit (i) how ordinary computation embeds into oracle computation, (ii) in what precise sense the machine component is independent of the oracle set, (iii) that an oracle machine is exactly a pair  $(M, \mathcal{O})$ , and (iv) what it means for a function to be computable *relative* to an oracle. The encoding example matches your earlier quintuple-based scheme with a minimal extension to handle the oracle bit.

## References

The original paper from Alan Turing where the concepts of Turing machine, universal Turing machine, and non-computable problems were introduced is [Tur36], however it is a difficult to read paper for the contemporary reader.

An easier to read introduction to computability theory, from the point of view of languages, can be found in [Sip12], and a more advanced introductions in [Coo03] and [Soa16]. In [Fer09] we can find a description of the most important computability models proposed so far. The Post Correspondence Problem was introduced by Emil Post in [Pos46]; for the details of the proof sketched in Example C.5 please refer to [Sip12].





## D. Coding

*Information is the resolution of uncertainty.*

Claude Shannon

In this section, we review the conceptual foundations and main results of coding theory and the closely related field of information theory.

Coding is the process of representing a sequence of symbols from one alphabet by a sequence of symbols from another alphabet. Coding has many practical applications, such as error detection and correction, cryptography, and telecommunications. Our focus here is on data compression, that is, encoding a message using fewer symbols than in its original representation, while preserving all the information it contains. Compression algorithms reduce message size by identifying and eliminating statistical redundancies. For example, one may assign shorter descriptions to the most frequent symbols in the source and longer descriptions to the less frequent ones. A particular class of codes, known as prefix-free codes, will play a central role in this book. Prefix-free codes establish a natural connection between coding theory and probability theory, a connection that will be essential in the context of the theory of nescience.

Information theory proposes that the amount of information conveyed by the occurrence of an event is given by the negative logarithm of the

probability of that event. In this sense, information can be interpreted as surprise: the less likely an event is, the more information we gain when it occurs. Although we will not adopt this interpretation of information in the theory of nescience, we will rely extensively on another fundamental concept from information theory: entropy. Entropy quantifies the uncertainty in the value of a random variable or the outcome of a random process. It is particularly relevant because it establishes a fundamental bound on compression: no code can achieve an average word length smaller than the entropy of the source alphabet.

There exist many interesting concepts derived from entropy, like joint entropy, conditional entropy, or mutual information. However, these concepts are more relevant in the context of communication because they allow us to solve the problem of how to transmit information in a reliable manner over a noisy channel. Here, they are introduced for completeness purposes, and to compare them with our own definitions of joint nescience and conditional nescience.

Several additional concepts are derived from entropy, including joint entropy, conditional entropy, and mutual information. These quantities play a central role in communication theory, as they are instrumental in addressing the problem of transmitting information reliably over noisy channels. In this book, however, they are introduced primarily for completeness and as a basis for comparison with our own definitions of joint nescience and conditional nescience.

## D.1 Coding

Intuitively, coding refers to the process of losslessly describing a sequence of symbols (a message) from some alphabet by another sequence of symbols from a (possibly different) alphabet. There is no universal agreement on what exactly constitutes a code, as different authors propose different definitions in the literature. Fortunately, the definition of a prefix-free code (the kind of code used in the theory of nescience) is standard and widely accepted.

Let  $\mathcal{S} = \{s_1, s_2, \dots, s_q\}$  be a finite set called the *source alphabet*, and let  $\mathcal{X} = \{x_1, x_2, \dots, x_r\}$  be a finite set called the *code alphabet*.

**Definition D.1.1 — Code.** A *code* for  $\mathcal{S}$  is a total function  $C : \mathcal{S} \rightarrow \mathcal{X}^+$ . If  $(s, x) \in C$  we say that  $s$  is a *source symbol* and  $x$  is the corresponding *code word*. If  $C$  is injective, we say that the code is *nonsingular*.

Nonsingularity ensures that each symbol in the source alphabet is represented by a distinct code word, allowing us to unambiguously describe the individual symbols of the source alphabet. In what follows, unless other-

wise stated, the term *code* will always mean a nonsingular code. Moreover, without loss of generality, we restrict our discussion to *binary codes*, that is,  $\mathcal{X} = \mathcal{B} = \{0, 1\}$ .

The property of nonsingularity can also be extended from individual symbols to strings of symbols. For that, we introduce the notion of extensions of a code.

**Definition D.1.2** The *extension of order n* of a code  $C$  is a function  $C^n : \mathcal{S}^n \rightarrow \mathcal{B}^+$  defined by

$$C^n(s_{i_1} \dots s_{i_n}) = C(s_{i_1}) \dots C(s_{i_n}),$$

where the right-hand side is the concatenation of the code words corresponding to the symbols  $s_{i_1} \dots s_{i_n} \in \mathcal{S}^n$ . An extension of order  $n$  is *nonsingular* if  $C^n$  is injective.

If it is clear from the context, we will also use the word *code* to refer to a nonsingular extension of order  $n$  of a code, and the elements of  $\mathcal{S}^n$  will be called *source words*.

■ **Example D.1** The code defined by  $C(a) = 0$ ,  $C(b) = 00$ ,  $C(c) = 01$ , and  $C(d) = 11$  is nonsingular, but its extension of order 2 is singular, since  $C^2(ab) = C^2(ba) = 000$ . ■

As Example D.1 illustrates, not all nonsingular codes have nonsingular extensions. In such cases, it may be impossible to recover the original message from its encoded version. To address this, we introduce the stronger property of unique decodability.

**Definition D.1.3** A code  $C$  is *uniquely decodable* if its extension  $C^n$  is nonsingular for all  $n$ .

The following proposition provides a useful characterization of unique decodability.

**Proposition D.1.1** A code  $C$  is uniquely decodable if and only if the function  $C^+ : \mathcal{S}^+ \rightarrow \mathcal{B}^+$  is injective.

*Proof.* If  $C^+$  is injective, then every restriction  $C^n$  is injective, and the code is uniquely decodable.

Conversely, assume that  $C^n$  is nonsingular for all  $n$ . Suppose, for contradiction, that  $C^+$  is not injective. Then there exist two source words  $s_1 \in \mathcal{S}^n$  and  $s_2 \in \mathcal{S}^m$  with  $n \neq m$  such that  $C^+(s_1) = C^+(s_2)$ . Construct  $s_3 = s_1s_2$  and  $s_4 = s_2s_1$ . Both have the same length, but

$$C^+(s_3) = C^+(s_1)C^+(s_2) = C^+(s_2)C^+(s_1) = C^+(s_4),$$

contradicting the injectivity of  $C^{n+m}$ . Therefore  $C^+$  must be injective. ■

■ **Example D.2** The code  $C(a) = 0$ ,  $C(b) = 01$ ,  $C(c) = 011$ , and  $C(d) = 0111$  is uniquely decodable. For example, the code word 0010011 corresponds uniquely to the source word *abac*. Unique decodability is achieved because the leading 0 acts as a delimiter, separating code words. ■

The Sardinas-Patterson theorem provides a necessary and sufficient condition for unique decodability. The theorem is based on an algorithmic construction that examines successive sets of possible ambiguities.

Let  $C_0$  denote the set of code words of a code  $C$ . Define

$$C_n = C^{-1}C_{n-1} \cup C_{n-1}^{-1}C,$$

for all  $n \in \mathbb{N}$ , where  $C^{-1}D = \{y \mid \exists x \in C, xy \in D\}$  is the left quotient of  $C$  and  $D$ . Finally, let

$$C_\infty = \bigcup_{n=1}^{\infty} C_n.$$

**Proposition D.1.2 — Sardinas-Patterson.** A code  $C$  is uniquely decodable if and only if  $C_0 \cap C_\infty = \emptyset$ .

*Proof.* If  $C_0 \cap C_\infty \neq \emptyset$ , then there exists a nonempty overlap between some code word and a concatenation of code words, implying that at least two distinct source words map to the same encoded string, so  $C$  is not uniquely decodable.

Conversely, if  $C_0 \cap C_\infty = \emptyset$ , then the iterative construction guarantees that no ambiguity arises in the decoding process, regardless of message length. Therefore the code is uniquely decodable. ■

■ **Example D.3** For the code of Example D.1:

$$C_0 = \{0, 00, 01, 11\}$$

$$C_1 = \{0, 1\}$$

Since  $C_0 \cap C_1 \neq \emptyset$ , the code is not uniquely decodable.

For the code of Example D.2:

$$C_0 = \{0, 01, 011, 0111\}$$

$$C_1 = \{1, 11, 111\}$$

$$C_2 = \emptyset$$

Here  $C_0 \cap C_\infty = C_0 \cap \bigcup_{n=1}^{\infty} C_n = C_0 \cap C_1 = \emptyset$ , so the code is uniquely decodable. ■

Next definition introduces the concept of prefix-free codes. Prefix-free codes will play a critical role in the computation of the amount of algorithmic information of an arbitrary string (described in Chapter E), and in our own theory of nescience. Prefix-free codes also allow us to link coding theory and probability theory through the Kraft inequality (Theorem D.2.1). Note that we prefer the name *prefix-free code* over the more standard *prefix code*, since the former more accurately describes the concept.

**Definition D.1.4 — Prefix-free Code.** A code  $C$  is *prefix-free* if for all  $i, j$  whth  $1 \leq i, j \leq q$  and  $i \neq j$ ,  $C(s_i)$  is not a prefix of  $C(s_j)$ .

Note that the fact that a string is a prefix of itself does not violate the prefix-free property, because the condition specifically excludes considering a string as a prefix of itself ( $i \neq j$ ) when determining whether a set of code words is prefix-free.

■ **Example D.4** The code  $C(a) = 0$ ,  $C(b) = 10$ ,  $C(c) = 110$ , and  $C(d) = 1110$  is prefix-free. Here, the symbol 0 plays the role of a delimiter (as in Example D.2), but placing it at the end of each code word ensures that no code word can be a prefix of another. ■

Prefix-free codes are uniquely decodable, as the next proposition shows.

**Proposition D.1.3** Let  $C$  be a prefix-free code. Then  $C$  is uniquely decodable.

*Proof.* Assume, for contradiction, that  $C$  is not uniquely decodable. Then there exist two distinct sequences  $r \neq s$  of source symbols such that  $C(r) = C(s)$ . Among all such pairs, choose one with the minimal common encoded length. Let  $r = ar'$  and  $s = bs'$  with  $a \neq b$  (if  $a = b$ , cancel the common first symbols to get a shorter counterexample, contradicting minimality). Then  $C(a)$  is a prefix of  $C(b)C(s')$  and  $C(b)$  is a prefix of  $C(a)C(r')$ . In particular, one of  $C(a)$  or  $C(b)$  must be a prefix of the other, contradicting the prefix-free property. Hence  $C$  must be uniquely decodable. ■

From an engineering point of view, it is highly convenient to have codes whose source symbols can be decoded as soon as the corresponding code words are received. In other words, it should not be necessary to wait for the next code word in order to decode the current symbol. For this reason, some authors refer to these codes as *instantaneous codes*.

**Definition D.1.5** A code  $C$  is *instantaneous* if, for any order  $n$  and for any sequence of code words  $C(s_{i_1}), C(s_{i_2}), \dots, C(s_{i_n})$ , each sequence of code words  $\mathbf{t} = C(s_{i_1})C(s_{i_2}) \dots C(s_{i_m}) \dots$  can be uniquely decoded as  $\mathbf{s} = s_{i_1}s_{i_2} \dots s_{i_m} \dots$  without ambiguity, regardless of the continuation of  $\mathbf{t}$ .

■ **Example D.5** The code described in Example D.2 is not instantaneous. For instance, after receiving the sequence 011, the decoded symbol could be  $c$  if the next bit is 0 or  $d$  if it is 1; thus, one must look ahead to decide. ■

Prefix-free codes and instantaneous codes are essentially two terms for the same concept. A prefix-free code is one in which no code word is a prefix of any other code word. This ensures a clear boundary between consecutive code words, allowing immediate decoding upon receipt.

**Proposition D.1.4** A code  $C$  is instantaneous if and only if it is prefix-free.

*Proof.* Suppose  $C$  is instantaneous but not prefix-free. Then there exist at least a pair of code words, say  $C(s_i)$  and  $C(s_j)$ , such that  $C(s_i)$  is a prefix of  $C(s_j)$ . Consider a sequence where  $C(s_j)$  is transmitted. Since  $C(s_i)$  is a prefix of  $C(s_j)$ , the decoder would decode  $C(s_i)$  from the initial part of  $C(s_j)$ , leading to ambiguity, as the rest of  $C(s_j)$  could be seen as another code or part of the next code. This contradicts the assumption that  $C$  is instantaneous.

Suppose  $C$  is prefix-free. Reading a concatenation of code words from left to right, the first position at which a code word ends is unambiguous, because no longer code word can have the current code word as a prefix. Thus each code word can be decoded immediately upon completion, with no lookahead. Hence  $C$  is instantaneous. ■

The last type of codes we review are fixed-length codes. We will use fixed-length codes to compute the length of a text when there are no regularities we can exploit to compress it.

**Definition D.1.6** If all the code words of a code have the same length, we say that the code is a *fixed-length code*.

Fixed-length codes are prefix-free (and therefore instantaneous).

**Proposition D.1.5** Let  $C$  be a fixed-length code, then  $C$  is prefix-free.

*Proof.* Let  $C$  be a fixed length code, and  $C(s_i)$  and  $C(s_j)$  the code words of two arbitrary source words  $s_i$  and  $s_j$ . Assume that  $C(s_i) <_p C(s_j)$ , given the fact that  $l(C(s_i)) = l(C(s_j))$  we have that  $C(s_i) = C(s_j)$  and so, the code  $C$  is prefix-free. ■

Of course, the converse does not hold: not all prefix-free codes are fixed-length. The code in Example D.4 is prefix-free but not fixed-length.

Figure D.1 provides a graphical representation of the relationships among the different types of codes introduced in this section.

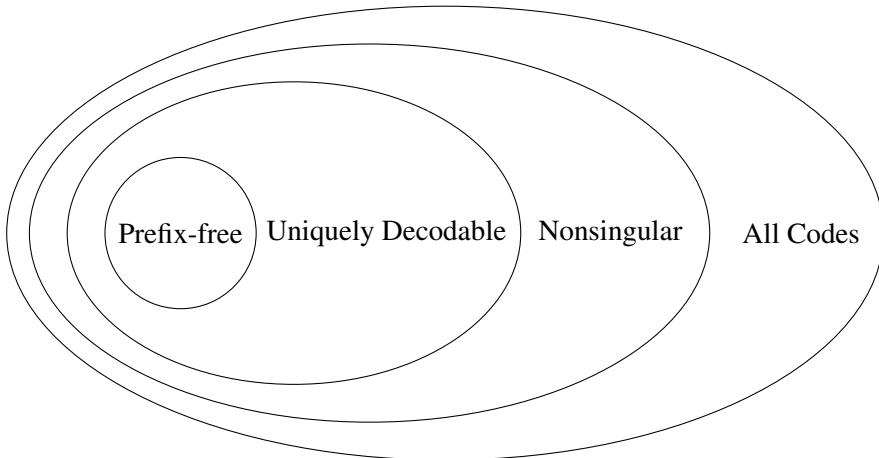


Figure D.1: Classification of Codes

## D.2 Kraft Inequality

The Kraft inequality provides a condition for the existence of a prefix-free code given a set of code word lengths. Kraft's inequality states that for a given set of code word lengths in a binary code, the sum of the reciprocals of powers of two corresponding to the code word lengths must be less than or equal to one. This condition is both necessary and sufficient: any prefix-free code must satisfy this inequality, and given a set of lengths that meets this condition, it is always possible to construct a corresponding prefix-free code. The elegance and utility of Kraft's inequality lie in its ability to link the lengths of code words with probabilistic structure.

**Theorem D.2.1 — Kraft Inequality.** Let  $\mathcal{L} = \{l_1, l_2, \dots, l_q\}$  be a set of lengths, where  $l_i \in \mathbb{N}$ , then there exists a binary prefix-free code  $C$  whose code words have the lengths of  $\mathcal{L}$  if, and only if,

$$\sum_{l_i \in \mathcal{L}} 2^{-l_i} \leq 1$$

*Proof.* Consider a binary tree whose branches are labeled with the symbols of the code alphabet, in such a way that the path from the root to the leaves traces out the symbols of a code word. The prefix-free condition implies that nodes representing complete code words cannot have descendants. An example of such a tree, for the code described in Example D.4, is shown in Figure D.2.

Let  $l_{\max} = \max \{l_1, l_2, \dots, l_q\}$ , that is, the length of the longest code word from the set of lengths. There will be at most  $2^{l_{\max}}$  leaf nodes in the tree,

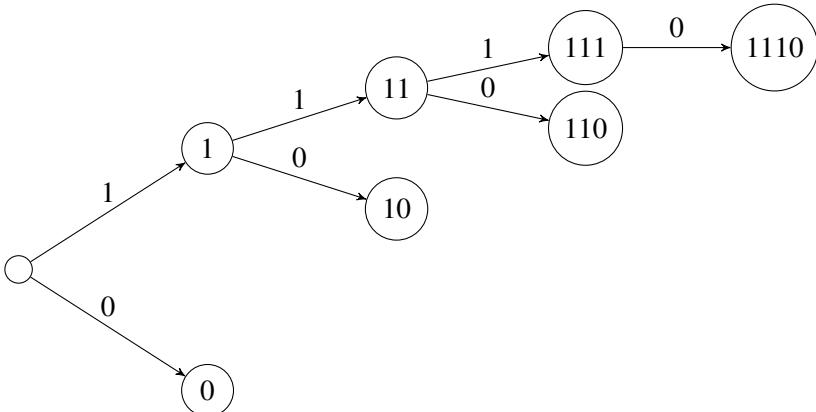


Figure D.2: Prefix-free Tree

but at level  $l_i$  we have to prune  $2^{l_{\max} - l_i}$  leaves, since the code is prefix-free. Summing over all the code words' lengths, we have that the total number of pruned leaves must be less than or equal to the maximum number of leaves, that is

$$\sum_{l_i \in \mathcal{L}} 2^{l_{\max} - l_i} \leq 2^{l_{\max}}$$

or, equivalently,

$$\sum_{l_i \in \mathcal{L}} 2^{-l_i} \leq 1$$

which is exactly the inequality we are trying to prove.

Conversely, given any set of code words' lengths  $\mathcal{L} = \{l_1, l_2, \dots, l_q\}$  that satisfy the Kraft inequality, we can always construct a binary tree, like the one in the Figure D.2. Label the first node (lexicographically) of depth  $l_1$  as code word 1, and remove its descendants from the tree. Then label the first remaining node of depth  $l_2$  as code word 2, and so on. Proceeding this way, we construct a prefix code with the specified lengths. ■

Given a code  $C$  whose code word lengths  $\mathcal{L}$  satisfy the Kraft inequality does not necessarily mean that the code is prefix-free, since what the inequality states is that there exists a prefix-free code with those word lengths, not that all codes with those word lengths are prefix-free.

■ **Example D.6** The code  $C(a) = 0, C(b) = 111, C(c) = 110$  and  $C(d) = 100$  satisfies the Kraft inequality, but it is not prefix-free. ■

McMillan's inequality extends Kraft's bound to all uniquely decodable codes: any such code with lengths  $l_i$  satisfies  $\sum_i 2^{-l_i} \leq 1$ . Combined with Kraft's theorem, this means a set of lengths meets this inequality if and only if it can be realized by a prefix-free—and thus uniquely decodable—code.

**Theorem D.2.2 — McMillan's Inequality.** Let  $\mathcal{L} = \{l_1, l_2, \dots, l_q\}$  be a set of lengths, where  $l_i \in \mathbb{N}$ , then there exists a uniquely decodable code  $C$  whose code words have the lengths of  $\mathcal{L}$  if, and only if,

$$\sum_{l_i \in \mathcal{L}} 2^{-l_i} \leq 1$$

*Proof.* Let  $S = \sum_{i=1}^q 2^{-l_i}$ . For each  $n \in \mathbb{N}$ , let  $C^n$  be the set of all concatenations of  $n$  code words. Because the code is uniquely decodable, distinct concatenations in  $C^n$  produce distinct binary strings. Consider an infinite fair-coin binary sequence  $X = (X_1, X_2, \dots)$ , and for any finite binary string  $u$  write  $[u] = \{X : X \text{ begins with } u\}$ . Then  $\mathbb{P}([u]) = 2^{-|u|}$ , and the cylinder sets  $\{[w] : w \in C^n\}$  are pairwise disjoint. Therefore,

$$\sum_{w \in C^n} 2^{-|w|} = \sum_{w \in C^n} \mathbb{P}([w]) \leq 1.$$

But the left-hand side factors as

$$\sum_{w \in C^n} 2^{-|w|} = \left( \sum_{i=1}^q 2^{-l_i} \right)^n = S^n.$$

Hence  $S^n \leq 1$  for all  $n$ , which implies  $S \leq 1$ . ■

In the theory of nescience, we focus on prefix-free codes without loss of generality: every uniquely decodable code satisfies Kraft's inequality, so there exists a prefix-free code with the same code-word lengths.

**Corollary D.2.3** There is an instantaneous (prefix-free) code with word lengths  $l_1, \dots, l_q$  if and only if there is a uniquely decodable code with these word lengths.

*Proof.* Every prefix-free code is uniquely decodable.

If a uniquely decodable code exists with lengths  $l_1, \dots, l_q$ , then by McMillan's inequality  $\sum_i 2^{-l_i} \leq 1$ . By Theorem D.2.1 (sufficiency), there exists a prefix-free code with the same lengths. ■

In the context of nescience, our interest lies not in particular code assignments but in the multiset of code word lengths. This emphasis lets us abstract away from code construction details and focus on the mathematical properties of lengths, which are central to the measures we will use.

### D.3 Optimal Codes

In coding theory, a compact code is an optimal encoding strategy that minimizes the expected length of code words, a concept central to evaluating a code's efficiency. The expected length of a code is determined by the weighted average of the lengths of its code words, with weights corresponding to the probability distribution of the source symbols. By designing code words so that more frequent symbols are assigned shorter lengths and less frequent ones longer lengths, compact codes effectively reduce the average size needed to encode information. This principle is pivotal in data compression, as it allows for a significant reduction in the space required for storage or the bandwidth needed for transmission.

Let  $\mathcal{S} = \{s_1, \dots, s_q\}$  be a finite source alphabet, and let  $P$  be a probability distribution on  $\mathcal{S}$ .

**Definition D.3.1** The *expected length* of a code  $C$ , denoted by  $L_C$ , is

$$L_C = \sum_{i=1}^q P(s_i) l_i,$$

where  $\mathcal{L} = \{l_1, \dots, l_q\}$  are the code-word lengths of  $C$ . We may simply write  $L$  when  $C$  is clear from context.

Our goal is to identify a code  $C$  that minimizes  $L$  under  $P$ . Such a code is optimal for  $\mathcal{S}$  and  $P$ .

Compact codes achieve the lowest possible weighted average codeword length.

**Definition D.3.2** A code  $C$  is *compact* (optimal) if, for the given source alphabet  $\mathcal{S}$  and distribution  $P$ , its expected length  $L_C$  is minimal among all codes over the same code alphabet.

The existence of compact codes for all possible source alphabets is a foundational aspect of coding theory, suggesting that for every finite source alphabet, an optimally efficient encoding scheme can be devised.

**Proposition D.3.1** For every finite source alphabet  $\mathcal{S}$  and distribution  $P$  there exists at least one code  $C$  that is compact.

*Proof.* Huffman's algorithm (see Section D.5) constructs a prefix-free code that minimizes the expected length for the given  $P$ . Hence an optimal code exists. ■

It is convenient to distinguish between *coding efficiency* and *redundancy* relative to the source entropy. Let  $H_r(P) = -\sum_i P(s_i) \log_r P(s_i)$  be the  $r$ -ary entropy (logarithms base  $r$ , where  $r$  is the code alphabet size).

**Definition D.3.3** The *coding efficiency* is

$$\eta = \frac{H_r(P)}{L} \in (0, 1],$$

and the (normalized) *redundancy* is

$$\rho = 1 - \eta = \frac{L - H_r(P)}{L} \in [0, 1].$$

Our aim is to *maximize* efficiency (equivalently, *minimize* redundancy). Note that  $L \geq H_r(P)$  (Theorem D.3.2 below), so  $\eta \leq 1$ .

Next theorem states that the entropy of the probability distribution  $P$  poses a limit to the average length of prefix-free codes.

**Theorem D.3.2** For any prefix-free  $r$ -ary code with lengths  $\{l_i\}$  and source distribution  $P$ ,

$$H_r(P) \leq L = \sum_{i=1}^q P(s_i)l_i,$$

with equality if and only if  $r^{-l_i} = P(s_i)$  for all  $i$ .

*Proof.* Let  $\alpha = \sum_i r^{-l_i} \leq 1$  by Kraft's inequality. Define a probability distribution  $\tilde{Q}$  by

$$\tilde{Q}(s_i) = \frac{r^{-l_i}}{\alpha}.$$

Then

$$\begin{aligned} L - H_r(P) &= \sum_i P(s_i) \left( l_i + \log_r P(s_i) \right) \\ &= \sum_i P(s_i) \log_r \left( \frac{r^{l_i} P(s_i)}{1} \right) = \sum_i P(s_i) \log_r \left( \frac{P(s_i)}{r^{-l_i}} \right). \end{aligned}$$

Write  $r^{-l_i} = \alpha \tilde{Q}(s_i)$  to get

$$\begin{aligned} L - H_r(P) &= \sum_i P(s_i) \log_r \left( \frac{P(s_i)}{\tilde{Q}(s_i)} \right) + \log_r \left( \frac{1}{\alpha} \right) \\ &= D_r(P \parallel \tilde{Q}) + \log_r(1/\alpha) \geq 0, \end{aligned}$$

since the  $r$ -ary Kullback–Leibler divergence  $D_r$  is nonnegative and  $\alpha \leq 1$ . Equality holds iff both terms vanish, i.e.,  $\alpha = 1$  and  $P = \tilde{Q}$ , which is equivalent to  $r^{-l_i} = P(s_i)$  for all  $i$ . ■

In data compression we are particularly interested in  $r$ -adic distributions.

**Definition D.3.4** A distribution  $P$  is called *r-adic* if each  $P(s_i)$  is an exact  $r$ -adic probability, i.e.,  $P(s_i) = r^{-n_i}$  for some integers  $n_i \geq 0$ .

Equality in Theorem D.3.2 occurs precisely for sources whose symbol probabilities are exact powers of  $r^{-1}$ .

**Corollary D.3.3** Equality  $L = H_r(P)$  in Theorem D.3.2 holds if and only if  $P$  is  $r$ -adic (equivalently, there exists a prefix-free code with  $l_i = -\log_r P(s_i)$ ).

*Proof.* Theorem D.3.2 shows equality holds exactly when  $r^{-l_i} = P(s_i)$  for all  $i$ , i.e., when each  $P(s_i)$  is an  $r$ -adic probability. ■

In analysis it is natural to consider the *ideal* real-valued lengths  $\ell_i^* = -\log_r P(s_i)$ , which need not be integers. Choosing integer lengths  $l_i = \lceil \ell_i^* \rceil$  yields a feasible prefix-free code (by Kraft) with near-optimal average length.

**Proposition D.3.4 — Shannon–Fano bound.** For  $l_i = \lceil -\log_r P(s_i) \rceil$ ,

$$H_r(P) \leq L < H_r(P) + 1.$$

*Proof.* By definition  $l_i < -\log_r P(s_i) + 1$ , hence

$$L = \sum_i P(s_i) l_i < \sum_i P(s_i) (-\log_r P(s_i) + 1) = H_r(P) + 1.$$

Moreover,  $\sum_i r^{-l_i} \leq \sum_i r^{\log_r P(s_i)} = \sum_i P(s_i) = 1$ , so the lengths are feasible. The lower bound  $L \geq H_r(P)$  is Theorem D.3.2. ■

Kraft's inequality and expected length allow us to compare codes for a fixed source.

**Definition D.3.5** Let  $C_1, C_2$  be two codes on  $\mathcal{S}$  with expected lengths  $L_{C_1}, L_{C_2}$  under  $P$ . We say that  $C_1$  is *more efficient* than  $C_2$  (for  $P$ ) if  $L_{C_1} \leq L_{C_2}$ , with strict inequality for at least one symbol probability configuration (equivalently,  $L_{C_1} < L_{C_2}$  for the given  $P$ ).

■ **Example D.7** The non-prefix-free code in Example D.6 has lengths  $(1, 3, 3, 3)$ , which satisfy Kraft's inequality. Therefore, there exists a prefix-free code with the *same* lengths; for instance,

$$C(a) = 1, \quad C(b) = 000, \quad C(c) = 001, \quad C(d) = 010.$$

For many  $P$ , these lengths yield a smaller expected length than the prefix-free code of Example D.4 (which has lengths  $(1, 2, 3, 4)$ ); the comparison is determined by  $P$  via  $L = \sum_i P(s_i) l_i$ . ■

Intuitively, a prefix-free code is *complete* if its set of code words cannot be augmented (by adding another code word) while preserving the prefix-free property.

**Definition D.3.6** A prefix-free code  $C$  is *complete* if there does not exist a binary string  $w$  such that  $C \cup \{w\}$  is still prefix-free.

Kraft's inequality characterizes completeness.

**Proposition D.3.5** A prefix-free code  $C$  with lengths  $\mathcal{L} = \{l_1, \dots, l_q\}$  is complete if and only if

$$\sum_{i=1}^q 2^{-l_i} = 1.$$

*Proof.* If the sum is  $< 1$ , there remains unused capacity in the binary tree at some depth, so one can add at least one more leaf without violating prefix-freeness, contradicting completeness. Conversely, if the sum equals 1, the descendant subtrees of the selected leaves exactly partition the leaf set at some depth; no additional leaf can be added without creating a prefix conflict. ■

## D.4 Entropy

In this section we introduce *entropy* as a measure of the uncertainty of a discrete random variable. Entropy appears in many contexts (communications, statistics, finance, etc.); here it is important because it will allow us to identify codes with the shortest possible average length.

Let  $A = \{a_1, a_2, \dots, a_n\}$  be a finite set, and let  $X$  be a random variable on  $A$  with probability mass function  $p(a)$ .

**Definition D.4.1 — Entropy.** The *entropy* of  $X$ , denoted  $H(X)$  and measured in *bits*, is

$$H(X) = \sum_{a \in A} p(a) \log \frac{1}{p(a)}.$$

Entropy depends only on the probabilities  $\{p(a)\}$ , not on the labels in  $A$ . Clearly  $H(X) \geq 0$  since  $0 \leq p(a) \leq 1$  implies  $-\log p(a) \geq 0$ . If  $p(a_i) = 0$  for some  $i$ , the corresponding summand is taken to be 0, consistent with  $\lim_{p \rightarrow 0^+} p \log(1/p) = 0$ . If we change the logarithm base to  $u$ , entropy scales by a factor  $\log_u 2$  (see Equation ??).

■ **Example D.8** Let  $X$  a random variable defined over the set  $A = \{a_1, a_2\}$ ,

with values  $p(a_1) = q$  and  $p(a_2) = 1 - q$ . Then, the entropy of  $X$  is given by:

$$H(X) = q \log \frac{1}{q} + (1 - q) \log \frac{1}{1 - q}$$

Figure D.3 shows the entropy of  $X$  for different values of  $q$ . If  $q = 0$  or  $q = 1$  the entropy is 0, that is, there is no uncertainty about which value of  $A$  we will get. The maximum value of  $H$  is 1, and it is reached when  $q = 1/2$ ; that is, we could say that 1 bit is the uncertainty associated to two equally probable symbols. ■

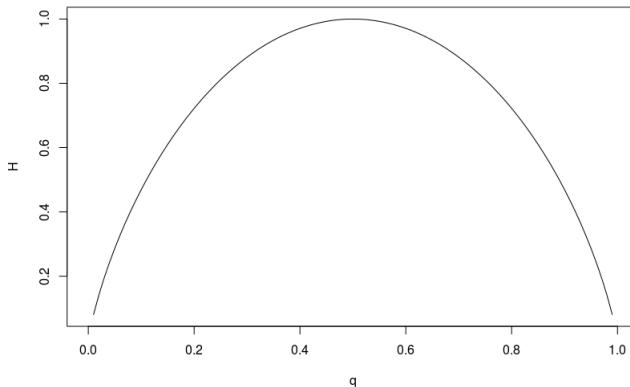


Figure D.3: Binary Entropy Function

The next proposition shows that entropy is maximized by the uniform distribution and that the maximum equals the logarithm of the alphabet size.

**Proposition D.4.1** For  $X$  on  $A$  with  $d(A) = n$ , we have  $H(X) \leq \log n$ , with equality if and only if  $p(a_1) = \dots = p(a_n) = 1/n$ .

*Proof.* Compute

$$\log n - H(X) = \sum_{i=1}^n p(a_i) \log n - \sum_{i=1}^n p(a_i) \log \frac{1}{p(a_i)} = \sum_{i=1}^n p(a_i) \log(n p(a_i)).$$

Using the change-of-base formula,

$$\log n - H(X) = (\log e) \sum_{i=1}^n p(a_i) \ln(n p(a_i)).$$

By the inequality  $\ln x \leq x - 1$ , applied with  $x = \frac{1}{n p(a_i)}$ , we have

$$\ln(n p(a_i)) \geq 1 - \frac{1}{n p(a_i)}.$$

Hence

$$\log n - H(X) \geq (\log e) \sum_{i=1}^n p(a_i) \left( 1 - \frac{1}{n p(a_i)} \right) = (\log e) \left( 1 - \frac{1}{n} \sum_{i=1}^n 1 \right) = 0.$$

Equality holds iff  $\ln(n p(a_i)) = 0$  for all  $i$ , i.e.,  $p(a_i) = 1/n$ . ■

■ **Example D.9** If we select a symbol from  $A$  according to  $p$ , the entropy  $H(X)$  is the minimum expected number of binary (Yes/No) questions required to identify the symbol. For equiprobable symbols this expected number is maximal and equals  $\log |A|$ . ■

We can extend the concept of entropy to a pair of random variables by means of using the joint probability mass function. In this way, the joint entropy will be a measure of the uncertainty associated to both variables. Let  $A = \{a_1, a_2, \dots, a_n\}$  and  $B = \{b_1, b_2, \dots, b_m\}$  two finite sets, and  $X$  and  $Y$  two random variables defined over the sets  $A$  and  $B$  respectively, with probability mass function  $p(a)$  and  $p(b)$ , and joint probability mass function  $p(a, b)$ .

**Definition D.4.2** The *joint entropy* of the random variables  $A$  and  $B$ , denoted by  $H(A, B)$ , is defined as:

$$H(A, B) = \sum_{a \in A} \sum_{b \in B} p(a, b) \log \frac{1}{p(a, b)}$$

Joint entropy is symmetric:  $H(A, B) = H(B, A)$ , since the double sum is unchanged by swapping  $a$  and  $b$ . The definition extends to any finite tuple  $(A_1, \dots, A_k)$  via the joint mass function  $p(a_1, \dots, a_k)$ .

Adding a second random variable with an unknown outcome may increase entropy, as the following proposition shows.

**Proposition D.4.2** We have that

$$H(A, B) \geq \max(H(A), H(B))$$

*Proof.* Because  $\sum_b p(a, b) = p(a)$ , we have  $p(a, b) \leq p(a)$  for all  $a, b$ , hence  $\log \frac{1}{p(a, b)} \geq \log \frac{1}{p(a)}$ . Therefore,

$$\begin{aligned} H(A, B) &= \sum_{a,b} p(a, b) \log \frac{1}{p(a, b)} \geq \sum_{a,b} p(a, b) \log \frac{1}{p(a)} \\ &= \sum_a p(a) \log \frac{1}{p(a)} = H(X). \end{aligned}$$

The inequality  $H(A, B) \geq H(B)$  is analogous. Taking the maximum yields the claim. ■

The joint entropy of two variables cannot exceed the sum of their individual entropies.

**Proposition D.4.3** We have  $H(A, B) \leq H(A) + H(B)$ , with equality if and only if  $A$  and  $B$  are independent (i.e.,  $p(a, b) = p(a)p(b)$  for all  $a, b$ ).

*Proof.* By Gibbs' inequality (which follows from  $\ln x \leq x - 1$ ),

$$\begin{aligned} 0 \leq \sum_{a,b} p(a, b) \log \frac{p(a, b)}{p(a)p(b)} &= \sum_{a,b} p(a, b) \log p(a, b) \\ &\quad - \sum_{a,b} p(a, b) \log p(a) - \sum_{a,b} p(a, b) \log p(b). \end{aligned}$$

Rearranging and recognizing the entropies,

$$0 \leq -H(A, B) + H(A) + H(B),$$

which gives  $H(A, B) \leq H(A) + H(B)$ . Equality in Gibbs' inequality holds if and only if  $p(a, b) = p(a)p(b)$  for all  $a, b$ , i.e., when  $A$  and  $B$  are independent. ■

The next concept derived from entropy is conditional entropy. It measures the remaining uncertainty about one random variable once the outcome of another is known.

**Definition D.4.3** The *conditional entropy* of the random variable  $B$  given the random variable  $A$ , denoted by  $H(B | A)$ , is defined as:

$$H(B | A) = \sum_{a \in A} \sum_{b \in B} p(a, b) \log \frac{1}{p(b | a)}$$

In general  $p(b | a) \neq p(a | b)$ , so  $H(B | A) \neq H(A | B)$ . We have  $H(B | A) = 0$  if and only if  $B$  is (almost surely) a function of  $A$ .

The next proposition formalizes that conditioning cannot increase uncertainty.

**Proposition D.4.4** Given the random variables  $X$  and  $Y$ , we have that  $H(Y | X) \leq H(Y)$ , and  $H(Y | X) = H(Y)$  if, and only if,  $p(a)$  and  $p(b)$  are independent.

*Proof.*

$$\begin{aligned} H(Y | X) &= \sum_{a \in A} \sum_{y \in B} p(a, y) \log \frac{1}{p(y | a)} = \sum_{a \in A} \sum_{b \in B} p(a, b) \log \frac{p(a)}{p(a, b)} \\ &= \sum_{a \in A} \sum_{b \in B} p(a, b) \log p(a) - \sum_{a \in A} \sum_{b \in B} p(a, b) \log p(a, b) \\ &= -H(X) + H(X, Y) \end{aligned}$$

By Proposition D.4.3,  $H(X, Y) \leq H(X) + H(Y)$ , hence  $H(Y | X) \leq H(Y)$ . Equality holds if and only if  $H(X, Y) = H(X) + H(Y)$ , i.e.,  $X$  and  $Y$  are independent. ■

From an intuitive standpoint, the uncertainty of a pair  $(X, Y)$  should equal the uncertainty of  $X$  plus the remaining uncertainty of  $Y$  after observing  $X$ . This is the *chain rule*.

**Proposition D.4.5 — Chain rule.** For any discrete  $X, Y$ ,  $H(X, Y) = H(X) + H(Y | X)$ .

*Proof.*

$$\begin{aligned} H(Y, X) &= \sum_{a \in A} \sum_{b \in B} p(a, b) \log \frac{1}{p(a, b)} = \sum_{a \in A} \sum_{b \in B} p(a, b) \log \frac{1}{p(a)p(a | b)} \\ &= \sum_{a \in A} \sum_{b \in B} p(a, b) \log \frac{1}{p(a)} + \sum_{a \in A} \sum_{b \in B} p(a, b) \log \frac{1}{p(a | b)} \\ &= H(X) + H(Y | X) \end{aligned}$$

■

The last derived concept of entropy we are going to see is mutual information. Intuitively, the mutual information of two random variables  $X$  and  $Y$  measures the information that  $X$  and  $Y$  share, that is, how much knowing one of these variables reduces the uncertainty about the other.

**Definition D.4.4** The *mutual information* of the random variable  $X$  and  $Y$ , denoted by  $I(X; Y)$ , is defined as:

$$I(X; Y) = \sum_{a \in A} \sum_{b \in B} p(a, b) \log \frac{p(a, b)}{p(a)p(b)}$$

Since  $p(a, b) = p(b, a)$  we have that  $I(X; Y) = I(Y; X)$ , that is, the order of the random variables does not affect the concept of mutual information.

Next proposition shows that mutual information is a positive quantity, and it is equal to 0 if, and only if, the random variables are independent.

**Proposition D.4.6** For any discrete random variables  $X$  and  $Y$  we have that  $I(X; Y) \geq 0$ , and  $I(X; Y) = 0$  if, and only if, the variables  $X$  and  $Y$  are independent.

*Proof.* Interpret  $I(X; Y)$  as the Kullback-Leibler divergence

$$I(X; Y) = D(p_{X,Y} \| p_X p_Y) = \sum_{a,b} p(a, b) \log \frac{p(a, b)}{p(a)p(b)} \geq 0$$

by Gibbs' inequality, with equality if and only if  $p(a,b) = p(a)p(b)$  for all  $a,b$ .  $\blacksquare$

Mutual information admits several equivalent forms that make its operational meaning explicit: it quantifies the reduction in uncertainty achieved by observing one variable about the other. Using the chain rule  $H(X,Y) = H(X) + H(Y | X) = H(Y) + H(X | Y)$ , we obtain the identities below, showing that  $I(X;Y)$  is exactly the decrease in entropy of  $X$  due to  $Y$  (and symmetrically for  $Y$ ).

**Proposition D.4.7** For any discrete random variables  $X$  and  $Y$ , we have that:

$$I(X;Y) = H(X) - H(X | Y) = H(Y) - H(Y | X)$$

*Proof.* By the chain rule,  $H(X,Y) = H(Y) + H(X | Y)$ . Substituting into

$$I(X;Y) = \sum_{a,b} p(a,b) \log \frac{p(a,b)}{p(a)p(b)} = -H(X,Y) + H(X) + H(Y),$$

we obtain  $I(X;Y) = H(X) - H(X | Y)$ . The other identity follows by symmetry.  $\blacksquare$

Another equivalent form of mutual information is its “inclusion-exclusion” expression: the sum of the marginal entropies minus the joint entropy. It follows immediately by substituting  $H(X | Y) = H(X,Y) - H(Y)$  (or symmetrically  $H(Y | X) = H(X,Y) - H(X)$ ) into  $I(X;Y) = H(X) - H(X | Y)$ , and highlights the Venn-diagram overlap interpretation.

**Proposition D.4.8** For any discrete random variables  $X$  and  $Y$ , we have that:

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

*Proof.* Combine  $I(X;Y) = H(X) - H(X | Y)$  with  $H(X | Y) = H(X,Y) - H(Y)$  (chain rule), yielding  $I(X;Y) = H(X) + H(Y) - H(X,Y)$ .  $\blacksquare$

A useful sanity check is the self-information case. Since observing  $S$  reveals  $S$  completely, the remaining uncertainty  $H(S | S)$  is zero; hence the mutual information of a variable with itself equals its entropy, as stated below.

**Proposition D.4.9** For any discrete random variable  $S$  we have that  $I(S;S) = H(S)$ .

*Proof.* By the identity  $I(X;Y) = H(X) - H(X | Y)$  with  $X = Y = S$ ,

$$I(S;S) = H(S) - H(S | S) = H(S) - 0 = H(S).$$

$\blacksquare$

■ **Example D.10** Let  $X \sim \text{Bernoulli}(1/2)$  and let  $Y = X \oplus N$ , where  $N \sim \text{Bernoulli}(\varepsilon)$  is independent noise ( $\oplus$  is XOR). Then  $H(X) = H(Y) = 1$ , and

$$H(X | Y) = H_b(\varepsilon) := -\varepsilon \log \varepsilon - (1 - \varepsilon) \log(1 - \varepsilon), \quad I(X; Y) = 1 - H_b(\varepsilon).$$

For instance, with  $\varepsilon = 0.1$ ,  $H_b(0.1) \approx 0.469$ , so  $I(X; Y) \approx 0.531$  bits. A Venn diagram interpretation: the circle areas represent  $H(X)$  and  $H(Y)$ , the overlap is  $I(X; Y)$ , and the non-overlapping parts are  $H(X | Y)$  and  $H(Y | X)$ .

■

Figure ?? provides Venn-style schematic for entropies, conditional entropies and mutual information.

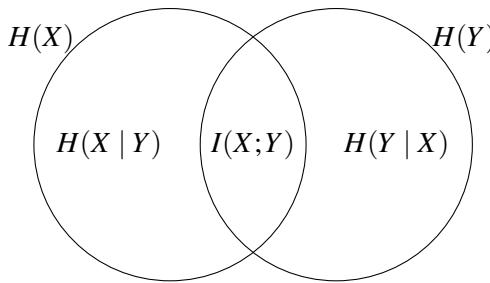


Figure D.4:  $H(X)$  and  $H(Y)$  with overlap  $I(X; Y)$ .

## D.5 Huffman Algorithm

From a practical point of view, there exists an algorithm, called *Huffman algorithm*, that provides a method to build compact prefix-free codes given a probability distribution. For simplicity, we will study first the particular case of constructing binary prefix-free codes, and later I will provide its generalization to the case of D-ary prefix-free codes.

The algorithm (see Algorithm D.1) expects as input a source alphabet  $\mathcal{S} = \{s_1, s_2, \dots, s_q\}$  and their corresponding probabilities  $P = \{p_1, p_2, \dots, p_q\}$ . For simplicity,  $Q = \{(s_1, p_1), (s_2, p_2), \dots, (s_q, p_q)\}$  will merge both sets into a single one. The algorithm works by constructing a binary tree  $T$ , similar to the one used in the proof of Theorem D.2.1. The algorithm requires  $d(Q) - 1$  iterations to finish. During each iteration, the two elements with the lowest probability are selected and removed from set  $Q$ , and a new tree node  $z$  is created, with the addition of the removed values, and added to the set  $Q$ . Once the tree has been constructed, we have to perform a tree transversal assigning a 0 to each left branch, and a 1 to each right branch, until we reach a leaf.

**Algorithm D.1** Huffman Algorithm

---

```

procedure HUFFMAN( $Q$ )
   $T \leftarrow$  empty tree
  for  $i \leftarrow 1, d(Q) - 1$  do
    allocate a new node  $z$ 
     $z.\text{left} = x = \text{EXTRACT-MIN}(Q)$ 
     $z.\text{right} = y = \text{EXTRACT-MIN}(Q)$ 
     $z.\text{freq} = x.\text{freq} + y.\text{freq}$ 
     $\text{INSERT}(Q, z)$ 
  end for
  return  $T$ 
end procedure

```

---

■ **Example D.11** Assume we have the source alphabet  $S = \{a, b, c, d, e, f\}$  with the associated probabilities  $P = \{0.35, 0.16, 0.08, 0.12, 0.06, 0.23\}$ . In Figure are depicted the contents of the set  $Q$  and the tree  $T$  for each iteration of the algorithm. At the end of the algorithm, if we perform a traversal of the  $T$  tree, we will get the following prefix-free compact code for the source alphabet  $S$ :

Source Word	Code Word
a	11
b	00
c	1011
d	100
e	1010
f	01

## D.6 Discretization of Continuous Variables

When summarizing large masses of raw data, it is often useful to distribute the data into classes, or categories, and to determine the number of individuals belonging to each class, called absolute frequency. The following definition formally introduces this concept.

**Definition D.6.1** Let  $S$  be a population consisting of  $n$  individuals, and let a variable  $X : S \rightarrow \mathcal{D}$  represent the mapping of individuals in  $S$  to values in the set  $\mathcal{D}$ , where  $k$  is the cardinality of  $\mathcal{D}$ . The *absolute frequency*, also

known simply as *frequency*, denoted by  $n_i$  for  $1 \leq i \leq k$ , quantifies the number of individuals in  $\mathcal{S}$  for which  $X$  assigns the value corresponding to the  $i$ -th category of  $\mathcal{D}$ .

The sum of the frequencies must be equal to population size, that is,  $\sum_{i=1}^k n_i = n$ .

If the variable  $X$  is continuous, the elements of  $\mathcal{D}$  are referred to as *class intervals*. The endpoints of these intervals are known as *class limits*; the smaller number is termed the *lower class limit*, and the larger number, the *upper class limit*. A *class interval* that lacks either an upper or a lower class limit is known as an *open class interval*. The *width* of a class interval is defined as the difference between the upper and lower class boundaries, denoted by  $a_i = e_i - e_{i-1}$ . The *class mark*, or the midpoint of a class interval, is calculated as  $a_i = \frac{e_i + e_{i-1}}{2}$ . It is assumed that all observations within a specific class interval are equivalent concerning their categorical assignment. Refer to Section ?? for more information about how to discretize a continuous variable into discrete intervals.

■ **Example D.12** In a study measuring adult heights within a community, researchers record heights ranging from 150 cm to 200 cm and organize them into 10 cm class intervals: 150-159 cm, 160-169 cm, 170-179 cm, 180-189 cm, and 190-199 cm. Each interval's lower and upper class limits are respectively the start and end points, such as 150 cm and 159 cm for the first interval. If the last interval had no specified upper limit, it would be considered an open class interval. The class width, typically 10 cm, is the operational span between boundaries, and the class mark, calculated as the midpoint of each interval (e.g., 154.5 cm for the first interval), provides a central value for summarizing data within that range. ■

A tabular arrangement of data by classes together with the corresponding class frequencies is called a *frequency distribution*, or *frequency table*.

**Definition D.6.2** Let  $\mathcal{S}$  be a population consisting of  $n$  individuals, and let  $X : \mathcal{S} \rightarrow \mathcal{D}$  be a variable that maps individuals in  $\mathcal{S}$  to  $k$  distinct values of the set  $\mathcal{D}$ . A *frequency distribution* is represented the set of pairs  $\{(d_i, n_i) : 1 \leq i \leq k\}$ , where  $d_i$  denotes the  $i$ -th interval and  $n_i$  represents the number of individuals from  $\mathcal{S}$  whose value under  $X$  falls within the interval  $d_i$ .

Frequency distributions are useful for statistical analysis and helps in visualizing data by grouping values, which simplifies the understanding of distribution and central tendencies within the data.

The relative frequency of a class is the frequency of the class divided by the total frequency of all classes

**Definition D.6.3** Let  $\mathcal{S}$  be a population consisting of  $n$  individuals, and let a variable  $X : \mathcal{S} \rightarrow \mathcal{D}$  represent the mapping of individuals in  $\mathcal{S}$  to values in the set  $\mathcal{D}$ , where  $k$  is the cardinality of  $\mathcal{D}$ . The *relative frequency*, denoted by  $f_i$ , is the ratio  $f_i = \frac{n_i}{n}$  for  $1 \leq i \leq k$ .

The sum of the relative frequencies is equal to one, that is,  $\sum_{i=1}^k f_i = 1$ .

The total frequency of all values less than the upper class boundary of a given class interval is called the cumulative frequency up to and including that class interval.

**Definition D.6.4** Let  $\mathcal{S}$  be a population consisting of  $n$  individuals, and let a variable  $X : \mathcal{S} \rightarrow \mathcal{D}$  represent the mapping of individuals in  $\mathcal{S}$  to values in the set  $\mathcal{D}$ , where  $k$  is the cardinality of  $\mathcal{D}$ . The *cumulative frequency*, denoted by  $N_i$  for  $1 \leq i \leq k$ , represents the total number of individuals in  $\mathcal{S}$  for which  $X$  assigns a value less than or equal to the upper limit of the  $i$ -th category of  $\mathcal{D}$ . This is mathematically expressed as  $N_i = \sum_{j=1}^i n_j$ , where  $n_j$  is the absolute frequency of the  $j$ -th category.

By accumulating the frequencies up to each category or interval, cumulative frequencies provides a running total that shows how many data points fall below a certain value.

### Discretization Algorithms

Let  $\mathcal{X}$  a continuous random variable that follows a probability density function  $P_{\mathcal{X}}$ , and assume we have collected  $n$  independent and identically distributed samples  $\mathbf{x} = \{x_1, \dots, x_n\}$  from  $\mathcal{X}$ . We are interested in computing the length of a compressed version of  $\mathbf{x}$  using an optimal compressor. Unfortunately, and except for some degenerate distributions, there is no lossless compression algorithm that produces a string with fewer bits than encoding directly the elements  $\mathbf{x}$ . Compression algorithms for continuous data only work in case that the elements of  $\mathbf{x}$  are not independent, as it is the case with images or sound. But, if this is not the case, the only option available to compress  $\mathbf{x}$  is to use a lossy compression algorithm, where some information is lost.

We are looking for an algorithm to produce a finite non-overlapping partition of  $m$  discrete intervals  $\mathcal{D} = \{[d_0, d_1], (d_1, d_2], \dots, (d_{m-1}, d_m]\}$ , where  $d_0 = \min \mathbf{x}_j$ , and  $d_m = \max \mathbf{x}_j$ , and  $d_i < d_{i+1}$  for  $i = 0, 1, \dots, m - 1$ , assign a unique label to each interval, and encode the elements of  $\mathbf{x}$  using this labeling schema. As compression algorithm we will use an optimal length code given the relative frequencies of the labels in the encoded vector. In this sense, our goal is to have a collection of intervals with sufficiently number of samples (so they are statistically significant) and that the distribution of frequencies resembles the original probability distribution  $P_{\mathcal{X}}$ .

A discretization algorithm is a mapping between a (possibly huge) number of numeric values and a reduced set of discrete values, and so, it is a process in which some information is potentially lost. The choice of discretization algorithm is something that could have a high impact in the practical computation of the nescience. We are interested in a discretization algorithm that produces a large number of intervals (low bias), with a large number of number of observations per interval (low variance). Common techniques include *equal width discretization*, *equal frequency discretization* and *fixed frequency discretization*. However, these techniques require the optimization of an hyperparameter, and so, they are not suitable for our purposes.

In a *proportional discretization approach* the number of intervals  $m$  and the number of observations per interval  $s$  are equally proportional to the number of observations  $n$ . The algorithm starts by sorting the values of  $\mathbf{x}_j$  in ascending order and then discretizing them into  $m$  intervals of approximately  $s$  (possibly identical) values each. In this way, as the number of training observations increases, both interval frequency and number of intervals increases, taking advantage of the larger number of observations. In the same way, when the number of observations decreases, we reduce both.

## References

Core references for the chapter "Coding"

[CT12]: Cover and Thomas is the standard reference for modern information theory. It introduces entropy, mutual information, source and channel coding theorems, and provides the theoretical basis for coding.

[Abr63]: Abramson's book is a classic early text that presents coding theory in an accessible way. It is useful for historical perspective and for understanding the development of the main coding ideas.

[Kra49]: Kraft's thesis introduces the Kraft inequality, a cornerstone result for the theory of uniquely decodable prefix codes.

[McM56]: McMillan extends Kraft's result to a more general form, establishing the Kraft-McMillan inequality fundamental to coding theory.

[GG12]: Gersho and Gray provide the standard reference on quantization methods used in lossy compression, important for bridging coding theory with practical applications.

[Llo82]: Lloyd's algorithm (often rediscovered as k-means) is a central method for quantization and lossy compression. It provides a concrete link between coding and clustering.

[LV13]: Li and Vitányi's book is the definitive source on Kolmogorov complexity, linking coding theory with randomness and algorithmic infor-

mation. It is essential for connecting coding to the broader framework of nescience.



## E. Complexity

*An approximate answer to the right problem  
is worth a good deal more than  
an exact answer to an approximate problem.*

John Tukey

In Appendix D, the concept of the complexity of a string based on the lengths of the codewords of a prefix-free code was introduced. This definition is limited by two main factors: first, it necessitates prior knowledge of the set of possible strings, and second, it requires the definition of a probability distribution over this set a priori. It would be highly desirable to expand the set of strings to encompass all strings (that is,  $\mathcal{B}^*$ ) without requiring a probability distribution, thereby providing an absolute notion of string complexity. Unfortunately, even if these issues are resolved, a more fundamental limitation arises when studying the complexity of strings using codes: certain strings that we intuitively expect to be simple cannot be compressed. For instance, the binary expansion of the constant  $\pi$  is widely conjectured to behave like a uniform distribution over the set  $\{0, 1\}$  and, as such, cannot be compressed. Yet it can be fully and effectively described by a very short mathematical formula. This motivates the need for an alternative definition of string complexity.

*Kolmogorov complexity*, also known as *algorithmic information theory*, offers a definition of the complexity of a string that directly addresses these issues. Intuitively, the amount of information in a finite string is measured by the length of the shortest computer program capable of producing the string. This approach does not require prior knowledge of the set of valid strings or their probability distribution. Furthermore, objects like  $\pi$  are appropriately classified as having low complexity. We may argue that Kolmogorov complexity provides a universal definition of the amount of information that closely aligns with our intuitive understanding. To compute the Kolmogorov complexity of a string, it is necessary to fix a universal description method or computer language, together with a universal computer. One might question whether, in doing so, the complexity of a string becomes dependent on the chosen language. Fortunately, it has been shown that this is not the case: all reasonable (and sufficiently powerful) languages yield the same description length, up to a fixed constant that depends on the choice of languages but not on the string itself. Unfortunately, Kolmogorov complexity also introduces a significant challenge: it is a non-computable quantity and, as such, must be approximated in practice.

At this point, one might ask whether it is possible to define the complexity of arbitrary objects, not just strings. The answer is yes, at least in theory. Given an object  $x$ , the task is to provide an encoding method that represents the object as a string. This encoding is useful only if we can losslessly and effectively reconstruct the original object from its description. However, providing such encodings is not always feasible, either because the objects in question are abstract (as in much of mathematics) or because practical reconstruction of the object from its description is currently impossible (for example, with living organisms<sup>1</sup>).

## E.1 Strings Complexity

In Section C.1, the concept of the Turing machine, an idealized model of computation, was introduced. We saw that Turing machines can be represented as partial computable functions  $T : \mathcal{B}^* \rightarrow \mathcal{B}^*$ , which assign to each input string  $s \in \mathcal{B}^*$  an output string  $T(s) \in \mathcal{B}^*$  (Definition C.4.1). We also introduced the concept of a universal Turing machine  $U : \mathcal{B}^* \times \mathcal{B}^* \rightarrow \mathcal{B}^*$  (Definition C.2.1), a machine that can simulate the behavior of any other Turing machine; that is, for all  $(x, v) \in \mathcal{B}^* \times \mathcal{B}^*$ , we have that  $U(x, v) = T_x(v)$ . Later, in Section D.1, the concept of a code, and in particular, the notion of a prefix-free code, was introduced (Definition D.1.4). We saw that this kind of code presents important properties (Theorem D.2.1). The next definition

---

<sup>1</sup>As of now, it is not possible to recreate an animal solely based on its DNA.

merges the best of both worlds, Turing machines and prefix-free codes, and introduces a new type of universal Turing machine.

**Definition E.1.1** A *prefix-free universal Turing machine* is a universal Turing machine  $U : \mathcal{B}^* \times \mathcal{B}^* \rightarrow \mathcal{B}^*$  such that, for every  $v \in \mathcal{B}^*$ , the domain  $U_v$  is prefix-free, where  $U_v : \mathcal{B}^* \rightarrow \mathcal{B}^*$  and  $U_v(p) = U(p, v)$  for all  $p \in \mathcal{B}^*$ .

Using modern computer science terminology we could say that  $U$  is the computer,  $p$  is the program, and  $v$  is the input to the program. Intuitively, the above definition requires that no computer program can be a prefix of any other program. This is not a limitation from the point of view of string lengths, since, by applying McMillan's theorem (Theorem D.2.1), given a uniquely decodable program, we could always find a prefix-free one that computes exactly the same function and has the same length. In practice, programming languages enforce syntactic rules that make programs effectively self-delimiting (for example, programs or functions must terminate with specific delimiters).

Fixing the input  $v$  allows us to regard the set of valid programs  $\{p : U(p, v) \downarrow\}$  as prefix-free. This ensures that descriptions can be uniquely parsed and avoids ambiguity when concatenating programs.

The concept of a prefix-free universal Turing machine allows us to introduce a new definition of the complexity of a string that aligns more closely with our intuitive understanding of the amount of computational information contained in an object (encoded as a string).

**Definition E.1.2 — Kolmogorov Complexity.** Fix a prefix-free universal Turing machine  $U : \mathcal{B}^* \times \mathcal{B}^* \rightarrow \mathcal{B}^*$ . The *Kolmogorov complexity* of a string  $s \in \mathcal{B}^*$ , denoted by  $K(s)$ , is defined as:

$$K(s) = \min_{p, v \in \mathcal{B}^*} \{l(p) + l(v) : U(p, v) = s\}.$$

Intuitively, the shortest description of a string  $s$  is given by two elements: a program  $p$  (a self-delimiting program) that captures all the regular patterns of the string, and a new string  $v$  that comprises those parts of  $s$  that do not present any regularity. We have to find the optimum balance between increasing the complexity of the program, trying to grasp more regularities, or increasing the size of the non-compressible part.<sup>2</sup>

■ **Example E.1** Consider the string composed of one thousand repetitions

<sup>2</sup>In the literature, the Kolmogorov complexity of the string  $s$  is defined as  $K(s) = \min_{p \in \mathcal{B}^*} \{l(p) : U(p, \lambda) = s\}$ , that is, the length of the shortest computer program that, without any additional input, can print the string  $s$ . We prefer to use the two-part definition  $l(p) + l(v)$  because it is more in line with the requirements of the theory of nescience.

of the substring "10", that is " $\underbrace{1010\dots1010}_{1.000\text{times}}$ ". We could write the following program:

```
example(char *v) {
    for (int i=1; i<=1000; i++)
        printf("%s", v);
}
```

and then run it with:

```
example("10");
```

in order to print it. The length of the original string is 2,000 bits, but suppose the program length is approximately 480 bits (assuming that every symbol is encoded using a uniform code of 8 bits), and the input length is 2 bits. We can then conclude that the string has a low complexity. Of course, in order to compute the actual Kolmogorov complexity of the string we would need to find the shortest Turing machine that prints that string.

On the contrary, a string composed of two thousand random bits would, with overwhelming probability, have high complexity, since no program significantly shorter than the string itself can generate it. ■

As we mentioned in the preface of this chapter, Kolmogorov complexity would not be particularly useful if the complexity of strings depended on the choice of universal Turing machine. The following theorem demonstrates that this concern is unfounded, up to a constant that depends on the choice of machines, but not on the strings themselves. This establishes Kolmogorov complexity as an inherent property of strings.

**Theorem E.1.1 — Invariance theorem.** Let  $U$  and  $U'$  be two universal Turing machines. Then, there exists a constant  $C_{U,U'}$ , depending only on  $U$  and  $U'$ , such that for each string  $s \in \mathcal{B}^*$  we have:

$$K_U(s) \leq K_{U'}(s) + C_{U,U'}.$$

*Proof.* Let  $p, v$  be the shortest strings such that  $U'(p, v) = s$ . Then we can encode the pair  $(U', p)$  and simulate it on  $U$ , obtaining  $U(\langle U', p, v \rangle, \lambda) = U'(p, v) = s$ . Encoding the pair  $(U', p)$  requires a fixed description of  $U'$  plus the description of  $p$ . Thus,  $K_U(s) \leq K_{U'}(s) + C_{U,U'}$ , where  $C_{U,U'}$  is the length of an interpreter for  $U'$  on  $U$ . ■

■ **Example E.2** Consider a universal programming language, such as Java, and an alternative language, such as Python. We can write a Python interpreter in Java, that is, a Java program that takes a Python script as input

and executes it. Then, to compute the complexity of a string  $s \in \mathcal{B}^*$  using Java,  $C_J(s)$ , it would be no greater than the complexity of the string using Python,  $C_P(s)$ , plus the length of the Python interpreter written in Java,  $C_{J,P}$ . Importantly, the length of the interpreter,  $C_{J,P}$ , does not depend on the string  $s$ . ■

Although we have proved that Kolmogorov complexity does not depend on the selected universal Turing machine, the size of the constant  $C_{U,U'}$  could pose a limitation in practical applications, especially when computing the complexity of short strings where the constant might significantly exceed the complexity of the string itself. This challenge is addressed by the Minimum Description Length principle, as described in Section F.5.

**Notation E.1.** *We denote by  $s^*$  the shortest program that outputs the string  $s$  on the universal Turing machine  $U$ , that is,  $s^* = \langle p, v \rangle$ ,  $U(s^*) = s$ , and  $l(s^*) = K(s)$ . If more than one program satisfies these properties, we select the first one using a lexicographical order induced by  $0 < 1$ .*

The size of the constant  $C_{U,U'}$  is not the only challenge presented by Kolmogorov complexity; another issue is its non-computability, that is, there is no algorithm capable of determining the shortest program that generates an arbitrary string. The following theorem on the uncomputability of Kolmogorov complexity marks a pivotal insight into the intrinsic limits of complexity theory.

**Theorem E.1.2** The function  $K : \mathcal{B}^* \rightarrow \mathbb{N}$  that assigns to each string  $s$  its Kolmogorov complexity  $K(s)$  is not computable.

*Proof.* Assume, for contradiction, that  $K$  is computable. Then we could construct a function that, for any  $n$ , finds the first string  $s$  such that  $K(s) > n$ . This function would produce such an  $s$  by a program of length  $O(\log n)$ , thereby giving a description of  $s$  much shorter than  $n$ . This contradicts the definition of Kolmogorov complexity. Therefore,  $K$  is not computable. ■

If  $K$  were computable, we could also solve the Halting Problem by constructing a program that, for any input program and input, computes whether the program halts by checking if its Kolmogorov complexity is finite. Since the Halting Problem is known to be undecidable, this provides an alternative contradiction.

In practice, we approximate Kolmogorov complexity using compression algorithms, such as the Huffman algorithm described in Section D.5, or more sophisticated schemes like Lempel-Ziv, which provide practical upper bounds on  $K(s)$  relative to their model class.

## E.2 Properties of Complexity

In this section, we delve into the properties of Kolmogorov complexity. We will explore the foundational principles that govern this complexity measure, including its invariance, symmetry, and non-computability. Through examining these properties, we gain deeper insights into the interplay between information, computation, and randomness.

Kolmogorov complexity is always a finite positive natural number.

**Proposition E.2.1** For all  $s \in \mathcal{B}^*$  we have that  $0 < K(s) < \infty$ .

*Proof.* Since  $K(s)$  is defined as the length of a program-input pair, it is a non-negative integer. For non-empty strings  $s$ , we have  $K(s) > 0$ . The property  $K(s) < \infty$  is a consequence of Proposition E.2.2 and the fact that we are only dealing with finite strings. ■

The Kolmogorov complexity of a string cannot surpass the sum of its own length and a constant.

**Proposition E.2.2** There is a constant  $c$  such that for all  $s \in \mathcal{B}^*$  we have that  $K(s) \leq l(s) + c$ .

*Proof.* Let  $s \in \mathcal{B}^*$  be an arbitrary string, and consider the encoding of a Turing machine  $p$  such that for any input  $v = s$ , it halts and outputs  $s$ . The program  $p$  is designed to simply reproduce its input. Given this setup, when  $p$  is executed on a universal Turing machine  $U$  with  $s$  as input, it satisfies the condition  $U(p, s) = s$ . The length of  $p$  is a constant  $c$  across all strings  $s$ . There exists a fixed program  $p_{\text{id}}$  that outputs its input; its length is constant and independent of  $s$ . By the definition of Kolmogorov Complexity  $K(s)$ , which seeks the minimum length of a program-input pair that generates  $s$ , the combination of  $p$  and  $s$  presents a feasible solution. Therefore, we have  $K(s) \leq l(s) + l(p) = l(s) + c$ . ■

The size of the constant  $c$  depends on the specific encoding schema used by the selected universal Turing machine  $U$ , but it is independent of the string  $s$ . In Section E.6, we will explore the characteristics of random strings, which are defined as strings that cannot be compressed. Such strings exhibit a Kolmogorov complexity close to their own length, that is,  $K(s) \geq l(s) - c$  for some constant  $c$ .

The absolute difference in Kolmogorov complexity between any string  $x$  and its transformed counterpart  $f(x)$ , via a computable bijection, is bounded by a constant  $c$ . That is, not only does  $f$  not increase the complexity of  $x$  by more than a constant, but also  $f$  does not decrease the complexity by more than a constant.

**Proposition E.2.3** Let  $f : \mathcal{B}^* \rightarrow \mathcal{B}^*$  is a computable bijection, then there exists a constant  $c$  such that  $|K(f(x)) - K(x)| < c$ .

*Proof.* Let  $P_f$  be the program that computes  $f$  and  $P_{f^{-1}}$  the program that computes the inverse of  $f$ . For any string  $x$ , let  $P_x$  be the shortest program that generates  $x$ . Then, a program  $P_{f(x)}$  that generates  $f(x)$  can be constructed by concatenating  $P_x$  with  $P_f$ . The length of this program is  $|P_{f(x)}| = |P_f| + |P_x|$ . Since  $|P_f|$  is a constant that does not depend on  $x$ , we can say that  $K(f(x)) \leq K(x) + |P_f|$ . Similarly, given  $f(x)$ , we can construct a program  $P'_x$  to generate  $x$  by applying  $P_{f^{-1}}$  to  $f(x)$ . The length of this program is  $|P'_x| = |P_{f^{-1}}| + |P_{f(x)}|$ . Thus,  $K(x) \leq K(f(x)) + |P_{f^{-1}}|$ . The two inequalities combined imply that  $|K(f(x)) - K(x)| \leq \max(|P_f|, |P_{f^{-1}}|) = c$ , where  $c$  is a constant that represents the maximum of the lengths of the programs that compute  $f$  and  $f^{-1}$ . This constant  $c$  does not depend on  $x$ , but rather on the complexity of the functions  $f$  and  $f^{-1}$ . ■

This proposition shows a remarkable stability of informational content under computable bijections, underscoring the intrinsic robustness of Kolmogorov complexity in the face of such transformations.

■ **Example E.3** Consider the function  $f : \mathcal{B}^* \rightarrow \mathcal{B}^*$  that reverses the order of the bits in a string, i.e.,  $f(x_1 x_2 \dots x_n) = x_n \dots x_2 x_1$ . This function is a computable bijection, since both  $f$  and its inverse (which is itself) can be computed by a fixed, finite program. If  $x$  is a highly compressible string, such as  $x = 1010 \dots 10$  repeated 1,000 times, then  $f(x)$  is also highly compressible (it is the same pattern written backwards). If  $x$  is an incompressible random string, then  $f(x)$  is also incompressible. The difference in Kolmogorov complexity between  $x$  and  $f(x)$  is bounded by the length of the fixed program that reverses the bits, a constant independent of  $x$ . ■

Finally, it is worth emphasizing that some strings are very compressible, and this phenomenon occurs at every string length. For instance, the string of  $n$  zeros,

$$0^n = \underbrace{00 \dots 0}_{n \text{ times}},$$

has a description of length  $O(\log n)$ : a program that prints "0" exactly  $n$  times. Therefore,  $K(0^n) \leq c + \log n$ , which is which is asymptotically much smaller than  $n$ . Similarly, strings with simple patterns, such as alternating zeros and ones, or repetitions of short substrings, can always be described concisely regardless of their total length.

### E.3 Joint Kolmogorov Complexity

The joint Kolmogorov complexity of two strings  $s$  and  $t$  is defined as the length of the shortest program  $p$  that, when executed on a universal Turing machine  $U$ , outputs the pair  $\langle s, t \rangle$ , in such a way that both strings can be unambiguously retrieved. Here,  $\langle s, t \rangle$  denotes a computable pairing function that encodes two strings into a single string in such a way that both components can be effectively recovered. The pairing function  $\langle s, t \rangle$  is assumed to be a fixed, computable bijection with computable inverse, so that both  $s$  and  $t$  can be effectively recovered. Different choices of pairing function affect  $K(s, t)$  by at most an additive constant.

**Definition E.3.1 — Joint Kolmogorov Complexity.** The *Joint Kolmogorov complexity* of the strings  $s, t \in \mathcal{B}^*$ , denoted by  $K(s, t)$ , is defined as:

$$K(s, t) = \min_{p, v \in \mathcal{B}^*} \{l(p) + l(v) : U(p, v) = \langle s, t \rangle\}$$

The notation  $K(s, t)$  and  $K(st)$  represent two different concepts in the context of Kolmogorov complexity.  $K(s, t)$  refers to the joint Kolmogorov complexity of two strings  $s$  and  $t$  as per Definition E.3.1, meanwhile  $K(st)$  represents the Kolmogorov complexity of the concatenation of  $s$  and  $t$ , without any additional structure to distinguish between them, and so, Definition E.1.2 is applied. The choice between  $K(s, t)$  and  $K(st)$  depends on whether it's important to preserve and utilize the distinction and relationship between  $s$  and  $t$ . If analyzing the interplay or the shared characteristics of  $s$  and  $t$  is relevant,  $K(s, t)$  is more appropriate. If the focus is on the information content of the combined sequence without regard to its origin from two separate strings,  $K(st)$  is used.

■ **Example E.4** Consider the strings  $s = 0000$  and  $t = 1111$ . The concatenation  $st = 00001111$  can be described by a short program that prints this eight-bit string directly, so  $K(st)$  is roughly the length of that description. However, the joint description  $\langle s, t \rangle$  requires that the decoding procedure be able to recover the boundary between  $s$  and  $t$ . Thus  $K(s, t)$  and  $K(st)$  differ by at most a fixed constant, reflecting the extra information required to separate  $s$  and  $t$ . ■

Our first proposition highlights a fundamental symmetry in Kolmogorov complexity, illustrating that the complexity of describing a pair of strings in either order differs by at most a constant. This reflects the intrinsic property that the information content is independent of the specific arrangement of the strings being described. The constant  $c$  encapsulates the overhead associated with the operations needed to reverse the order of the strings.

**Proposition E.3.1** There is a constant  $c$  such that for all  $x, y \in \mathcal{B}^*$  we have that  $|K(x, y) - K(y, x)| \leq c$ .

*Proof.* Let  $U$  be a universal Turing machine, and let  $p_{yx}$  be the shortest program that outputs  $\langle y, x \rangle$  when executed on  $U$ . To obtain a program that outputs  $\langle x, y \rangle$ , we can prepend  $p_{yx}$  with a fixed program that swaps the order of the two components in the decoded pair. This additional program has a constant length  $c$ , independent of  $x$  and  $y$ . Therefore,  $K(x, y) \leq K(y, x) + c$ .  $\blacksquare$

Next proposition underscores the subadditive nature of Kolmogorov complexity, proving that the total complexity of describing two strings jointly cannot exceed the sum of their individual complexities by more than a fixed constant, irrespective of the strings' content.

**Proposition E.3.2** There is a constant  $c$  such that for all  $s, t \in \mathcal{B}^*$  we have that  $K(s, t) \leq K(s) + K(t) + c$ .

*Proof.* Let  $s^*$  and  $t^*$  be the shortest self-delimiting programs that generate  $s$  and  $t$ , respectively. Since  $s^*$  and  $t^*$  are prefix-free, their concatenation can be parsed unambiguously by a universal Turing machine  $U$ . A fixed wrapper program of length  $c$  instructs  $U$  to run both  $s^*$  and  $t^*$  in sequence and output the pair  $\langle s, t \rangle$ . Therefore,  $K(s, t) \leq K(s) + K(t) + c$ .  $\blacksquare$

The final proposition establishes a lower bound on the joint complexity of two strings relative to their individual complexities.

**Proposition E.3.3** There is a constant  $c$  such that for all  $s, t \in \mathcal{B}^*$  we have

$$K(s, t) \geq \max(K(s), K(t)) - c.$$

*Proof.* Let  $p$  be the shortest program that outputs the pair  $\langle s, t \rangle$ , so  $K(s, t) = l(p)$ . From  $\langle s, t \rangle$ , both  $s$  and  $t$  can be effectively recovered using a fixed decoding program of length  $c$ . Thus  $K(s) \leq K(s, t) + c$  and  $K(t) \leq K(s, t) + c$ . Rearranging gives  $K(s, t) \geq K(s) - c$  and  $K(s, t) \geq K(t) - c$ , which together imply  $K(s, t) \geq \max(K(s), K(t)) - c$ .  $\blacksquare$

## E.4 Conditional Kolmogorov complexity

In this section, we explore the concept of *conditional Kolmogorov complexity*, which measures how the description length of a string  $s$  may decrease when prior knowledge of another string  $t$  is available. This notion highlights the impact of background information on the compressibility of a description. Here,  $\langle v, t \rangle$  denotes a computable pairing function with computable inverse,

ensuring both  $v$  and  $t$  can be recovered unambiguously. Different choices of pairing function affect  $K(s|t)$  by at most an additive constant.

**Definition E.4.1 — Conditional Kolmogorov Complexity.** The *conditional Kolmogorov complexity* of a string  $s \in \mathcal{B}^*$  given the string  $t \in \mathcal{B}^*$  is defined as:

$$K(s|t) = \min_{p,v \in \mathcal{B}^*} \{l(p) + l(v) : U(p, \langle v, t \rangle) = s\}$$

This definition is equivalent up to a constant to the standard formulation  $K(s|t) = \min_p \{|p| : U(p, t) = s\}$ , since the encoding  $\langle v, t \rangle$  can be replaced by  $t$  alone with at most constant overhead. As with unconditional Kolmogorov complexity, the conditional complexity is machine-independent: for any two universal Turing machines  $U$  and  $U'$ , there exists a constant  $C_{U,U'}$  such that for all  $s, t \in \mathcal{B}^*$ ,

$$K_U(s|t) \leq K_{U'}(s|t) + C_{U,U'}.$$

■ **Example E.5** Let  $s = 1010101010$  and  $t = 10$ . The unconditional complexity  $K(s)$  is proportional to the length of  $s$ , since it is a ten-bit string. However, given  $t$ , we can describe  $s$  succinctly by a short program: “print  $t$  five times.” Thus  $K(s|t)$  is only  $O(\log 5)$ , far smaller than  $K(s)$ . This illustrates how prior knowledge reduces the description length. ■

As with the unconditional Kolmogorov complexity, the conditional Kolmogorov complexity is a finite non-negative integer.

**Proposition E.4.1** For all  $s, t \in \mathcal{B}^*$  we have that  $0 \leq K(s|t) < \infty$ .

*Proof.*  $K(s|t)$  is the length of a program-input pair, so it is non-negative. The property  $K(s|t) < \infty$  is a consequence of Proposition E.2.2 and the fact that we are only dealing with finite strings. ■

Next proposition posits that when a string  $s$  is conditioned upon itself, its complexity reduces to at most a universal constant.

**Proposition E.4.2** There is a constant  $c$  such that for all  $s \in \mathcal{B}^*$  we have that  $K(s|s) \leq c$ .

*Proof.* When a string  $s$  is conditioned on itself, the information needed to generate  $s$  from  $s$  can be encapsulated in a Turing machine that simply copies its input to its output. This machine, being independent of the specific content of  $s$ , has a fixed length  $c$ . ■

This proposition explores the relationship between unconditional and conditional Kolmogorov complexities, establishing an upper bound for the

latter. It asserts that for any strings  $s$  and  $t$ , the complexity of  $s$  given  $t$  is at most the complexity of  $s$  alone, plus a constant  $c$ . This highlights the intuitive notion that having additional information can only reduce the complexity of describing a string, or in the worst case, add a constant overhead, but does not increase it beyond this bound.

**Proposition E.4.3** There is a constant  $c$  such that for all  $s, t \in \mathcal{B}^*$  we have that  $K(s|t) \leq K(s) + c$ .

*Proof.* Let  $p_s$  be the shortest program that generates  $s$  without any auxiliary input, so  $|p_s| = K(s)$ . Construct a new program  $p'$  that ignores the conditional input  $t$  and executes  $p_s$ . The additional instructions have fixed length  $c$ , independent of  $s$  or  $t$ . Therefore  $K(s|t) \leq K(s) + c$ . ■

Conditional complexity is not symmetric. In general,  $K(s|t) \neq K(t|s)$ .

**Proposition E.4.4** There exist strings  $s, t \in \mathcal{B}^*$  such that  $K(s|t) \neq K(t|s)$ .

*Proof.* Consider  $s$  as a random  $n$ -bit string, and let  $t$  be a shortest description of  $s$ . Then  $K(s|t) = O(1)$ , since given  $t$  one can directly reconstruct  $s$ . On the other hand,  $K(t|s) \geq n - O(1)$ , since otherwise we would obtain a shorter description of  $s$ , contradicting minimality. Thus, in general, conditional Kolmogorov complexity is asymmetric. ■

The relationship between conditional, unconditional, and joint Kolmogorov complexities offers a comprehensive perspective on the informational interdependencies of binary strings. It posits that the complexity of a string  $s$  given another string  $t$  is at most the complexity of  $s$  alone, which in turn is no greater than the joint complexity of both  $s$  and  $t$ .

**Proposition E.4.5** For all  $s, t \in \mathcal{B}^*$  we have that  $K(s|t) \leq K(s) \leq K(s, t)$ .

*Proof.* The first inequality follows from Proposition E.4.3. The second inequality follows from Proposition E.3.3, since from  $\langle s, t \rangle$  we can recover  $s$  with constant overhead. ■

The Kolmogorov complexity chain rule is a fundamental principle that connects the joint complexity of two strings with their individual and conditional complexities. It asserts that the total complexity of a pair of strings  $s$  and  $t$  can be decomposed into the complexity of  $s$  plus the complexity of  $t$  given  $s$ , up to a logarithmic additive term. This relationship mirrors the additive property of entropy in information theory and provides a powerful tool for understanding the interplay between information content and conditional information in the context of Kolmogorov complexity.

**Proposition E.4.6 — Kolmogorov chain rule.** For all  $s, t \in \mathcal{B}^*$  we have

$$K(s, t) = K(s) + K(t \mid s) + O(\log(K(s, t))).$$

*Proof.* The upper bound follows by concatenating a shortest program  $p_s$  that generates  $s$  with a shortest program  $p_{t|s}$  that generates  $t$  given  $s$ , plus a fixed wrapper program that runs them in sequence. The overhead is logarithmic, due to the need for self-delimiting encodings of program lengths.

For the lower bound, note that from  $\langle s, t \rangle$  one can reconstruct  $s$ , and then from  $s$  and a decoding procedure reconstruct  $t$ . This yields  $K(s, t) \geq K(s) + K(t|s) - O(\log(K(s, t)))$ . Together, these inequalities establish the claim. ■

■ **Example E.6** Let  $s = 0000$  and  $t = 1111$ . The joint description  $\langle s, t \rangle$  can be generated by first describing  $s$  (a short program "print four zeros") and then describing  $t$  given  $s$  (a short program "print four ones"). Hence  $K(s, t) \approx K(s) + K(t|s)$  up to logarithmic overhead, illustrating the chain rule in practice. ■

## E.5 Information Distance

In this section, we aim to introduce a universal metric for quantifying the absolute information distance between two or more individual entities encoded as strings of symbols. Intuitively, the information distance between two strings  $s$  and  $t$  can be understood as the length of the shortest computer program for a universal computer that enables the generation of  $s$  given  $t$  and vice versa.

■ **Definition E.5.1** The *information distance* between two strings  $s, t \in \mathcal{B}^*$  with respect to a universal Turing machine  $U$ , denoted by  $ID_U(s, t)$ , is defined as

$$ID_U(s, t) = \min\{l(p) : U(p, s) = t, U(p, t) = s\}$$

For any two universal Turing machines  $U_1$  and  $U_2$ , the information distance between two strings differs by at most an additive constant  $c$ , which depends only on the choice of machines and not on the specific strings. It is also important to note that despite its theoretical significance, information distance is non-computable: there does not exist an algorithm that can compute it exactly for arbitrary strings.

One might consider using the conditional Kolmogorov complexity  $K(s \mid t)$  as a measure of information distance. However, this quantity is asymmetric (see Proposition E.4.2), making it unsuitable as a distance. Similarly, the

sum  $K(s \mid t) + K(t \mid s)$  is also inadequate, as it double-counts the overlapping information needed to transform  $s$  into  $t$  and vice versa.

The following result shows how information distance can be expressed in terms of conditional Kolmogorov complexities.

**Proposition E.5.1** Let  $s, t \in \mathcal{B}^*$  be two binary strings. Then

$$ID_U(s, t) = \max\{K(s \mid t), K(t \mid s)\} + O(\log \max\{K(s \mid t), K(t \mid s)\}).$$

*Proof sketch.* Suppose without loss of generality that  $K(s \mid t) \geq K(t \mid s)$ . A shortest program  $p$  of length  $K(s \mid t)$  transforms  $t$  into  $s$ . To also transform  $s$  into  $t$ , we append a fixed routine that, given  $s$ , reconstructs  $t$  by inverting the transformation. This requires at most logarithmic overhead to encode program lengths in a self-delimiting way. Thus one program of length  $K(s \mid t) + O(\log K(s \mid t))$  suffices in both directions. Therefore,

$$ID_U(s, t) = \max\{K(s \mid t), K(t \mid s)\} + O(\log \max\{K(s \mid t), K(t \mid s)\}).$$

■

■ **Example E.7** Let  $s = 0011$  and  $t = 1100$ . The bitwise exclusive-or  $s \oplus t = 1111$  captures the difference between the two strings. Given  $s$  and the xor-mask  $s \oplus t$ , one can reconstruct  $t$  by  $t = s \oplus (s \oplus t)$ , and conversely obtain  $s$  from  $t$  and the mask. Hence, the information distance between  $s$  and  $t$  is essentially the Kolmogorov complexity of the xor-mask,  $K(s \oplus t)$ , up to logarithmic overhead. If the mask is simple (all ones), the distance is small; if it is random, the distance is close to the full length of the strings. ■

It is convenient to introduce the following function, which captures the essence of information distance:

$$E(x, y) = \max\{K(x \mid y), K(y \mid x)\}.$$

**Proposition E.5.2**  $E(x, y)$  is a metric up to logarithmic additive terms.

*Proof sketch.* Non-negativity and symmetry are immediate from the definition. Identity holds since  $E(x, x) = O(1)$ , while if  $x \neq y$ , at least one of  $K(x \mid y)$  or  $K(y \mid x)$  is large, so  $E(x, y) > 0$ . For the triangle inequality, let  $x, y, z$  be strings. From  $x$  one can compute  $y$  using a program of length  $K(y \mid x)$ , and from  $y$  one can compute  $z$  using a program of length  $K(z \mid y)$ . Composing these programs, from  $x$  we can compute  $z$  using a program of length  $K(y \mid x) + K(z \mid y) + O(\log)$ . Hence

$$K(z \mid x) \leq K(y \mid x) + K(z \mid y) + O(\log).$$

A similar argument applies symmetrically, establishing the triangle inequality for  $E(x, y)$  up to logarithmic additive terms. ■

The information distance  $E(x, y)$  admits an alternative characterization in terms of the joint and individual Kolmogorov complexities, as follows.

**Proposition E.5.3**

$$E(x, y) = \max\{K(x | y), K(y | x)\} = K(xy) - \min\{K(x), K(y)\} + O(\log K(xy)).$$

*Proof sketch.* Assume without loss of generality that  $K(x) \leq K(y)$ . By the chain rule,

$$K(xy) = K(x) + K(y | x) + O(\log K(xy)).$$

Thus

$$K(y | x) = K(xy) - K(x) + O(\log K(xy)).$$

Since  $E(x, y) = \max\{K(x | y), K(y | x)\}$ , the dominant term is  $K(y | x)$  in this case. Hence

$$E(x, y) = K(xy) - \min\{K(x), K(y)\} + O(\log K(xy)).$$

■

We now introduce the notion of *admissible information distances*.

**Definition E.5.2** An *admissible information distance*  $d(x, y)$  is a total function mapping pairs of strings to non-negative integers such that: (1)  $d(x, y)$  is upper semicomputable, (2) it satisfies the metric properties up to  $O(1)$ , and (3) it is normalized in the sense that  $\sum_y 2^{-d(x, y)} \leq 1$  for all  $x$ .

Among all admissible information distances,  $E(x, y)$  plays a distinguished role by being the smallest one up to an additive constant, as stated next.

**Proposition E.5.4**  $E(x, y)$  is an admissible information distance, and moreover it is minimal: for every other admissible information distance  $d(x, y)$ , we have

$$E(x, y) \leq d(x, y) + O(1).$$

*Proof idea.*  $E(x, y)$  is upper semicomputable because conditional complexities are. It satisfies the metric axioms up to logarithmic additive terms, as shown earlier. Normalization follows because conditional Kolmogorov complexities induce a semimeasure. Minimality holds because any admissible distance can be simulated by conditional descriptions, while  $E(x, y)$  already captures the maximal overlap of descriptions. Thus  $E(x, y)$  is universal among admissible information distances. ■

This universality means that  $E(x, y)$  encompasses all other admissible information distances: it is the minimal such function up to additive constants. Therefore,  $E(x, y)$  provides a canonical, universal measure of information distance.

## Normalized Information Distance

Information distance is an absolute measure; however, when assessing similarity, we are often more concerned with relative measures. For instance, two strings of length 1,000,000 differing by 1000 bits are perceived as relatively more similar than two strings of length 1000 that differ by the same number of bits. This motivates the introduction of a normalized version of information distance: the size of the description required for transformation should be evaluated relative to the sizes of the objects being compared.

**Definition E.5.3** The *normalized information distance* between two binary strings  $s, t \in \mathcal{B}^*$ , denoted by  $NID(s, t)$ , is defined as:

$$NID(s, t) = \frac{\max\{K(s \mid t), K(t \mid s)\}}{\max\{K(s), K(t)\}}.$$

As expected, the normalized information distance takes values between 0 and 1, up to negligible additive terms.

**Proposition E.5.5** The normalized information distance  $NID(s, t)$  takes values in the range  $[0, 1]$  up to vanishing additive terms.

*Proof.* Non-negativity follows because Kolmogorov complexities are non-negative, so  $NID(s, t) \geq 0$ . For the upper bound, we use the fact that  $K(s \mid t) \leq K(s) + O(1)$  and  $K(t \mid s) \leq K(t) + O(1)$ . Therefore

$$\max\{K(s \mid t), K(t \mid s)\} \leq \max\{K(s), K(t)\} + O(1).$$

Dividing by  $\max\{K(s), K(t)\}$  gives

$$NID(s, t) \leq 1 + O\left(\frac{1}{\max\{K(s), K(t)\}}\right).$$

Hence, up to negligible terms,  $NID(s, t) \in [0, 1]$ . ■

The normalized information distance not only captures relative similarity but also inherits the essential structure of a metric space, satisfying the axioms of a metric up to vanishing additive terms.

**Proposition E.5.6** The normalized information distance  $NID(x, y)$  is a metric, up to negligible errors.

*Proof sketch.* Non-negativity and symmetry are immediate from the definition. For identity,  $NID(x,y) = O(1/\max\{K(x),K(y)\})$  when  $x = y$ , while if  $x \neq y$  at least one of the conditional complexities is large, so  $NID(x,y) > 0$ . For the triangle inequality, let  $x,y,z$  be strings. From the inequality

$$K(z|x) \leq K(y|x) + K(z|y) + O(\log),$$

we obtain

$$E(x,z) \leq E(x,y) + E(y,z) + O(\log),$$

where  $E(\cdot,\cdot)$  is the (unnormalized) information distance. Normalizing by  $\max\{K(x),K(z)\}$  introduces at most a vanishing additive error. Thus  $NID$  satisfies the metric axioms up to negligible terms. ■

The normalized information distance can also be expressed directly in terms of the joint and individual Kolmogorov complexities, as shown below.

### Proposition E.5.7

$$NID(x,y) = \frac{K(xy) - \min\{K(x),K(y)\} + O(\log K(xy))}{\max\{K(x),K(y)\}}.$$

*Proof sketch.* From the earlier result for information distance we know

$$E(x,y) = \max\{K(x|y),K(y|x)\} = K(xy) - \min\{K(x),K(y)\} + O(\log K(xy)).$$

Dividing both sides by  $\max\{K(x),K(y)\}$  yields the stated expression for  $NID(x,y)$ . ■

### Normalized Compression Distance

Although the normalized information distance is not computable, it has a wide range of potential applications. By approximating Kolmogorov complexity with practical compressors, we can obtain a computable surrogate of  $NID$ . Let  $Z(s)$  denote the length in bits of the string  $s$  compressed using a compressor  $Z$  (such as gzip, bzip2, or PPMZ). Similarly,  $Z(s|t)$  denotes the compressed size of  $s$  when the compressor is given  $t$  as auxiliary input. This motivates the following definition.

**Definition E.5.4** The *normalized compression distance* between two strings  $s,t \in \mathcal{B}^*$ , given the compressor  $Z$ , and denoted by  $NCD_Z(s,t)$ , is defined as:

$$NCD_Z(s,t) = \frac{\max\{Z(s|t),Z(t|s)\}}{\max\{Z(s),Z(t)\}}$$

In practice, most compressors do not support conditional compression, making  $Z(s \mid t)$  difficult to compute directly. Fortunately, the definition can be reformulated in terms of concatenated compression, which avoids this issue.

**Proposition E.5.8** The normalized compression distance between two strings  $s, t \in \mathcal{B}^*$ , given the compressor  $Z$ , satisfies:

$$NCD_Z(s, t) = \frac{Z(st) - \min\{Z(s), Z(t)\}}{\max\{Z(s), Z(t)\}}.$$

*Proof sketch.* From the definition of  $NCD_Z$ , we have

$$NCD_Z(s, t) = \frac{\max\{Z(s|t), Z(t|s)\}}{\max\{Z(s), Z(t)\}}.$$

For real-world compressors, the conditional compression  $Z(s|t)$  can be approximated by

$$Z(s|t) \approx Z(st) - Z(t),$$

since compressing the concatenation  $st$  encodes  $s$  with  $t$  as a prefix, effectively using  $t$  as context. Similarly,

$$Z(t|s) \approx Z(st) - Z(s).$$

Therefore,

$$\max\{Z(s|t), Z(t|s)\} \approx Z(st) - \min\{Z(s), Z(t)\}.$$

Substituting this into the definition yields

$$NCD_Z(s, t) = \frac{Z(st) - \min\{Z(s), Z(t)\}}{\max\{Z(s), Z(t)\}}.$$

This equality holds up to negligible additive errors, which vanish for ideal compressors. ■

The normalized compression distance constitutes a family of distances, each defined by the choice of compressor  $Z$ . The effectiveness of  $Z$  determines how closely the normalized compression distance mirrors the normalized information distance, ultimately influencing how well  $NCD$  approximates  $NID$  in practical applications.

## E.6 Incompressibility and Randomness

A string is considered incompressible if its Kolmogorov complexity is approximately equal to its length; in other words, there is no significantly shorter description or program that can produce it.

**Definition E.6.1** For each constant  $c$  we say that a string  $s \in \mathcal{B}^*$  is  $c$ -incompressible if  $K(s) \geq l(s) - c$ .

The next proposition shows that incompressible strings exist for every string length.

**Proposition E.6.1** For every length  $n$ , there exists a string of length  $n$  that is incompressible.

*Proof.* By a counting argument, the number of programs of length less than  $n$  is at most  $2^n - 1$ , while the number of binary strings of length  $n$  is exactly  $2^n$ . Therefore, at least one string of length  $n$  cannot be generated by any program shorter than  $n$ , and is thus incompressible. ■

We extend the term *incompressible string* to include all  $c$ -incompressible strings with  $c$  small. In this sense, most strings are incompressible.

**Proposition E.6.2** For any  $n$  and constant  $c > 0$ , at least a fraction  $1 - 2^{-(c-1)}$  of the strings of length  $n$  are  $c$ -incompressible.

*Proof.* Consider the number of programs of length at most  $n - c$ . There are at most

$$\sum_{i=0}^{n-c} 2^i = 2^{n-c+1} - 1$$

such programs. Since there are  $2^n$  strings of length  $n$ , at least  $2^n - (2^{n-c+1} - 1)$  strings must be  $c$ -incompressible. Thus the fraction of  $c$ -incompressible strings of length  $n$  is at least

$$1 - \frac{2^{n-c+1} - 1}{2^n} = 1 - 2^{-(c-1)} + \frac{1}{2^n}.$$

For large  $n$ , this tends to  $1 - 2^{-(c-1)}$ . ■

The notion of randomness, especially in the context of sequences or strings, is often associated with unpredictability, lack of pattern, or absence of structure. Kolmogorov complexity formalizes this intuition by linking randomness to incompressibility: a string is random if it cannot be generated by any program significantly shorter than the string itself.

**Definition E.6.2** We say that a string  $s \in \mathcal{B}^*$  is *random* if it is  $c$ -incompressible for some fixed small constant  $c$ .

Random strings are characterized by high Kolmogorov complexity, meaning they are incompressible. The shortest program that can generate a random string is essentially the string itself. Such strings contain the maximum amount of information possible.

■ **Example E.8** Consider a string generated by flipping a fair coin for each bit, such as 1011010110110101. With overwhelming probability, this string will be incompressible, since no shorter program or pattern can generate it. Any attempted compression would yield a program of length comparable to the string itself. ■

The unpredictability of random strings stems from their incompressibility. Because no algorithm can exploit patterns in such strings, predicting their bits is no better than random guessing.

Random strings are typical in the space of all strings. In the sense of Kolmogorov complexity, almost all strings are random, while only a vanishing fraction admit significantly shorter descriptions.

■ **Example E.9** Consider the set of all conceivable high-resolution digital photographs, each represented as a binary string encoding pixel colors and intensities. Only a tiny fraction of these images exhibit recognizable regularities, such as a uniform blue sky or a solid monochromatic background, which can be compressed into shorter binary descriptions. By contrast, the overwhelming majority of possible images resemble random strings: they lack compressible patterns and are essentially incompressible. ■

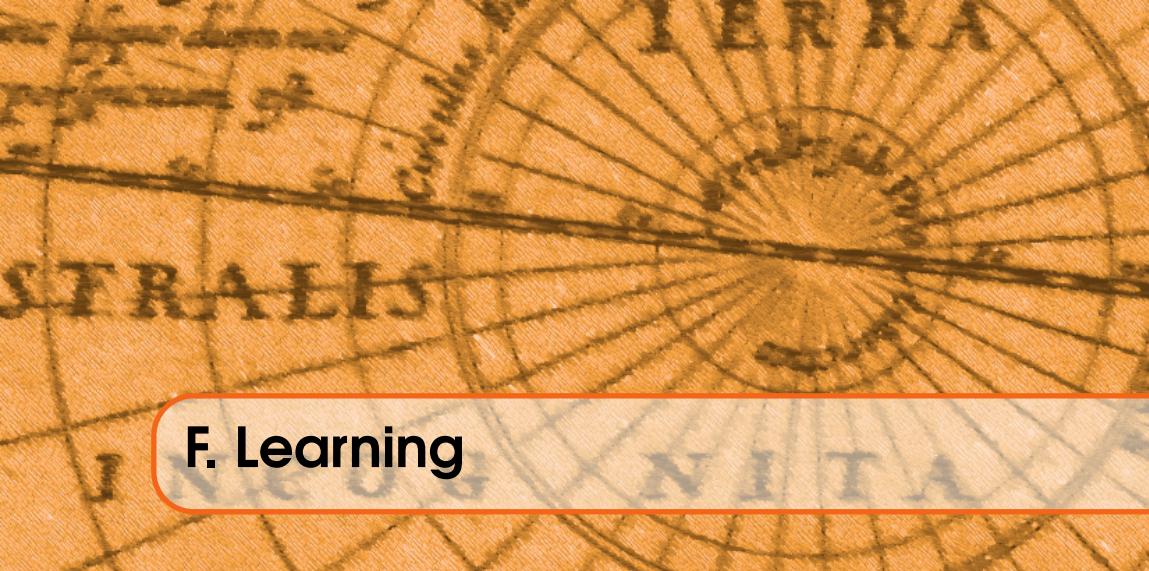
## References

Kolmogorov complexity, named after the Soviet mathematician Andrey Kolmogorov, is a measure of the complexity of a string of text or other data. It is defined as the length of the shortest possible description of the string in some fixed universal description language. This measure is inherently uncomputable in general, as proven by the halting problem's undecidability, but it provides a powerful theoretical tool for understanding data complexity. Beyond theoretical interest, Kolmogorov complexity has applications in pattern recognition, data compression, and the study of randomness. It offers a framework for understanding the limits of compressibility and the nature of information.

The concept of Kolmogorov complexity emerged independently in the works of several researchers in the early 1960s. Andrey Kolmogorov introduced it in 1965 [Kol65], motivated by trying to formalize the concept of

randomness and complexity through the lens of information theory. Ray Solomonoff laid the groundwork for algorithmic information theory, introducing a related concept that would later be recognized as a form of Kolmogorov complexity [Sol64]. His work focused on the idea of describing data compactly using probabilistic models. Almost simultaneously with Kolmogorov, Gregory Chaitin developed similar ideas [Cha69]. Chaitin is known for introducing the concept of algorithmic randomness and for his work on the incompleteness theorem, which relates to the limits of formal systems in proving the complexity of sequences. The motivations behind the development of Kolmogorov Complexity were multifaceted, encompassing the desire to better understand the nature of information, randomness, and the limits of computation and prediction, thus laying the groundwork for numerous applications in theoretical computer science, mathematics, and beyond.

For those readers interested in delving into the details of Kolmogorov complexity, a variety of foundational texts and advanced treatments are available. [LV13] provides a comprehensive coverage to the concepts and applications of Kolmogorov complexity, and it's widely regarded as the definitive textbook on the subject, although it is not recommended for beginners. [Cal02] delves deeply into the foundations of algorithmic information theory, focusing on the rigorous mathematical exploration of randomness and complexity through Kolmogorov complexity. [CT12] broader in scope, this book provides an excellent foundation in information theory, including discussions relevant to Kolmogorov complexity. It's a great resource for understanding the context in which Kolmogorov complexity operates within information theory.



## F. Learning

*Some mathematical statements are true for no reason,  
they're true by accident.*

Gregory Chaitin

Learning from data lies at the heart of scientific discovery and technological innovation. It refers to the broad collection of methods and algorithms designed to automatically build mathematical models from sample datasets, usually with the aim of making predictions, classifying objects, or revealing hidden structure in the data. In the past decade, learning algorithms have achieved striking success in diverse domains such as self-driving cars, practical speech recognition, effective web search, and personalized recommendation systems.

In this chapter, we will explore how the problem of learning from data is formally formulated within the field of machine learning. In doing so, this chapter builds on the probabilistic foundations introduced earlier in Section ??, extending them toward the design of algorithms that can infer general patterns from finite observations.

We will begin by reviewing the classical approach to learning based on *statistical inference*, where learning is framed as estimating the parameters of an assumed probabilistic model. We will then turn to *machine learning*

*methods*, which often relax these assumptions and focus on constructing predictive models that generalize well to unseen data, even when the underlying data-generating process is unknown or too complex to model explicitly.

In addition, we will study two powerful principles that are central to the *theory of nescience*: the *Minimum Description Length (MDL)* and the *Minimum Message Length (MML)* principles. These approaches reinterpret learning as a problem of finding the most concise explanation for the data, connecting model selection to fundamental concepts from algorithmic information theory and data compression. This perspective will also allow us to extend the concept of nescience—originally defined for textual descriptions—to empirical datasets.

Finally, we will review the area of *multiobjective optimization*, which addresses learning problems involving multiple, often conflicting, objectives. Unlike traditional single-objective approaches, multiobjective methods aim to find a set of Pareto-optimal solutions, requiring additional techniques to incorporate preference structures and select a final decision.

Throughout the chapter, our goal will be to develop a unified view of learning, spanning from classical statistical inference to modern machine learning and nescience-based methods, and to prepare the ground for the introduction of new learning algorithms inspired by the theory of nescience in Chapter 8.

## F.1 Statistical Inference

*Statistical inference* has traditionally been presented as the process of making sense of reality through data. It applies probabilistic models to relate finite observations to broader populations, aiming to offer predictions and structured conclusions. However, statistical inference is not a direct reflection of reality but rather a tool that depends heavily on assumptions and idealized models. These models simplify complex phenomena, and the validity of their conclusions relies on how well these assumptions align with the underlying data-generating processes.

The random sampling model serves as a stepwise framework in statistical inference, guiding how we connect finite samples to broader populations:

Step 1.- Defining the Population and Collecting Data: The first step is to define the population of interest and determine how data will be collected. This involves specifying the sampling process and ensuring that each data point is drawn independently and identically distributed (iid) from the same population, with every unit having a nonzero probability of selection. Once the sampling process is defined, data is collected as a sample from the population, and each observation is treated as a realization of a random

variable following the specified distribution.

Step 2.- Constructing and Checking the Random Sampling Model: Based on assumptions about how the data was generated, the random sampling model is mathematically expressed. For example, we may assume that  $X_1, \dots, X_n$  are iid random variables with common distribution  $f(x | \theta)$ , where  $\theta$  is an unknown parameter of the population distribution. This step formalizes the relationship between the data and the population. Real-world data rarely fits these idealized assumptions perfectly, so it is essential to check for deviations such as selection bias or lack of independence, and to address them appropriately.

Step 3.- Applying the Likelihood Function and Estimating Parameters: The likelihood function plays a central role in parameter estimation by quantifying how plausible different parameter values are given the observed data  $\mathbf{X}$ . Methods such as maximum likelihood estimation (MLE) or Bayesian inference refine these estimates, translating raw data into structured knowledge about the population. For example, the sample mean  $\bar{X}$  is often used as an estimate of the population mean  $\mu$ .

Step 4.- Quantifying Uncertainty and Generalizing to the Population: Recognizing that the sample is just one realization of a random process, uncertainty must be quantified to understand how sample statistics behave across repeated samples. Sampling distributions, confidence intervals, and hypothesis tests help describe this variability. The Central Limit Theorem ensures that for large samples, the distribution of the sample mean  $\bar{X}$  approximates a normal distribution, making inference more reliable. Finally, results are generalized from the sample to the population using frequentist methods such as confidence intervals and  $p$ -values, or Bayesian methods that update beliefs about  $\theta$  by incorporating prior knowledge.

The random sampling model provides a structured, step-by-step approach for making inferences from data. However, each step depends on assumptions that rarely hold perfectly in practice. As such, conclusions drawn from statistical inference should be viewed as tentative and context-dependent, subject to revision when new data or insights become available.

**Definition F.1.1** A *statistical model* is a family of probability distributions, together with a specification of its probability distribution, and the identification of the parameters, denoted by  $\theta$ , of that distribution.

A statistic is a function of the observable data. A statistic is used to summarize or describe some aspect of the sample. For example, the sample mean and the sample variance are statistics because they summarize data drawn from a sample.

**Definition F.1.2** Let  $\mathbf{X} = (X_1, \dots, X_n)$  be a random sample. A *statistic* is a random variable  $T = r(X_1, \dots, X_n)$ , where  $r()$  is an arbitrary real-valued function of  $n$  variables.

A *parametric random variable* is a random variable  $X$  that belongs to a family of distributions parameterized by  $\theta$ . The parameter  $\theta$  can either be a single scalar or a vector of values, and it is often treated as a fixed but unknown quantity. In the Bayesian setting,  $\theta$  itself is modeled as a random variable with a prior distribution. The set  $\Theta = \{\theta_1, \theta_2, \dots\}$ , consisting of all possible values of  $\theta$ , is called the *parameter space*, and its elements are referred to as *parameters*. When the parameter  $\theta$  is unknown, the distribution of the random variable  $X$  is said to be conditional on  $\theta$ , denoted by  $f(X | \theta)$ .

As in Chapter ??, in this section we are only considering parametric discrete random variables defined over discrete probability spaces.

■ **Example F.1** We have seen in Definition B.6.4 that a binomial distribution with parameters  $n$  and  $p$  is a model for a family of experiments in which we are interested in knowing the number of successes in a sequence of  $n$  independent binary trials, where the probability of success is  $p$ . If  $X$  is a random variable following a binomial distribution, the probability of getting exactly  $k$  successes is given by:

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

In statistical inference, we are usually interested in the inverse problem. That is, we have the actual result of an experiment composed of  $n$  trials, in which we know how many successes  $k$  we have obtained, and we would like to estimate the parameter  $p$ , that is, the probability of success. ■

We assume that the true value of the unknown parameter  $\theta$  can be inferred, typically by analyzing a collection of data samples. The observable data  $\mathbf{X} = (X_1, \dots, X_n)$  is modeled as a random sample from the distribution  $f(X | \theta)$  conditional on  $\theta$ .

An estimator of a parameter  $\theta$  is a function of the random sample  $\mathbf{X}$  that we hope provides a value close to the unknown parameter  $\theta$ .

**Definition F.1.3** Let  $\mathbf{X} = (X_1, \dots, X_n)$  be a random sample from a discrete random variable  $X$  with parameter  $\theta$ . An *estimator* of the parameter  $\theta$  is a real-valued function  $\delta(X_1, \dots, X_n)$ . If  $X_1 = x_1, \dots, X_n = x_n$  are observed, then  $\delta(x_1, \dots, x_n)$  is called the *estimate* of  $\theta$ .

The estimator itself is a random variable, and its probability distribution can be derived from the joint distribution of  $X_1, \dots, X_n$ . Every estimator is a statistic, but not every statistic is an estimator. The estimate is a real number.

When selecting an estimator, it is important to measure how "good" an estimate is. One common approach is to quantify the loss or cost associated with choosing an estimate that deviates from the true parameter value. This is done using a loss function, which assigns a numerical penalty to the difference between the estimate and the actual parameter.

**Definition F.1.4** A *loss function* is a real-valued function of two variables,  $L(\theta, a)$ , where  $\theta \in \Theta$  and  $a$  is a real number.

The loss function represents the cost incurred when the true parameter value is  $\theta$  and the estimate is  $a$ . In other words,  $L(\theta, a)$  quantifies the loss when there is a discrepancy between the estimate and the true value.

Let  $\xi(\theta)$  denote the prior probability mass function of  $\theta$  on the set  $\Theta$ . Given a particular estimate  $a$ , the expected loss for discrete random variables is computed as:

$$E[L(\theta, a)] = \sum_{\theta \in \Theta} L(\theta, a) \xi(\theta)$$

The expected loss is the weighted average of the losses, where the weights are the probabilities  $\xi(\theta)$  assigned to each possible value of  $\theta$ . The goal is to choose an estimate  $a$  that minimizes this expected loss.

■ **Example F.2** The loss function  $L(\theta, a) = (\theta - a)^2$  is called the *squared error loss*. Let  $\theta$  be a real-valued parameter. Suppose that squared error loss is used and that the posterior mean of  $\theta$ ,  $E(\theta | \mathbf{X})$ , is finite. Then, the Bayes estimator of  $\theta$  under squared error loss is

$$\delta^*(\mathbf{X}) = E(\theta | \mathbf{X}).$$

In words, the posterior mean minimizes the expected squared deviation from the true parameter.

The loss function  $L(\theta, a) = |\theta - a|$  is called the *absolute error loss*. When absolute error loss is used, a Bayes estimator of a real-valued parameter  $\theta$  is any median of the posterior distribution of  $\theta$ . That is,

$$\delta^*(\mathbf{X}) \in \text{Median}(\theta | \mathbf{X}).$$

In words, the posterior median minimizes the expected absolute deviation from the true parameter. ■

### F.1.1 Maximum Likelihood Estimator

The Maximum Likelihood Estimator (abbreviated MLE) is a method of estimating the parameters of a probability model. It is one of the most commonly used techniques in statistics for fitting model parameters to observed data.

The core idea behind MLE is to find the parameter values that maximize a likelihood function, which represents the probability of observing the given data under the model. MLE is widely used due to its desirable properties for large samples, but its effectiveness relies on correct model specification and may struggle with small sample sizes or computational difficulties.

The likelihood represents the probability (or likelihood) of observing the data as a function of the parameters of the model.

**Definition F.1.5** Let  $X_1, X_2, \dots, X_n$  be  $n$  independent and identically distributed discrete random variables, and let  $P(X_i = x_i | \theta)$  be the probability mass function of  $X_i$ , where  $\theta \in \Theta$  is an unknown parameter, and  $\Theta$  is the parameter space. The *likelihood function* is then defined as:

$$L(\theta | X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i = x_i | \theta),$$

where  $x_1, x_2, \dots, x_n$  are the observed data points (held fixed when viewing  $L$  as a function of  $\theta$ ).

The likelihood function is a product of probabilities. Taking the logarithm of the likelihood, the *log-likelihood function*, simplifies this product into a sum:

$$\ell(\theta | X_1, X_2, \dots, X_n) = \log L(\theta | X_1, X_2, \dots, X_n) = \sum_{i=1}^n \log P(X_i | \theta).$$

Sums are generally easier to work with than products, particularly when performing differentiation for optimization purposes. This transformation makes it more straightforward to compute derivatives and apply optimization algorithms.

Maximum likelihood estimation is a method that determines values for the parameters of a model. The parameter values are found such that they maximise the likelihood that the process described by the model produced the data that were actually observed.

**Definition F.1.6** The *Maximum Likelihood Estimator*, abbreviated as MLE, for discrete random variables is defined as the parameter value that maximizes the likelihood function, i.e.,

$$\hat{\theta}_n \in \arg \max_{\theta \in \Theta} L(\theta | X_1, X_2, \dots, X_n).$$

Equivalently, any maximizer of the log-likelihood  $\ell(\theta | X_1, \dots, X_n)$  is an MLE. When there are multiple maximizers, any of them may be taken.

The MLE can also be found by maximizing the log-likelihood function:

$$\hat{\theta}_n = \arg \max_{\theta \in \Theta} \ell(\theta \mid X_1, X_2, \dots, X_n)$$

■ **Example F.3** Consider  $X_1, X_2, \dots, X_n$  i.i.d. random variables from a Bernoulli distribution with parameter  $p$ . The probability mass function is

$$P(X_i = x_i \mid p) = p^{x_i} (1-p)^{1-x_i}, \quad x_i \in \{0, 1\}.$$

The likelihood function for this sample is

$$L(p \mid X_1, X_2, \dots, X_n) = \prod_{i=1}^n p^{X_i} (1-p)^{1-X_i} = p^S (1-p)^{n-S},$$

where  $S = \sum_{i=1}^n X_i$  is the total number of successes. The function  $f(p) = p^a (1-p)^b$  on  $p \in (0, 1)$  attains its maximum at  $p = \frac{a}{a+b}$ . In our case,  $a = S$  and  $b = n - S$ , so

$$\hat{p} = \frac{S}{S + (n - S)} = \frac{S}{n}.$$

Therefore, the maximum likelihood estimator for  $p$  is

$$\hat{p} = \frac{1}{n} \sum_{i=1}^n X_i,$$

which is simply the sample mean of the Bernoulli trials (the proportion of successes). ■

The maximum likelihood estimator is consistent under suitable conditions.

**Proposition F.1.1** Let  $\hat{\theta}_n$  be an MLE for  $\theta$  based on an i.i.d. sample from a discrete model  $\{P_\theta : \theta \in \Theta\}$ . Suppose the model is identifiable (i.e.,  $P_\theta = P_{\theta_0}$  implies  $\theta = \theta_0$ ),  $\Theta$  is compact, and for each  $x$ ,  $\log P(X = x \mid \theta)$  is continuous in  $\theta$ . If  $E_{\theta_0} [|\log P(X_1 \mid \theta)|] < \infty$  for all  $\theta \in \Theta$ , then

$$\hat{\theta}_n \xrightarrow{P} \theta_0 \quad \text{as } n \rightarrow \infty,$$

where  $\theta_0$  is the true value of the parameter.

*Sketch.* (1) By identifiability and Gibbs' inequality, the expected log-likelihood (per observation) is uniquely maximized at  $\theta_0$ :

$$Q(\theta) := E_{\theta_0} [\log P(X_1 \mid \theta)] \leq Q(\theta_0),$$

with equality iff  $\theta = \theta_0$ .

(2) By the Law of Large Numbers and the continuity/integrability assumptions, the average log-likelihood converges uniformly to  $Q(\theta)$  on  $\Theta$ :

$$\sup_{\theta \in \Theta} \left| \frac{1}{n} \sum_{i=1}^n \log P(X_i | \theta) - Q(\theta) \right| \xrightarrow{P} 0.$$

(3) By the argmax continuous mapping principle, any maximizer  $\hat{\theta}_n$  of the sample log-likelihood converges in probability to the unique maximizer of  $Q$ , namely  $\theta_0$ . ■

The MLE has several notable disadvantages. Firstly, it is highly dependent on sample size; for small samples, MLE may exhibit bias or high variance, as its desirable asymptotic properties, such as consistency, are guaranteed only for large datasets. Additionally, MLE requires correct model specification. It is sensitive to assumptions about the underlying model, and if the model is misspecified, the resulting estimates can be misleading. From a computational perspective, maximizing the likelihood function can pose numerical challenges, often necessitating complex optimization techniques that are computationally demanding and may result in convergence to local rather than global maxima. Furthermore, MLE can be sensitive to extreme observations; because it seeks to maximize the likelihood of the observed sample, large or influential data points may have a disproportionate impact on the estimate, leading to distortions under contamination.

■ **Example F.4** Consider a situation where we are dealing with a uniform distribution  $U(0, \theta)$ , where  $\theta$  is the unknown upper bound of the distribution. We have a sample  $X_1, X_2, \dots, X_n$  of i.i.d. observations from this distribution, and we wish to estimate the parameter  $\theta$  using the maximum likelihood estimator. The log-likelihood function is:

$$\ell(\theta | X_1, X_2, \dots, X_n) = -n \log \theta, \quad \text{subject to } \theta \geq \max(X_1, X_2, \dots, X_n).$$

Since  $\ell(\theta)$  is decreasing in  $\theta$ , the maximum likelihood occurs at the smallest feasible value, namely

$$\hat{\theta} = \max(X_1, X_2, \dots, X_n).$$

For small samples,  $\hat{\theta}$  is biased downward:  $E[\hat{\theta}] = \frac{n}{n+1} \theta$  (e.g., if  $\theta = 10$  and  $n = 5$ , then  $E[\hat{\theta}] = \frac{5}{6} \cdot 10 = 8.33$ ). Moreover, the estimator depends only on the sample maximum; the other  $n - 1$  observations do not affect the estimate, which is statistically inefficient in finite samples. Finally, the estimator has high sensitivity to the largest observation: small perturbations of the maximum change  $\hat{\theta}$  directly (and under data contamination, a single large outlying value can inflate  $\hat{\theta}$  substantially). ■

## F.1.2 Bayesian Inference

Unlike the Maximum Likelihood Estimator (MLE), which selects the parameter values that maximize the probability of the observed data, *Bayesian inference* takes a fundamentally different approach by treating parameters as random variables endowed with probability distributions rather than fixed values. Bayesian inference also incorporates prior knowledge, allowing for an informed estimation process even before observing data. This approach enables Bayesian methods to dynamically update beliefs as new data become available. Bayes' theorem (see Theorem B.3.2) formalizes this process by combining a prior probability mass function with the likelihood of the observed data to produce a posterior probability mass function, which represents the updated belief about the parameter.

**Definition F.1.7** Let  $f(X | \theta)$  be the probability mass function of a discrete random variable  $X$  with parameter  $\theta$ . The probability distribution of the parameter  $\theta$ , denoted by  $\xi(\theta)$ , is called the *prior distribution*.

The prior distribution must be defined over the parameter space  $\Theta$ . It is called the prior distribution because it represents our knowledge or belief about the parameter  $\theta$  before observing any data.

**Definition F.1.8** Let  $f(X | \theta)$  be the probability mass function of a discrete random variable  $X$  with parameter  $\theta$ , and let  $\mathbf{X} = (X_1, \dots, X_n)$  be a random sample from  $X$ . The conditional probability mass function of the parameter  $\theta$  given the observed values  $X_1 = x_1, \dots, X_n = x_n$ , denoted  $\xi(\theta | x_1, \dots, x_n)$ , is called the *posterior distribution*.

The posterior distribution represents our updated knowledge of the parameter  $\theta$  after taking into account the observed data. A strongly informative prior can dominate the posterior, while a weak or diffuse prior allows the likelihood to play a larger role (and, under regularity, the likelihood typically dominates as  $n$  grows).

The posterior pmf is computed using Bayes' theorem:

$$\xi(\theta | \mathbf{X}) = \frac{L(\mathbf{X} | \theta) \xi(\theta)}{P(\mathbf{X})},$$

where:

- $\xi(\theta)$  is the prior distribution, which encodes our belief about  $\theta$  before seeing the data;
- $L(\mathbf{X} | \theta)$  is the likelihood function, representing the probability of observing the data given  $\theta$ ; under i.i.d. sampling,  $L(\mathbf{X} | \theta) = \prod_{i=1}^n f(x_i | \theta)$ ;

- $P(\mathbf{X})$  is the marginal probability (also called the evidence), ensuring the posterior pmf is a valid probability mass function, computed as

$$P(\mathbf{X}) = \sum_{\theta \in \Theta} L(\mathbf{X} | \theta) \xi(\theta).$$

Thus, the posterior pmf  $\xi(\theta | \mathbf{X})$  represents an updated belief about  $\theta$  after incorporating the observed data. (If  $\Theta$  is uncountable, replace sums by integrals and pmfs by densities.)

■ **Example F.5** Suppose we have a biased coin with an unknown probability  $\theta$  of landing heads. We want to estimate  $\theta$  using Bayesian inference after observing a few coin flips. Before flipping the coin, we assume a prior belief about  $\theta$ . Let's say we assume a uniform prior:

$$\xi(\theta) = \begin{cases} \frac{1}{3}, & \theta \in \{0.2, 0.5, 0.8\} \\ 0, & \text{otherwise} \end{cases}$$

This prior suggests we believe  $\theta$  is equally likely to be 0.2, 0.5, or 0.8 before any data is observed. Now, we flip the coin  $n = 3$  times and observe  $x = 2$  heads. Assuming independent flips, the likelihood function is given by:

$$p(X = x | \theta) = \binom{3}{2} \theta^2 (1 - \theta)^1$$

For the possible values of  $\theta$ :

$$p(2 | 0.2) = \binom{3}{2} (0.2)^2 (0.8)^1 = 3(0.04)(0.8) = 0.096$$

$$p(2 | 0.5) = \binom{3}{2} (0.5)^2 (0.5)^1 = 3(0.25)(0.5) = 0.375$$

$$p(2 | 0.8) = \binom{3}{2} (0.8)^2 (0.2)^1 = 3(0.64)(0.2) = 0.384$$

Using Bayes' rule:

$$\xi(\theta | X = 2) = \frac{p(X = 2 | \theta) \xi(\theta)}{P(X = 2)}$$

where the denominator is the marginal probability:

$$P(X = 2) = \sum_{\theta} p(X = 2 | \theta) \xi(\theta)$$

$$P(X = 2) = (0.096 \times \frac{1}{3}) + (0.375 \times \frac{1}{3}) + (0.384 \times \frac{1}{3})$$

$$P(X = 2) = 0.032 + 0.125 + 0.128 = 0.285$$

Now, calculating the posterior probabilities:

$$\xi(0.2 | X = 2) = \frac{0.096 \times \frac{1}{3}}{0.285} = \frac{0.032}{0.285} \approx 0.112$$

$$\xi(0.5 | X = 2) = \frac{0.375 \times \frac{1}{3}}{0.285} = \frac{0.125}{0.285} \approx 0.439$$

$$\xi(0.8 | X = 2) = \frac{0.384 \times \frac{1}{3}}{0.285} = \frac{0.128}{0.285} \approx 0.449$$

Before observing the data, we believed  $\theta$  was equally likely to be 0.2, 0.5, or 0.8. After observing 2 heads in 3 flips, our belief has shifted: the most probable value for  $\theta$  is now 0.8 (44.9% probability), followed by 0.5 (43.9% probability). The probability that  $\theta = 0.2$  has dropped significantly to 11.2%.

■

A Bayes estimator is a point estimate derived from the posterior distribution in Bayesian inference, analogous to how the MLE selects parameter values maximizing the likelihood. However, while MLE relies solely on observed data, the Bayes estimator incorporates both prior knowledge and observed data through the posterior. It is a single value that represents the "best guess" for the unknown parameter by minimizing the expected loss under a given loss function.

Suppose we observe the value  $\mathbf{x}$  of the random vector  $\mathbf{X}$  before estimating  $\theta$ , and let  $\xi(\theta | \mathbf{x})$  denote the posterior pmf of  $\theta$  on  $\Theta$ . For each estimate  $a$ , the posterior expected loss is

$$E[L(\theta, a) | \mathbf{x}] = \sum_{\theta \in \Theta} L(\theta, a) \xi(\theta | \mathbf{x}).$$

**Definition F.1.9** Let  $L(\theta, a)$  be a loss function. For each possible value  $\mathbf{x}$  of  $\mathbf{X}$ , let  $\delta^*(\mathbf{x})$  be a value of  $a$  that minimizes  $E[L(\theta, a) | \mathbf{x}]$ . Then  $\delta^*$  is called a *Bayes estimator* of  $\theta$ . Once  $\mathbf{X} = \mathbf{x}$  is observed,  $\delta^*(\mathbf{x})$  is called a *Bayes estimate* of  $\theta$ .

Equivalently, for each possible value  $\mathbf{x}$  of  $\mathbf{X}$ , the value  $\delta^*(\mathbf{x})$  is chosen so that

$$E[L(\theta, \delta^*(\mathbf{x})) | \mathbf{x}] = \min_a E[L(\theta, a) | \mathbf{x}].$$

The theory of Bayes estimators provides a coherent framework for parameter estimation. To apply it, one must specify both a loss function and a prior

distribution for the parameter. Specifying a meaningful prior in a multidimensional parameter space  $\Theta$  can be challenging because it requires modeling dependencies among parameters, and the computational cost of posterior inference typically grows with dimension (often necessitating MCMC or variational methods).

## F.2 Machine Learning

Machine learning is the study of algorithms that, given a finite sample of individuals from a population, learn a rule to make predictions about new, previously unseen individuals. Each individual is represented by a vector of attributes, one of which is designated as the target. Since the true relationship between the predictors and the target is unknown, the algorithm must approximate it. The primary objective is to construct a function, called a model, that uses the observed attributes to predict the target with minimal error, both on the training examples and on future individuals drawn from the same population.

Although statistical inference and machine learning share the goal of using data to draw conclusions about unseen cases, they tend to emphasise different aspects of this task. Classical statistical inference often begins with a probabilistic model and seeks estimators that maximise likelihood, yield interpretable parameters, and satisfy theoretical properties such as consistency, efficiency, or convergence. In contrast, machine learning typically starts with a flexible class of functions, often implemented as large-scale software artefacts such as neural networks, and focuses on minimising an empirical loss that reflects predictive accuracy, even in the absence of formal guarantees. In practice, however, the boundary is fluid: statisticians often minimise risk functions, and many machine learning methods are grounded in statistical theory.

In this book, we present machine learning as a distinct discipline, adopting a fully deterministic approach. We introduce models, residuals, and optimization criteria without relying on probability theory or its results. This choice is deliberate: by avoiding probabilistic assumptions, we align the theoretical foundations of machine learning with the framework of the theory of nescience, facilitating the integration and reuse of results across both domains.

We begin by defining the basic mathematical objects that will underpin the rest of the discussion: populations and their individuals.

**Definition F.2.1** A *population* is a non-empty, well-defined set  $\mathcal{S}$ . The elements  $\mathbf{x} \in \mathcal{S}$  are called *individuals*.

Throughout this book, we restrict our attention to countable populations. Both finite and countably infinite cases are permitted.

Each individual is described by a fixed list of attributes (also called features), and each attribute has a domain that specifies its set of admissible values.

**Definition F.2.2** Each individual  $\mathbf{x}$  in the population  $\mathcal{S}$  is characterized by  $p$  attributes ( $p \geq 1$ ), such that  $\mathbf{x} = (x_1, x_2, \dots, x_p) \in \mathcal{S}$ . For each  $i \in 1, \dots, p$ , the *domain* of the  $i$ -th attribute is defined as  $\mathcal{D}_i = \{x_i : \mathbf{x} \in \mathcal{S}\}$ .

The set  $\mathcal{D}_i$  may be any type of set—for example, the real numbers  $\mathcal{D}_i = \mathbb{R}$ , a set of integers such as  $0, 1, \dots, k$ , or a collection of categorical labels like red, green, blue. An attribute is called *quantitative* if the elements of  $\mathcal{D}_i$  can be measured numerically. Quantitative attributes can be further classified as *discrete* if they take a countable number of distinct values, or *continuous* if they can take any value within a given range or interval.

An attribute is called *qualitative* or *categorical* if it represents characteristics that cannot be measured numerically. A qualitative attribute is said to be *nominal* if there is no natural order among its categories, and *ordinal* if such an order exists.

■ **Example F.6** Consider a population consisting of the inhabitants of a small village, where we are interested in studying three attributes: age, gender, and daily water consumption. The attribute representing age, mapping individuals to their age in years, is a quantitative discrete attribute, as it takes integer values. Gender, mapping individuals to categories like "Male" or "Female," is a qualitative nominal attribute, since the categories have no inherent order. Daily water consumption, mapping individuals to the number of liters they drink per day, is a quantitative continuous attribute, as it can take any real value within a measurable range. ■

Broadly speaking, machine learning algorithms can be classified into two main categories: supervised and unsupervised. In *supervised* learning, we are given a collection of training samples (also called *predictors*) along with their corresponding observed target values (or *labels*), and our goal is to predict the target for new, previously unseen observations. In contrast, in *unsupervised* learning, no target values are provided; we are given only training samples, and the objective is to uncover the underlying structure of the data. We postpone the discussion of unsupervised learning to another section.

**Definition F.2.3** Let  $\mathcal{S}$  be a population whose individuals  $\mathbf{x}$  are characterized by  $p$  attributes, with  $\mathbf{x} = (x_1, x_2, \dots, x_p) \in \mathcal{S}$ . A *target variable*,

denoted by  $\mathbf{y}$ , is a vector corresponding to another attribute of the individuals in  $\mathcal{S}$ , distinct from the  $p$  attributes used as predictors. The *domain* of the target variable is  $\mathcal{D}_y = y_i$ .

We typically choose statistical learning methods based on whether the target attribute is quantitative or qualitative. (Whether the predictors are qualitative or quantitative is generally considered less critical.) Supervised learning algorithms are applied to *regression problems* when the target attribute is quantitative, and to *classification problems* when the target attribute is qualitative.

We assume the existence of an underlying function, or model, that relates the predictors to the target.

**Definition F.2.4** Let  $\mathcal{S}$  be a population and  $\mathbf{y}$  a target attribute. A *model* is a function  $f : \mathcal{S} \rightarrow \mathbf{y}$  such that, for every individual  $\mathbf{x} \in \mathcal{S}$ , we have  $y = f(\mathbf{x})$ .

In practice, and for most of the populations, the set of attributes measured is not sufficient to fully characterize the target variable, meaning that  $f(\mathbf{x})$  do not maps to  $y_k$ , but to something approximated. This fact is modeled in the discipline of machine learning using a random variable, and saying that  $y_k = f(\mathbf{x}) + \varepsilon_k$  where the errors  $\varepsilon_1, \dots, \varepsilon_n$  are independent,  $\mathbb{E}[\varepsilon_k] = 0$ , and  $\text{Var}[\varepsilon_k] < \infty$ .

In practice, for most populations, the measured attributes are not sufficient to fully determine the target variable. This means that  $f(\mathbf{x})$  does not exactly map to the observed value  $\mathbf{y}$ , but rather to an approximation. In the traditional formulation of machine learning, this uncertainty is modeled using a random variable, assuming that

$$y_k = f(\mathbf{x}) + \varepsilon_k,$$

where the errors  $\varepsilon_1, \dots, \varepsilon_n$  are independent random variables, with  $\mathbb{E}[\varepsilon_k] = 0$  and  $\text{Var}[\varepsilon_k] < \infty$ .

Learning algorithms operate on a finite sample of data, called the *training set*, from which they infer an approximation to the unknown function  $f$ .

**Definition F.2.5** A *training data set*, is a finite subset of the population  $\mathcal{S}$ , consisting of a collection of predictor  $\mathbf{X}$  and the corresponding target  $\mathbf{y}$ .

Let  $x_{ij}$  represent the value of the  $j$ -th predictor for the  $i$ -th observation, where  $i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, p$ . Correspondingly, let  $y_i$  represent the target value for the  $i$ -th observation. Then, the training data consist of the pairs  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , where each  $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ .

The objective of supervised learning is to construct a model  $\hat{f} : \mathcal{S} \rightarrow \mathbf{y}$  such that the predicted value  $\hat{y}$  is as close as possible to the true value  $y(\mathbf{x})$ .

**Definition F.2.6** Given a training set  $\mathcal{T}$ , a learning procedure returns an *estimator*

$$\hat{f} : \prod_{i \neq j} \mathcal{D}_i \longrightarrow \mathcal{D}_j.$$

For a new individual  $\mathbf{x} \notin \mathbf{x}_1, \dots, \mathbf{x}_n$ , the predicted target value is

$$\hat{y} = \hat{f}(\mathbf{x}_{-j}),$$

where  $\mathbf{x}_{-j}$  denotes the vector of predictor values, excluding the target.

The challenge for the machine learning algorithms is to ensure that the estimator  $\hat{f}$  generalizes well to the entire population  $\mathcal{S}$ , even though it was constructed using only the training data  $\mathbf{X}$ .

## F.2.1 Parametric vs. Non-parametric Models

Most statistical learning methods can be characterized as either *parametric* or *non-parametric*. This distinction concerns the assumptions we make about the form of the underlying function  $f$  that relates the input variables  $X$  to the output variable  $Y$ .

Parametric methods follow a two-step model-based approach.

- 1 First, we assume a specific functional form for  $f$  (for example, a linear or polynomial function) that depends on a finite set of parameters  $\theta$ .
- 2 Second, we use the training data to estimate the values of these parameters.

By making a strong assumption about the shape of  $f$ , the problem of estimating  $f$  reduces to estimating a relatively small number of parameters.

This makes parametric methods computationally efficient and less data-demanding. However, if the chosen functional form does not match the true (unknown)  $f$ , the model will suffer from *underfitting*: it will be unable to capture the actual structure of the data. On the other hand, choosing a highly flexible parametric form (with many parameters) increases the risk of *overfitting*: the model may follow the noise in the training data too closely, resulting in poor generalization to unseen data.

■ **Example F.7 Linear Regression.** In simple linear regression we assume a linear functional form

$$f(x) = \beta_0 + \beta_1 x,$$

with just two parameters  $\beta_0$  and  $\beta_1$  to be estimated from the data. This strong assumption allows the model to be fitted with very little data, but it cannot capture nonlinear relationships between  $x$  and  $y$ . ■

Non-parametric methods make no explicit assumption about the functional form of  $f$ . Instead, they allow the data to determine the shape of the model. Because they are more flexible, non-parametric methods can in principle approximate a much wider range of functions than parametric ones.

However, this flexibility comes at a cost: non-parametric methods typically require a much larger amount of training data to achieve accurate estimates of  $f$ , and they are often harder to interpret by humans. Moreover, they can also overfit if not properly regularized or if the dataset is too small.

■ **Example F.8  $k$ -Nearest Neighbours (k-NN).** In  $k$ -NN regression, to predict the value of  $Y$  at a new point  $x$ , we take the average of the  $Y$  values of the  $k$  closest training points to  $x$ . There is no assumed functional form: as more data points are added, the estimated function can take on increasingly complex shapes. ■

In short, parametric models are simpler, data-efficient, and easier to interpret but rely on strong assumptions about the form of  $f$ , while non-parametric models are more flexible and expressive but data-hungry and harder to interpret. Choosing between them involves balancing bias (parametric models tend to have higher bias) and variance (non-parametric models tend to have higher variance).

## F.2.2 Generative vs. Discriminative Models

Another important distinction among statistical learning methods is between *generative* and *discriminative* models. These two classes of models differ in what they attempt to learn from the data, and consequently in how they make predictions.

Generative models aim to learn the joint probability distribution  $P(X, Y)$  of the input variables  $X$  and the output variable  $Y$ . From the joint distribution, they can recover the conditional distribution  $P(Y | X)$  used for prediction by applying Bayes' theorem:

$$P(Y | X) = \frac{P(X, Y)}{P(X)}.$$

Learning  $P(X, Y)$  usually involves modeling two components: the prior class probabilities  $P(Y)$  and the class-conditional densities  $P(X | Y)$ . Once these are known, we can generate new synthetic data points  $(X, Y)$ , which is why these models are called “generative.”

Generative models can be advantageous when:

- we want to simulate or generate new data,
- we have very limited data (priors help regularize estimation),
- we care about understanding the underlying data distribution.

However, they may perform worse at prediction when their assumptions about  $P(X | Y)$  are incorrect.

■ **Example F.9 Naïve Bayes Classifier.** Naïve Bayes assumes that the features  $X_1, \dots, X_d$  are conditionally independent given  $Y$ . It estimates  $P(Y)$  and each  $P(X_j | Y)$  separately, and uses Bayes' rule to classify new observations. This model can be trained with very little data and is robust to noise, but its strong independence assumption may be unrealistic. ■

Discriminative models directly learn the conditional distribution  $P(Y | X)$  or a direct decision function  $f: X \rightarrow Y$  without modeling  $P(X)$ . They focus solely on the decision boundary between classes rather than modeling how the data is generated.

This often allows discriminative models to achieve higher predictive accuracy, especially when the amount of training data is large. However, they do not provide a full probabilistic model of the data and cannot generate new samples  $(X, Y)$ .

■ **Example F.10 Logistic Regression.** Logistic regression models the conditional probability of  $Y$  given  $X$  as

$$P(Y = 1 | X = x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}},$$

without making any assumption about the distribution of  $X$ . It directly estimates the decision boundary separating the classes. ■

In essence, generative models learn how the data are produced, while discriminative models learn how to separate the classes. Generative models can be more data-efficient and interpretable but are more prone to model misspecification, whereas discriminative models tend to be more accurate given sufficient data but offer less insight into the data-generating process.

## F.3 Model Accuracy

When assessing the performance of a machine learning model, we want to quantify how close the model's predictions  $\hat{Y} = \hat{f}(X)$  are to the true responses  $Y = f(X) + \varepsilon$ . As introduced in Equation ??, the error term  $\varepsilon$  accounts for factors that have not been included in the model and for inherent randomness in the data.

We can distinguish two conceptually different sources of prediction error:

- The *irreducible error*, given by the variance of  $\varepsilon$ , reflects variability that cannot be explained by any model based on the given predictors.

- The *reducible error* comes from the difference between the true function  $f$  and our estimate  $\hat{f}$ , which can be made smaller by choosing a better model or training procedure.

Formally, assuming that both  $X$  and  $\hat{f}$  are fixed,

$$\begin{aligned} E[(Y - \hat{Y})^2] &= E[(f(X) + \varepsilon - \hat{f}(X))^2] \\ &= (f(X) - \hat{f}(X))^2 + \text{Var}(\varepsilon). \end{aligned}$$

Here, the second term  $\text{Var}(\varepsilon)$  is the irreducible error, which places a lower bound on the achievable prediction error. Unfortunately, this quantity is almost always unknown in practice.

### F.3.1 Bias–Variance Decomposition

More generally, if we view  $\hat{f}$  as a random estimator depending on the particular training sample used, we obtain the *bias–variance decomposition*:

$$E[(Y - \hat{f}(X))^2] = \underbrace{\text{Bias}(\hat{f}(X))^2}_{\text{systematic error}} + \underbrace{\text{Var}(\hat{f}(X))}_{\text{estimation variability}} + \underbrace{\text{Var}(\varepsilon)}_{\text{irreducible error}}.$$

This decomposition makes explicit the tradeoff between bias and variance: simpler models have high bias but low variance, whereas more complex models have low bias but high variance.

### F.3.2 Regression Metrics

A common metric for regression tasks is the *mean squared error*:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{f}(X_i))^2.$$

Minimizing the MSE is equivalent to maximizing the likelihood under a Gaussian noise model. Indeed, if we assume  $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$  independently, then

$$\log P(Y | X, \theta) = -\frac{1}{2\sigma^2} \sum_{i=1}^n (Y_i - \hat{f}(X_i; \theta))^2 + \text{const},$$

so the maximum likelihood estimator (MLE) of the parameters  $\theta$  is the one that minimizes the MSE.

Other common regression metrics include:

- The *root mean squared error* (RMSE) is the square root of the MSE, which brings the error back to the original units of  $Y$ .

- The *mean absolute error* (MAE) is

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{f}(X_i)|,$$

which is more robust to outliers than MSE.

- The *coefficient of determination*  $R^2$  is

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{f}(X_i))^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2},$$

and measures the proportion of variance in  $Y$  explained by the model.

### F.3.3 Classification Metrics

When the response variable  $Y$  is qualitative (categorical), we cannot directly compute squared errors. Instead, we measure how often the predicted class label matches the true class.

A simple and widely used metric is the *misclassification rate*:

$$\text{Error rate} = \frac{1}{n} \sum_{i=1}^n I(Y_i \neq \hat{f}(X_i)),$$

where  $I(\cdot)$  is 1 if its argument is true and 0 otherwise. The *accuracy* is simply 1 minus the error rate.

In imbalanced classification problems, accuracy can be misleading. In such cases we use metrics based on the *confusion matrix*, such as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN},$$

where  $TP$ ,  $FP$ , and  $FN$  are the numbers of true positives, false positives, and false negatives, respectively. The *F1-score* combines them as their harmonic mean:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Our ultimate goal is to build a model that generalizes well to previously unseen data. For this reason, model accuracy should be evaluated not only on the training dataset but also on a separate test dataset. A model with very low training error but high test error is said to *overfit* the training data. Techniques such as cross-validation, regularization, and early stopping are commonly used to detect and reduce overfitting.

## F.4 Minimum Message Length

The *Minimum Message Length* (MML) principle is based on the idea that a good theory, or explanation, for a dataset is a small collection of premises under which the data is not surprising. The best theories are those that are concise, explain most of the data, and achieve high accuracy. An *explanation message* is composed of two parts: the first part comprises all the premises induced from the data, including numerical values; the second part contains all the data that cannot be derived from these premises. The message also assumes the existence of some already known and accepted premises (prior knowledge). Given the prior premises and the message, it should be possible to recover the original dataset. According to the MML principle, theories are not rejected because of contradictory measurements; such contradictions only increase the length of the second part of the message.

In the MML framework, a message is a losslessly encoded version of the original data. The first part of the message contains a probabilistic model of the data, and the second part is the data encoded using this model. Our goal is to find the shortest possible explanation message. If the length of the explanation message is longer than that of the raw data, the theory is considered inadequate.

Bayes' theorem (see Theorem B.3.2) states that the probability  $P(H | E)$  of a hypothesis  $H$  given evidence  $E$  is:

$$P(H | E) = \frac{P(E | H)P(H)}{P(E)}.$$

We are interested in finding the hypothesis  $H$  with the highest posterior probability  $P(H | E)$ , assuming a fixed evidence  $E$ . That is, we want to maximize  $P(E | H)P(H)$ , or equivalently, maximize the joint probability  $P(H \wedge E)$ .

The *Minimum Message Length* principle (MML for short) builds on the fact that, using an optimal code, the length of encoding  $H \wedge E$  as a binary string is given by  $-\log_2 P(H \wedge E)$  (see Theorem D.3.2). That is:

$$\begin{aligned} l(H \wedge E) &= -\log_2 P(H \wedge E) = -\log_2 (P(E | H)P(H)) \\ &= -\log_2 P(E | H) - \log_2 P(H). \end{aligned}$$

Thus, the most probable model  $H$  is the one that allows us to encode  $H \wedge E$  with the shortest possible binary string. This encoded string consists of two parts: a general assertion about the data (the model itself) and a detailed description of the data given that assertion.

Let  $\mathbb{X}$  be the discrete set of all possible datasets,  $\mathcal{X}$  a random variable taking values in  $\mathbb{X}$ , and  $f(X | \theta)$  a probability distribution for  $\mathcal{X}$  parameterized by  $\theta$ .

**Definition F.4.1** Let  $\Theta$  be a discrete set of possible parameters for  $f$ , with prior probability distribution  $h(\theta)$  for  $\theta \in \Theta$ , and let  $\hat{\theta} \in \Theta$  be an inferred parameter. An *assertion* is the encoded version of  $\hat{\theta}$  using an optimal code given the distribution  $h$ .

The length of the assertion using an optimal binary code is  $-\log_2 h(\hat{\theta})$  (see Section D.3). Note that  $\theta$  could be a single scalar or a vector of values. Moreover,  $\theta$  could index more than one family of probability distributions  $f$ .

**Definition F.4.2** Let  $X \in \mathbb{X}$  be a dataset, and  $\hat{\theta} \in \Theta$  an inferred value for the distribution  $f$ . A *detail* is the encoded version of  $X$  using an optimal code given the probability distribution  $f(X | \hat{\theta})$ .

The length of the detail using an optimal binary code is  $-\log_2 f(X | \hat{\theta})$ . In other words, the length of the detail is the negative log-likelihood of  $X$  given  $\hat{\theta}$ .

**Definition F.4.3** Let  $X \in \mathbb{X}$  be a dataset, and  $\hat{\theta} \in \Theta$  an inferred value for the distribution  $f$ . A *message* for the dataset  $X$  given an inference  $\hat{\theta}$  is the concatenation of the assertion for  $\hat{\theta}$  and the corresponding detail for  $X$  given  $\hat{\theta}$ .

The length of a message using an optimal binary code is  $-\log_2 h(\hat{\theta}) - \log_2 f(X | \hat{\theta})$ . This length allows us, for example, to compare the posterior plausibility of two competing explanations or hypotheses  $\hat{\theta}_1$  and  $\hat{\theta}_2$ .

**Definition F.4.4** Let  $X \in \mathbb{X}$  be a dataset. The *Minimum Message Length* of  $X$ , denoted by  $MML(X)$ , is defined as:

$$MML(X) = \arg \min_{\hat{\theta} \in \Theta} (-\log_2 h(\hat{\theta}) - \log_2 f(X | \hat{\theta})).$$

In practice, the actual messages are not physically constructed, since we are only interested in their lengths, not in their content. It is assumed that the sets  $\mathbb{X}$  and  $\Theta$ , and the functions  $f(X | \theta)$  and  $h(\theta)$ , are known a priori, and thus need not be included as part of the encoded message.

■ **Example F.11** Consider an experiment in which we toss a weighted coin 100 times. Denote by 1 a face and by 0 a cross, so that each experiment produces a binary string of length 100. Our collection of all possible datasets is  $\mathbb{X} = \mathcal{B}^{100}$ ,  $\theta$  is a number in the interval  $[0, 1]$ , and the likelihood  $f(X | \theta)$  follows a binomial distribution (that is,  $f(X | \theta) = \theta^n(1 - \theta)^{100-n}$  where  $n$  is the number of faces in  $X$ ). Since we do not know anything about how the coin is weighted, we assume that  $h(\theta)$  is the uniform distribution over  $[0, 1]$  (that is,  $h(\theta) = 1$  for  $\theta \in \Theta$ ). Under these assumptions, the length of a

message for  $X$  given an inferred parameter  $\hat{\theta}$  is:

$$-\log_2 h(\hat{\theta}) - \log_2 f(X | \hat{\theta}) = -n \log_2 \hat{\theta} - (100 - n) \log_2(1 - \hat{\theta}).$$

We are interested in finding the value of  $\hat{\theta}$  that minimizes the length of the encoded version of  $X$ , that is, the minimum message length for  $X$ . ■

A Maximum A Posteriori (MAP) analysis of the experiment in Example F.11 would produce the same estimate for  $\hat{\theta}$  as the Minimum Message Length approach. Moreover, given that  $h(\theta)$  follows a uniform distribution, a Maximum Likelihood Estimation (MLE) approach would yield exactly the same value for  $\hat{\theta}$ . This illustrates that MML generalizes MAP and MLE by making explicit the trade-off between model complexity (assertion length) and goodness of fit (detail length).

## F.5 Minimum Message Length

The *Minimum Message Length* (MML) principle is a Bayesian approach to inductive inference closely related to the Minimum Description Length (MDL) principle introduced in Section ???. While MDL was developed as a practical reformulation of Kolmogorov complexity, MML was originally proposed by Wallace and Boulton in 1968 as a formal method for statistical inference grounded in information theory and Bayesian probability.

Both MDL and MML share the fundamental insight that *learning can be understood as compression*: a good model of the data is one that allows the data to be described concisely. However, while MDL is rooted in coding theory and emphasizes designing universal codes for classes of models, MML is firmly rooted in Bayesian statistics and emphasizes probabilistic inference. In MDL the model cost and the data cost are combined using coding arguments, while in MML they are combined using Bayes' theorem.

Like MDL, MML represents the communication of a dataset  $D = \{x_1, \dots, x_n\}$  as a two-part message:

- 1 The first part describes a hypothesis  $H$  (including its parametric form and numerical parameter values).
- 2 The second part describes the data  $D$  given that the receiver already knows  $H$ .

Formally, the total message length is

$$L(H, D) = L(H) + L(D | H),$$

where  $L(H)$  is the codelength (in bits) of describing the hypothesis, and  $L(D | H)$  is the codelength of the data encoded using that hypothesis.

Under information theory, the Shannon–Fano code assigns an optimal codelength of  $-\log_2 P(D | H)$  bits to data  $D$  when using hypothesis  $H$ . Thus,

$$L(D | H) = -\log_2 P(D | H).$$

The first part  $L(H)$  is determined by the prior probability of  $H$ :

$$L(H) = -\log_2 P(H),$$

where  $P(H)$  is a prior density over hypotheses.

By Bayes' theorem, minimizing the total message length is equivalent to maximizing the posterior probability of  $H$ :

$$P(H | D) \propto P(H)P(D | H) \iff \arg \min_H L(H) + L(D | H).$$

This gives MML a clear Bayesian interpretation: the best model is the one that achieves the greatest compression of the data *and* has the highest prior plausibility.

The term  $L(D | H) = -\log_2 P(D | H)$  is the empirical negative log-likelihood of  $H$  on  $D$ . If the dataset  $D$  is drawn from an unknown true distribution  $P^*$ , then for large  $n$  this term approaches the *cross-entropy* between  $P^*$  and  $P_H$ :

$$\frac{1}{n}L(D | H) \approx H(P^*, P_H) = -\sum_x P^*(x) \log_2 P_H(x).$$

Minimizing this cross-entropy encourages models that assign high probability to the observed data, while the prior term  $L(H)$  penalizes overly complex hypotheses. This provides a natural protection against overfitting: complex models that perfectly fit the sample may have very small  $L(D | H)$  but incur a large  $L(H)$ , whereas overly simple models may have small  $L(H)$  but large  $L(D | H)$ .

■ **Example F.12** Suppose we are given a binary sequence  $D = (x_1, \dots, x_n)$  and consider two candidate hypotheses:  $H_1$ , a Bernoulli model with parameter  $\theta$ , and  $H_2$ , a first-order Markov chain with four parameters.

Under MML, the message length of  $H_1$  would be

$$L(H_1) + L(D | H_1) = -\log_2 P(H_1) - \log_2 P(D | \theta),$$

where  $P(\theta)$  is a prior over the Bernoulli parameter. Similarly, the Markov chain would have

$$L(H_2) + L(D | H_2) = -\log_2 P(H_2) - \log_2 P(D | \theta_{00}, \theta_{01}, \theta_{10}, \theta_{11}).$$

Even if  $H_2$  fits the data better (lower  $L(D | H)$ ), it may be penalized for its greater complexity via  $L(H_2)$ , leading to the selection of the simpler  $H_1$  if the data does not justify the extra parameters. ■

The MML principle states that the best explanation for a dataset is the one that minimizes the total length of a two-part message describing both the model and the data. This connects statistical inference and data compression:

$$\arg \min_H (-\log_2 P(H) - \log_2 P(D | H))$$

which is equivalent to Bayesian MAP estimation expressed in information-theoretic terms. In practice, MML provides a robust framework for model selection, avoiding overfitting by balancing model complexity and goodness of fit, and offering a computable alternative to the ideal but uncomputable notion of Kolmogorov complexity.

### F.5.1 Comparison between MDL and MML

Although the Minimum Description Length (MDL) and Minimum Message Length (MML) principles are closely related and often yield similar practical results, they are conceptually and technically distinct. Both are rooted in information theory and share the central idea that learning is equivalent to compression: the best hypothesis is the one that gives the shortest description of the data. However, they approach this idea from different philosophical and mathematical standpoints.

MDL is rooted in coding theory and algorithmic information theory. It was designed as a computable approximation to Kolmogorov complexity. MML, in contrast, is rooted in Bayesian statistics and formal decision theory. While MDL begins with the design of universal codes over sets of hypotheses, MML starts from a Bayesian prior over hypotheses and applies Bayes' theorem to produce an optimal code.

MDL does not require an explicit prior over models: the description length of a hypothesis is determined by a chosen coding scheme, and priors are only implicit in the choice of that scheme. MML, on the other hand, explicitly incorporates prior probabilities  $P(H)$  over hypotheses, making the prior an integral part of the inference process. This gives MML a clear Bayesian interpretation, while MDL is often regarded as a frequentist or agnostic method.

In its refined form, MDL encodes the data using a *one-part universal code* for the whole model class, which avoids having to specify the parameter values explicitly. MML always uses a *two-part message*: first the model (with its parameter values), then the data given the model. This makes MML closer to Maximum A Posteriori (MAP) estimation, whereas refined MDL is closer to minimax-optimal universal coding.

In MDL, the model description length is a tool to penalize complex models and is not necessarily interpretable as a probability. In MML, the

model description length is explicitly the negative log of the prior probability,  $L(H) = -\log_2 P(H)$ , and the data length is the negative log-likelihood,  $L(D | H) = -\log_2 P(D | H)$ .

In practice, MDL is often used for model selection and structure learning when explicit priors are unavailable or undesirable, while MML is preferred when strong prior knowledge about the models or parameters is available. Both criteria penalize overfitting by trading off model complexity against data fit, but they do so under different philosophical assumptions.

## F.6 Multiobjective Optimization

Multiobjective optimization is the area of mathematics that deals with the problem of simultaneously optimizing two or more conflicting functions. It has been applied in many areas of science, including engineering, economics, and logistics, where there is typically no single solution that simultaneously satisfies all objectives. In such situations, a decision must be made in the presence of trade-offs between the conflicting goals.

From a formal point of view, we are interested in solving the following *multiobjective optimization* problem:

$$\begin{aligned} \text{minimize} \quad & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ \text{subject to} \quad & \mathbf{x} \in \mathbf{S} \end{aligned}$$

where  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, \dots, k$ , are two or more *objective functions*, and the nonempty set  $\mathbf{S} \subset \mathbb{R}^n$  is the *feasible region*, whose elements  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  are *decision vectors*. The image of the feasible region  $f(\mathbf{S}) \subset \mathbb{R}^k$ , denoted by  $\mathbf{Z}$ , is called the *objective region*, and its elements  $\mathbf{z} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))$  are *objective vectors*. In some applications, the feasible region is defined by a collection of inequality constraints  $\mathbf{S} = \{\mathbf{x} \in \mathbb{R}^n \mid g(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_m(\mathbf{x})) \leq 0\}$ .

In this book we will be dealing with nonlinear multiobjective minimization problems, where at least one of the objective functions or constraint functions is not linear. Objective functions can also be incommensurable, that is, measured in different units or on different scales.

Since the objective functions are conflicting, there does not exist a single solution that is optimal with respect to every objective function (the objective region is only partially ordered).

**Definition F.6.1** A decision vector  $\mathbf{x} \in \mathbf{S}$  *dominates* another decision vector  $\mathbf{y} \in \mathbf{S}$  if  $f_i(\mathbf{y}) \leq f_i(\mathbf{x})$  for all  $i \in \{1, \dots, k\}$  and  $f_j(\mathbf{y}) < f_j(\mathbf{x})$  for at least one  $j \in \{1, \dots, k\}$ . An objective vector  $\mathbf{z} \in \mathbf{Z}$  *dominates* another objective vector  $\mathbf{w} \in \mathbf{Z}$  if  $w_i \leq z_i$  for all  $i \in \{1, \dots, k\}$  and  $w_j < z_j$  for at

least one  $j \in \{1, \dots, k\}$ .

Dominance can be studied from the point of view of the decision variable space or the objective space. An objective vector dominates another objective vector if, and only if, its corresponding decision vector also dominates the other decision vector.

We are interested in those objective vectors for which none of its individual components can be improved without deteriorating at least one of the others.

**Definition F.6.2** A decision vector  $\mathbf{x} \in \mathbf{S}$  is *Pareto optimal* if there does not exist another decision vector  $\mathbf{y} \in \mathbf{S}$  such that  $\mathbf{y}$  dominates  $\mathbf{x}$ . An objective vector  $\mathbf{z} \in \mathbf{Z}$  is Pareto optimal if there does not exist another objective vector  $\mathbf{w} \in \mathbf{Z}$  such that  $\mathbf{w}$  dominates  $\mathbf{z}$ .

Pareto optimality can also be studied from the point of view of the decision variable space or the objective space. An objective vector is Pareto optimal if, and only if, its corresponding decision vector is Pareto optimal.

**Definition F.6.3** The set of Pareto optimal solutions, denoted by  $\mathbf{P}_D$ , is called the *Pareto optimal set*. The set of Pareto optimal solutions in the space of objectives, denoted by  $\mathbf{P}_O$ , is called the *Pareto frontier*.

Sometimes, in practice, it is convenient to use a more restrictive definition of optimality, in which we identify those vectors for which there does not exist any other vector that improves over all the components simultaneously.

**Definition F.6.4** A decision vector  $\mathbf{x} \in \mathbf{S}$  is *weakly Pareto optimal* if there does not exist another decision vector  $\mathbf{y} \in \mathbf{S}$  such that  $f_i(\mathbf{y}) < f_i(\mathbf{x})$  for all  $i = 1, \dots, k$ . An objective vector  $\mathbf{z} \in \mathbf{Z}$  is weakly Pareto optimal if there does not exist another objective vector  $\mathbf{w} \in \mathbf{Z}$  such that  $w_i < z_i$  for all  $i = 1, \dots, k$ .

An objective vector is weakly Pareto optimal if its corresponding decision vector is weakly Pareto optimal. Obviously, the Pareto optimal set is a subset of the weakly Pareto optimal set.

■ **Example F.13** In figure F.1 we have depicted a sample of the objective region of a multiobjective optimization problem composed by two real-valued objective functions  $f_1$  and  $f_2$  that we are interested in minimizing. White points are not weakly Pareto optimal since there exist points that improve both components at the same time (for example, point **B** improves point **C** in both functions). Black points are weakly Pareto optimal since there is no point that improves both components at the same time. Point **A** is not Pareto optimal since point **B** improves one component without deteriorating the other. ■

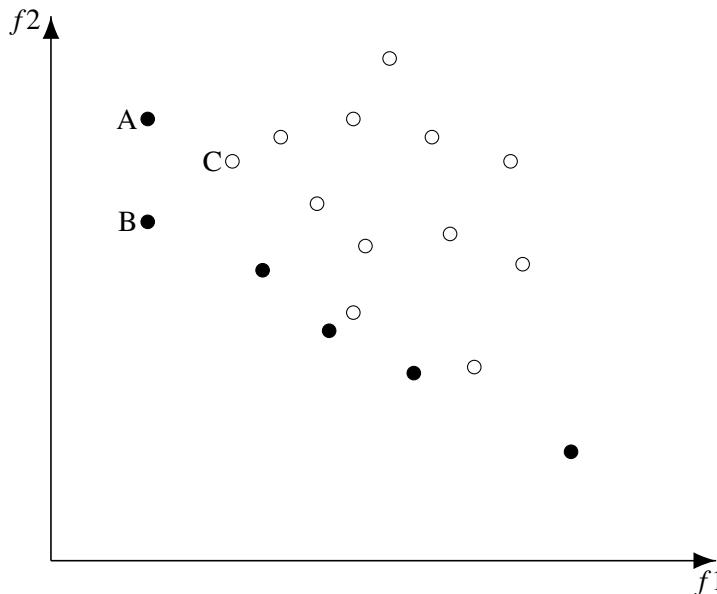


Figure F.1: Pareto optimality.

Mathematically speaking, all the solutions that compose the Pareto optimal set are equally good. However, for most practical applications, it is highly desirable to obtain a single solution. Finding this solution requires additional information not included in the definition of the optimization problem. The relation of preference between objective function values is expressed through a *decision maker*, who is assumed to have additional insights about the problem to be solved. In this sense, solving a multiobjective optimization problem requires finding those feasible decision vectors that are Pareto optimal and that also satisfy the additional requirements imposed by the decision maker.

In practice, we assume that the preferences of the decision maker can be expressed using a value function.

**Definition F.6.5** A *value function* is a function  $U : \mathbb{R}^k \rightarrow \mathbb{R}$  that assigns to each objective vector  $\mathbf{z} = (z_1, \dots, z_k)$  a single real value  $U(\mathbf{z})$ .

Value functions are maximized: among all the Pareto optimal objective vectors, the decision maker selects the one that achieves the largest value of  $U(\mathbf{z})$ . This reflects the idea that the decision maker aggregates multiple criteria into a single utility score and prefers the alternative that yields the highest overall satisfaction. In this way, a multiobjective optimization problem is converted into a single-objective problem over the Pareto frontier.

Value functions allow us to order the vectors of the objective region  $\mathbf{Z}$ . We are interested in applying the value function to the Pareto optimal subset to find a unique solution to the multiobjective optimization problem.

### F.6.1 Range of the Solutions

We are interested in investigating the range of the solutions included in the Pareto optimal set. To do that, we have to find the lower and upper bounds of this set. In the rest of this section, we assume that the objective functions are bounded over the feasible region  $\mathbf{S}$ .

An objective vector that minimizes all objective functions is called an ideal objective vector, while an objective vector that maximizes (within the Pareto set) all objective functions is called a nadir objective vector. These two vectors can be formally defined using the notions of *infimum* and *supremum*.

**Definition F.6.6** The *ideal objective vector*  $\mathbf{z}^* \in \mathbb{R}^k$  is defined componentwise as

$$z_i^* = \inf_{\mathbf{x} \in \mathbf{S}} f_i(\mathbf{x}), \quad i = 1, \dots, k.$$

If there exists an ideal objective vector that belongs to the objective region, that is  $\mathbf{z}^* \in \mathbf{Z}$ , then that vector would be a solution of the optimization problem, and that solution would be unique. In general, the ideal vector is not feasible, but it constitutes a lower bound to the Pareto optimal set.

**Definition F.6.7** The *nadir objective vector*  $\mathbf{z}^{nad} \in \mathbb{R}^k$  is defined componentwise as

$$z_i^{nad} = \sup_{\mathbf{z} \in \mathbf{P}_O} z_i, \quad i = 1, \dots, k,$$

where  $\mathbf{P}_O \subseteq \mathbf{Z}$  is the Pareto frontier.

The nadir vector is the componentwise worst objective value among all Pareto optimal solutions, and it provides an upper bound to the Pareto set. However, unlike the ideal vector, it cannot usually be computed directly, because the full Pareto frontier is unknown. Instead, it is often estimated using the so-called *payoff table*.

**Definition F.6.8** Let  $\mathbf{z}^* = \{z_1^*, \dots, z_k^*\}$  be the ideal objective vector, and let  $\{\mathbf{x}_1^*, \dots, \mathbf{x}_k^*\}$  be the decision vectors that individually minimize each objective function:

$$z_i^* = f_i(\mathbf{x}_i^*), \quad \text{with } \mathbf{x}_i^* \in \arg \min_{\mathbf{x} \in \mathbf{S}} f_i(\mathbf{x}).$$

This collection  $\{\mathbf{x}_1^*, \dots, \mathbf{x}_k^*\}$  is called the *payoff table*.

By construction, each  $\mathbf{x}_i^*$  minimizes  $f_i$  over  $\mathbf{S}$ , but not necessarily the other objectives. Let  $f_{ij}^*$  denote the value of the objective function  $f_i$  computed at  $\mathbf{x}_j^*$ , that is,  $f_{ij}^* = f_i(\mathbf{x}_j^*)$  for  $i, j = 1, \dots, k$ . Using these values, we can estimate the nadir objective vector as follows.

**Definition F.6.9** Given the payoff table  $\{\mathbf{x}_1^*, \dots, \mathbf{x}_k^*\}$ , the *nadir objective vector* is approximated as

$$z_i^{nad} = \max_j f_{ij}^*, \quad i = 1, \dots, k.$$

The ideal objective vector and the nadir objective vector may, or may not, be feasible. In Figure F.2 are depicted the ideal (vector  $\mathbf{E}$ ) and nadir (vector  $\mathbf{F}$ ) vectors of the multiobjective optimization problem of Example F.13.

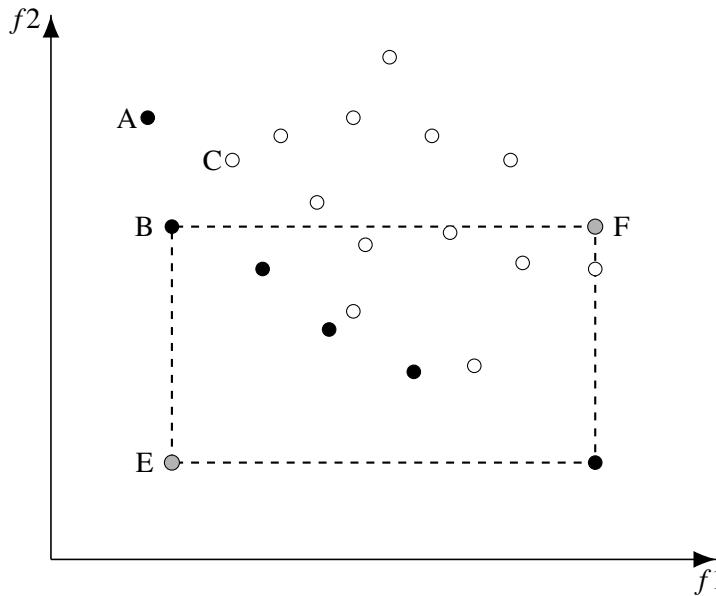


Figure F.2: Ideal and nadir bounds (dashed) over a schematic objective space.

For some applications, the range of values of the objective functions can differ by orders of magnitude. In those situations, it is advisable to normalize them, so the values are in the same scale. We can use the ideal and nadir vector for this normalization process, by replacing each objective function  $f_i(\mathbf{x})(i = 1, \dots, k)$  by the normalized function

$$\frac{f_i(\mathbf{x}) - z_i^*}{z_i^{nad} - z_i^*}$$

## F.6.2 Trade-offs

Since the functions we want to minimize are conflicting, we often have to assume that the only way to gain a benefit in one aspect of the problem is to lose something in another aspect. How much we have to give up in one objective to achieve a certain improvement in another is called a *trade-off*.

**Definition F.6.10** Let  $\mathbf{x}^1, \mathbf{x}^2 \in \mathbf{S}$  be two decision vectors. The *ratio of change* between the functions  $f_i$  and  $f_j$  for the vectors  $\mathbf{x}^1, \mathbf{x}^2$ , denoted by  $\Delta_{ij}$ , is defined as:

$$\Delta_{ij}(\mathbf{x}^1, \mathbf{x}^2) = \frac{f_i(\mathbf{x}^1) - f_i(\mathbf{x}^2)}{f_j(\mathbf{x}^1) - f_j(\mathbf{x}^2)}$$

for all  $i, j = 1, \dots, k$  such that  $f_j(\mathbf{x}^1) - f_j(\mathbf{x}^2) \neq 0$ .

$\Delta_{ij}$  is called a *partial trade-off* between  $f_i$  and  $f_j$  from  $\mathbf{x}^1$  to  $\mathbf{x}^2$  if  $f_l(\mathbf{x}^1) = f_l(\mathbf{x}^2)$  for all  $l = 1, \dots, k, l \neq i, j$ . If  $f_l(\mathbf{x}^1) \neq f_l(\mathbf{x}^2)$  for at least one  $l \neq i, j$ , then  $\Delta_{ij}$  is called a *total trade-off*.

If the trade-off between two objective functions is extremely small or extremely large (that is, if a small change in one aspect of the problem produces a disproportionately large change in another), then the solution behaves similarly to a weakly Pareto optimal solution that is not truly Pareto optimal. In some practical applications, it is convenient to filter out those solutions that exhibit this undesirable behavior. This leads to the concept of *properly Pareto optimal* solutions, which impose a bound on the possible trade-offs between objectives.

**Definition F.6.11** A decision vector  $\mathbf{x} \in \mathbf{S}$  is *properly Pareto optimal* if it is Pareto optimal and if there exists a real number  $M > 0$  such that, for every  $i = 1, \dots, k$  and every  $\mathbf{y} \in \mathbf{S}$  satisfying  $f_i(\mathbf{y}) < f_i(\mathbf{x})$ , there exists at least one  $j \in \{1, \dots, k\}$  such that  $f_j(\mathbf{x}) < f_j(\mathbf{y})$  and

$$\frac{f_i(\mathbf{x}) - f_i(\mathbf{y})}{f_j(\mathbf{y}) - f_j(\mathbf{x})} \leq M.$$

An objective vector  $\mathbf{z} \in \mathbf{Z}$  is properly Pareto optimal if the decision vector corresponding to it is properly Pareto optimal.

Intuitively, a solution is properly Pareto optimal if improving one objective even slightly is possible only at the expense of worsening at least one other objective by a proportionally bounded amount. This rules out extreme cases where an infinitesimal improvement in one objective requires an arbitrarily large deterioration in another.

Note that the properly Pareto optimal set is a subset of the Pareto optimal set, and the Pareto optimal set is a subset of the weakly Pareto optimal set.

### F.6.3 Optimization Methods

Generating Pareto optimal solutions plays a central role in multiobjective optimization. From a mathematical perspective, the problem is considered to be solved when the entire Pareto optimal set has been identified. However, this is often not enough for practical decision-making. In real-world applications, we typically want to obtain a single solution. This means that once the Pareto optimal set is known, we must find a way to impose a complete order on its elements. This requires the participation of a decision maker who provides additional information in the form of a preference structure over the objective space.

A standard way to generate Pareto optimal solutions is through scalarization: converting the multiobjective optimization problem into a single-objective optimization problem, or into a family of such problems. In this approach, the original vector-valued objective function is replaced by a real-valued scalarization function that incorporates the original objectives. To be useful, a scalarization function is usually expected to satisfy the following conditions: i) for every Pareto optimal solution  $\mathbf{x} \in \mathbf{P}_D$ , there exists a choice of parameters of the scalarization function such that  $\mathbf{x}$  is the unique optimal solution of the scalarized problem; ii) every solution produced by the scalarization function is Pareto optimal.

Several scalarization-based methods exist, and they can be classified according to how the decision maker participates in the solution process:

- No-preference methods: no preference information from the decision maker is used.
- A posteriori methods: first a (possibly large) set of Pareto optimal solutions is generated; then the decision maker selects the preferred one among them.
- A priori methods: the decision maker specifies preference information before the optimization is carried out, and this information is used to generate a single preferred solution.
- Interactive methods: the decision maker provides preference information progressively during the optimization process, which iteratively guides the search toward a preferred solution.

In no-preference methods, the knowledge of the decision maker is not taken into account, and the optimization problem is solved using a relatively simple method. In a posteriori methods, the Pareto optimal set (or part of it) is identified first, and then the decision maker selects the preferred solution among the alternatives. A priori methods and interactive methods incorporate

preference information directly into the search, thereby producing solutions more tailored to the decision maker's values.

### Global Criterion

The global criterion method is a no-preference method in which the distance between a given reference point and the feasible objective region is minimized. In this method, all the objective functions are considered equally important. The reference point is usually chosen as the ideal objective vector  $\mathbf{z}^*$ , and a common distance metric is the  $L_p$ -norm. Under these assumptions, the global criterion method becomes the following minimization problem:

$$\begin{aligned} \text{minimize} \quad & \left( \sum_{i=1}^k (f_i(\mathbf{x}) - z_i^*)^p \right)^{\frac{1}{p}} \\ \text{subject to} \quad & \mathbf{x} \in \mathbf{S}. \end{aligned}$$

Different values of  $p$  produce different solutions to this minimization problem. Common choices are  $p = 1$ ,  $p = 2$ , and  $p = \infty$ .

**Proposition F.6.1** The solution of the  $L_p$ -based global criterion problem is Pareto optimal.

*Proof.* Let  $\mathbf{x}$  be a solution of the  $L_p$ -based global criterion problem, with  $1 \leq p < \infty$ , and assume that  $\mathbf{x}$  is not Pareto optimal. Then, by Definition F.6.2, there must exist a point  $\mathbf{y} \in \mathbf{S}$  such that  $f_i(\mathbf{y}) \leq f_i(\mathbf{x})$  for all  $i = 1, \dots, k$ , and  $f_j(\mathbf{y}) < f_j(\mathbf{x})$  for at least one  $j$ . Then we have that  $(f_i(\mathbf{y}) - z_i^*)^p \leq (f_i(\mathbf{x}) - z_i^*)^p$  for all  $i \neq j$  and  $(f_j(\mathbf{y}) - z_j^*)^p < (f_j(\mathbf{x}) - z_j^*)^p$ . Adding all these terms and raising to the  $1/p$  power gives

$$\left( \sum_{i=1}^k (f_i(\mathbf{y}) - z_i^*)^p \right)^{\frac{1}{p}} < \left( \sum_{i=1}^k (f_i(\mathbf{x}) - z_i^*)^p \right)^{\frac{1}{p}},$$

which contradicts the assumption that  $\mathbf{x}$  is a solution to the minimization problem. ■

Although all the solutions selected by the  $L_p$ -based global criterion are Pareto optimal, not all Pareto optimal solutions can be obtained by this method. In practice, it is often convenient to normalize the ranges of the objective functions, so that objectives with values closer to the ideal vector do not dominate the criterion. A common normalization factor used in practice is  $z_i^{nad} - z_i^*$ .

### Weighting Method

The weighting method is one of the simplest and most widely used techniques for generating different Pareto optimal solutions. The idea is to associate each objective function with a weighting coefficient and minimize the weighted sum of the objectives. In this way, the multiple objective functions are transformed into a single objective function. We suppose that the weighting coefficients  $w_i$  are real numbers such that  $w_i \geq 0$  for all  $i = 1, \dots, k$ , and that they are normalized so that  $\sum_{i=1}^k w_i = 1$ . The multiobjective optimization problem is then transformed into the following *weighting problem*:

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^k w_i f_i(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in \mathbf{S}, \end{aligned}$$

where  $w_i \geq 0$  for all  $i = 1, \dots, k$  and  $\sum_{i=1}^k w_i = 1$ .

Weighting coefficients equal to zero should generally be avoided, because they effectively remove an objective from consideration. Furthermore, the objective functions should be normalized or scaled so that their values are approximately of the same order of magnitude; otherwise, the effect of the weights can be misleading and difficult to control. Only when the objectives are scaled comparably can one steer the search toward desirable solutions by adjusting the weights.

**Proposition F.6.2** If all weighting coefficients are strictly positive ( $w_i > 0$  for all  $i = 1, \dots, k$ ), then any solution of the weighting problem is Pareto optimal.

*Proof.* Assume that  $\mathbf{x}^*$  solves the weighting problem but is not Pareto optimal. Then there exists a  $\mathbf{y} \in \mathbf{S}$  such that  $f_i(\mathbf{y}) \leq f_i(\mathbf{x}^*)$  for all  $i$  and  $f_j(\mathbf{y}) < f_j(\mathbf{x}^*)$  for some  $j$ . Since  $w_i > 0$ , multiplying by the weights gives

$$\sum_{i=1}^k w_i f_i(\mathbf{y}) < \sum_{i=1}^k w_i f_i(\mathbf{x}^*),$$

contradicting the optimality of  $\mathbf{x}^*$ . Therefore,  $\mathbf{x}^*$  must be Pareto optimal. ■

If the solution of the weighting problem is unique, it is Pareto optimal even if some weights are zero. However, not all Pareto optimal solutions can be found using the weighting method: it only generates solutions lying on the convex hull of the Pareto frontier. In practice, the condition  $w_i \geq \varepsilon$  with some small  $\varepsilon > 0$  is often used instead of strict positivity, which requires a careful choice of  $\varepsilon$ .

The weighting method can also be used as an a priori method by letting the decision maker specify a weighting vector representing their preferences. In this case, the weighting problem can be viewed as maximizing (the negative of) a linear *value function*. However, it must be noted that changing the weighting coefficients linearly does not imply that the values of the objective functions will change linearly. In particular, if some of the objective functions are correlated, then seemingly “good” weighting vectors may produce poor results, and seemingly “bad” weighting vectors may produce useful results. Furthermore, weighting coefficients are often hard to interpret and understand for average decision makers.

When the weighting method is used as an a priori method, it is sometimes said that the weights reflect the *relative importance* of the objective functions. However, it is more accurate to interpret them as representing the *rates at which the decision maker is willing to trade off* values of the objective functions. This interpretation better captures their intended role.

Finally, using the weighting method as an a priori method implicitly assumes that the decision maker’s underlying value function is (or can be approximated by) a linear function of the objectives. This assumption may be unrealistic in many cases, and it explains why the weighting method can be difficult to control in practice.

### $\varepsilon$ -Constraint Method

The  $\varepsilon$ -constraint method is another classical scalarization technique for generating Pareto optimal solutions. Unlike the weighting method, which combines all objective functions into a single weighted sum, the  $\varepsilon$ -constraint method selects one of the objective functions to optimize and converts the remaining objectives into inequality constraints bounded by user-specified thresholds (the  $\varepsilon$  values).

Let us assume without loss of generality that  $f_1$  is the chosen primary objective function to be minimized. Then the multiobjective optimization problem is transformed into the following  *$\varepsilon$ -constraint problem*:

$$\begin{aligned} & \text{minimize} && f_1(\mathbf{x}) \\ & \text{subject to} && f_i(\mathbf{x}) \leq \varepsilon_i, \quad i = 2, \dots, k, \\ & && \mathbf{x} \in \mathbf{S}. \end{aligned}$$

Different choices of the bounds  $\varepsilon_2, \dots, \varepsilon_k$  produce different solutions. By systematically varying these values, one can generate a set of Pareto optimal solutions.

**Proposition F.6.3** Any optimal solution of the  $\varepsilon$ -constraint problem is Pareto optimal, provided that all the constraints  $f_i(\mathbf{x}) \leq \varepsilon_i$  are active or at least binding for some feasible choice of  $\varepsilon_i$ .

*Proof.* Suppose  $\mathbf{x}^*$  solves the  $\varepsilon$ -constraint problem and is not Pareto optimal. Then there exists  $\mathbf{y} \in \mathbf{S}$  such that  $f_i(\mathbf{y}) \leq f_i(\mathbf{x}^*)$  for all  $i$ , and  $f_j(\mathbf{y}) < f_j(\mathbf{x}^*)$  for at least one  $j$ . In particular,  $f_1(\mathbf{y}) < f_1(\mathbf{x}^*)$  and  $f_i(\mathbf{y}) \leq \varepsilon_i$  for  $i \geq 2$ , meaning that  $\mathbf{y}$  is also feasible and strictly improves the objective. This contradicts the optimality of  $\mathbf{x}^*$ . ■

A key advantage of the  $\varepsilon$ -constraint method is that, unlike the weighting method, it can generate Pareto optimal solutions lying in the non-convex parts of the Pareto frontier. This makes it particularly suitable for problems where the Pareto set is not convex.

In practice, choosing suitable  $\varepsilon_i$  values can be challenging. If they are too restrictive, the problem becomes infeasible; if they are too loose, the solution will simply minimize  $f_1$  with little regard to the other objectives. A common strategy is to estimate lower and upper bounds for each  $f_i$  (for example, the ideal and nadir values defined in Section F.6.1) and then sample values of  $\varepsilon_i$  within these ranges. This allows the decision maker to explore different trade-offs between objectives systematically.

The  $\varepsilon$ -constraint method can be used as a posteriori method (by generating many solutions for different  $\varepsilon$  values and letting the decision maker choose) or as an interactive method (by progressively adjusting the  $\varepsilon_i$  values based on the decision maker's feedback).

## References

Core references for the chapter "Learning"

[Jam+13]: This book introduces modern statistical learning methods with a focus on practical applications. It bridges classical statistical inference and machine learning, making it ideal as an entry point for the chapter's first sections.

[Ped+11]: Scikit-learn is one of the most widely used open-source machine learning libraries. Citing it highlights the connection between theoretical principles and modern implementations that readers can experiment with.

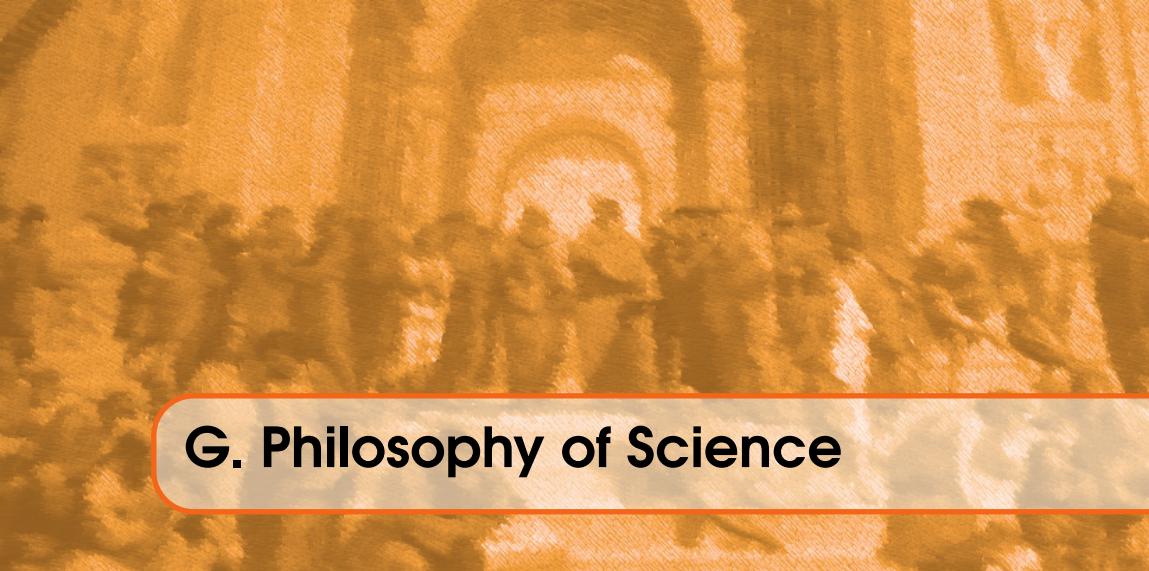
[Grü07]: Grunwald's book is the definitive reference on the MDL principle. It formalizes the idea of balancing model complexity and data fit and provides both theoretical results and practical applications.

[QR89]: This classic paper shows how MDL can be applied to decision tree induction, providing a concrete and accessible example of the principle in action.

[WB68]: The foundational paper introducing the Minimum Message Length (MML) principle. It frames model selection as an optimal coding problem, balancing hypothesis description and data encoding.

[Wal05]: Wallace's comprehensive monograph on MML presents the full theoretical development and applications of the principle, making it the standard reference.

[Mie12]: Miettinen's book is a standard reference on multiobjective optimization. It introduces Pareto optimality, scalarization methods, and decision-making frameworks, providing the mathematical basis for the chapter's last section.



## G. Philosophy of Science

*To go where you don't know,  
you have to go the way you don't know.*

San Juan de la Cruz

The *philosophy of science* is the branch of philosophy that examines the foundations, methods, and implications of scientific inquiry. It explores how scientific knowledge is generated, the validity of scientific theories, the nature of scientific reasoning, and the role of values in science. By addressing fundamental questions about objectivity, reality, and the limits of scientific explanation, the philosophy of science helps us understand how science works and its impact on our understanding of the world.

The philosophy of science provides a theoretical framework for analyzing the core elements that constitute our theory of science. It enables us to critically examine the foundations and assumptions of our theory, guiding us toward the identification of the essential questions that we must address. Moreover, philosophical inquiry compels us to rigorously push the boundaries of our analysis, ensuring that our theory is explored to its ultimate consequences, both logically and conceptually. This approach strengthens the robustness of the theory, even if these results do not directly translate into practical applications.

A central theme in scientific methodology is the distinction between *discovery* and *justification*. Discovery refers to how new ideas or hypotheses emerge, often through creativity, intuition, or serendipity, while justification involves the use of evidence and logical reasoning to evaluate and validate these ideas. Philosophers of science have historically emphasized justification, focusing on methods to rigorously test and analyze scientific claims. However, both discovery and justification are critical to understanding scientific progress, and this chapter will address both aspects.

In this chapter, we provide a concise overview of key elements from the philosophy of science, as well as relevant concepts from other branches of philosophy, such as metaphysics, epistemology, and ontology, that are important to the theory of nescience. Certain other topics within the philosophy of science, such as the problem of objectivity (examining whether science can be truly objective or is influenced by social and personal values) or the role of values in science, including the relationship between ethical, social, and political values and scientific practices, are not included in this review, as they are not directly relevant to a mathematical theory of nescience.

The chapter begins with a brief introduction to the field and its importance in understanding scientific inquiry. We delve into the problem of which entities can be known, examining the scope of scientific knowledge and the nature of observable and unobservable phenomena. The chapter also addresses the concept of scientific representation, discussing how models, theories, and laws reflect aspects of reality. We examine how science discovers new knowledge, the principles behind the scientific method, and the various ways in which scientists formulate and test hypotheses. Finally, we explore the limits of science, considering the boundaries of what science can explain and where its explanatory power may fall short.

## G.1 What is Science

*Science* is a systematic method of investigating the world around us, aimed at generating reliable knowledge through observation and reasoning. Unlike other forms of inquiry, science is rooted in the idea that knowledge must be based on evidence that can be tested and verified. By combining theoretical thinking with empirical data, science offers a powerful way to explain, predict, and understand natural phenomena. Sometimes science seeks explanations for practical purposes, but other times it is sought simply to satisfy our intellectual curiosity.

The *scientific method* is understood as the systematic process by which science acquires new knowledge. In schools, the scientific method is often taught as a series of steps: first observing and describing a phenomenon, then

coming up with a hypothesis to explain it, testing that hypothesis through experiments, analyzing the results, and finally making a conclusion. However, the idea of a single, universal scientific method that can be applied to all sciences isn't widely accepted anymore. Instead, scientists and philosophers recognize that different scientific fields require different methods because each field faces its own challenges and complexities.

The following is a list of the key features that make science distinct and valuable as a way of knowing.

- *Empiricism*: At its core, science is an empirical endeavor, meaning it relies on observation, experimentation, and measurable evidence to understand the natural world. Scientific knowledge is grounded in data that can be gathered through direct or indirect observation, ensuring that claims can be tested and verified by others.
- *Testability*: A key characteristic of science is its focus on developing testable hypotheses, that is, statements or predictions that can be empirically investigated. This means that scientific claims must be falsifiable, open to potential disproof if the evidence does not support them. This distinguishes science from fields that rely on unfalsifiable or speculative claims.
- *Theoretical Frameworks*: Science is not merely about collecting facts; it seeks to develop broader explanations through models, laws, and theories. Models offer simplified representations of complex systems, while theories are well-supported frameworks that explain the underlying mechanisms of phenomena. These frameworks help interpret data and guide further investigation, offering coherence to the body of scientific knowledge.
- *Self-Correction*: A defining feature of science is its self-correcting nature. Scientific theories are not static; they evolve as new evidence comes to light. When new data contradicts a theory, science adapts, modifies, refines, or sometimes discards the theory in favor of one that better fits the evidence. This continuous process of revision ensures that scientific knowledge becomes more accurate over time.
- *Generalizability*: Science strives to uncover universal principles that apply across various contexts, not just to isolated cases. While science begins with specific observations, its goal is to identify general laws and patterns that explain a broad range of phenomena. This pursuit of generalizable knowledge allows science to predict future occurrences and provide deeper insights into the workings of the natural world.

The main task of the philosophy of science is to understand how techniques such as experimentation, observation, and theory building enable scientists to discover the secrets of nature. The philosophy of science draws

on other areas of philosophy, such as ontology, epistemology, metaphysics, and logic. *Ontology*, the branch of philosophy concerned with the nature of existence, addresses the question of which entities exist in the real world. It examines the types of entities that science can study, whether they are observable, abstract, or indeterminate, and engages with the philosophical debates surrounding their existence. Ontology provides the foundation for exploring the boundaries of scientific knowledge and the limits of inquiry. *Epistemology* is the branch of philosophy concerned with knowledge itself—how we acquire it, what justifies it, and what its limits are. While ontology deals with what exists, epistemology addresses how we come to know what exists. In the context of science, epistemology examines the methods, evidence, and reasoning that underpin scientific investigation, exploring how scientific knowledge is built, how reliable it is, and what counts as a justified belief in scientific practice. Both ontology and epistemology are part of the broader field of *metaphysics*, which deals with the fundamental nature of reality and existence. Metaphysics explores the most basic concepts and categories of being, such as time, space, causality, and possibility, as well as the relationship between mind and matter. *Logic*, the study of valid reasoning, underpins the entire scientific enterprise by providing the tools to analyze arguments, identify fallacies, and structure sound reasoning. In science, logic is used to ensure the coherence of theoretical frameworks, the validity of inferences drawn from data, and the consistency of explanations and predictions. Formal logic, including propositional and predicate logic, provides a systematic way to evaluate arguments and detect errors, while informal logic addresses reasoning in everyday scientific discourse. Together, ontology, epistemology, metaphysics, and logic provide a comprehensive philosophical foundation for understanding what science studies, how it builds knowledge, and the fundamental nature of the reality science seeks to explain.

This chapter on philosophy of science is structured around key components that illustrate how scientific knowledge is developed and justified (see Figure G.1). It begins with a discussion on *Entities*, which can be either concrete or abstract, representing the objects of scientific study. From these entities, knowledge is acquired through *Observation*, which involves gathering empirical data and facts. These observations are then transformed into *Representations*, such as recorded data, or facts, that scientists use to analyze and interpret phenomena. Through the process of *Discovery*, these representations lead to the formulation of *Explanations*, which take the form of scientific *theories and laws* that describe underlying principles governing the natural world. The chapter also explores how scientific explanations influence future observations, shaping what is investigated and how data is interpreted. Additionally, thought Experiments provide an alternative way

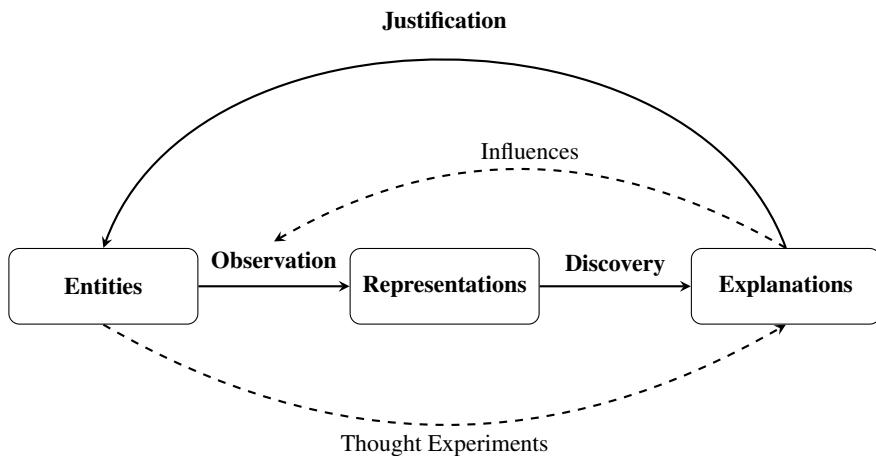


Figure G.1: Chapter Organization

of reasoning about scientific problems beyond direct empirical observation. Finally, the chapter addresses *Justification*, emphasizing the criteria by which scientific claims are validated, ensuring that discoveries and explanations are robust, reliable, and well-supported by evidence. This structure provides a comprehensive view of the scientific process, from the identification of entities to the formation and justification of scientific theories.

## G.2 What is an Entity

In this section, we focus on the fundamental problem of determining which kinds of entities can be known or investigated by science.

A *knowable entity* is an object, phenomenon, or concept that can be investigated, understood, or described through scientific or intellectual inquiry. Knowable entities are those that, either directly or indirectly, can be observed, measured, inferred, or modeled, using available tools, methods, or theories. They are within the scope of human knowledge, and their properties can be analyzed or explained. Examples of knowable entities include the stars and planets, animals, or computer algorithms. A *non-knowledgeable entity* refers to something that cannot be directly observed, measured, or understood using current scientific or intellectual methods. This could be due to limitations in technology, the abstract or metaphysical nature of the entity, or inherent epistemological boundaries. Non-knowledgeable entities might remain beyond the reach of human understanding either temporarily (until methods evolve) or permanently (due to their nature). Examples of non-knowledgeable entities include the nature of consciousness, the origin of the universe, or the existence of a

deity. The distinction between knowable and non-knowable entities is not always fixed, but varies depending on the clarity and precision of the area of interest and the evolving nature of scientific understanding in each field.

**Extend unknowable to consider existing by unknowable and unknowability due to inexistence**

Scientific research encompasses both *concrete entities* and *abstract entities*. Concrete entities refer to physical objects or phenomena that can be directly observed, measured, or interacted with, such as stars, cells, or chemical compounds. These entities form the basis of empirical research, where data is collected through direct observation or experimentation. In contrast, abstract entities are conceptual and do not have a physical presence, such as numbers, algorithms, or theoretical models. Abstract entities play a crucial role in scientific research, particularly in fields like mathematics and theoretical physics, where they provide the framework for understanding and modeling concrete phenomena. While abstract entities cannot be observed directly, they can be known through indirect methods. The inclusion of these abstract entities in scientific inquiry raises important philosophical questions about their ontological status: Are these abstract entities real in the same way that physical objects are, or are they simply conceptual tools? This question remains an open and debated issue in the philosophy of science.

Finally, in scientific inquiry, there is also a distinction between *observable entities*, which can be directly perceived or measured, and *non-observable entities*, which cannot be directly observed but can still be detected or inferred from empirical evidence. Observable entities include things like trees, planets, and bacteria, objects that can be seen or measured using scientific instruments. Non-observable, but still detectable, entities include things like subatomic particles and gravitational forces. These entities are often crucial for explaining observable phenomena and are identified through the use of scientific instruments and theoretical frameworks. For example, we cannot observe a quark in the same way we observe a tree, but through scientific theory, experimentation, and the detection of indirect evidence, we infer its existence.

A central debate within the scope of science is the tension between *reductionism* and *holism*. Reductionism is the view that complex systems can be fully understood by breaking them down into their simplest, most fundamental parts. For example, a reductionist might argue that biological processes can be explained entirely by chemistry, and chemistry by physics. This approach assumes that understanding the smallest components of a system will provide a complete explanation of the whole. In contrast, holism argues that some phenomena cannot be fully understood by reducing them to their components. Instead, the whole system exhibits properties that cannot

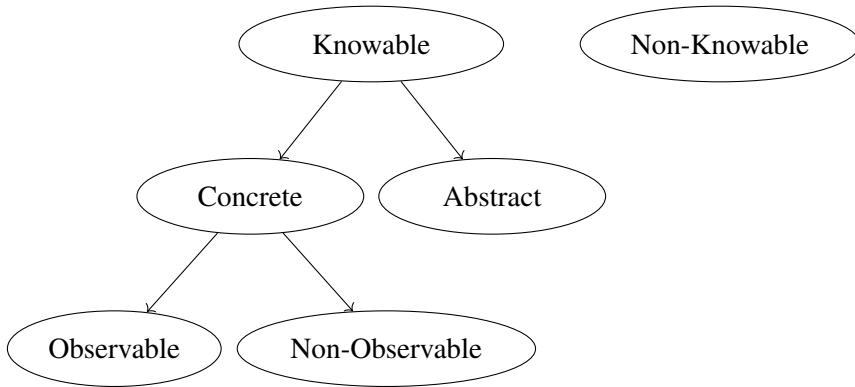


Figure G.2: Classification of Research Entities

be predicted or explained by analyzing its parts in isolation. For example, in ecology, the interactions within an ecosystem can produce emergent properties that are not reducible to the behavior of individual species.

Finally, the scope of science is shaped by the debate between *scientific realism* and *anti-realism*, which addresses the question of whether the entities posited by scientific theories are real or merely useful constructs. The contrast between realism and anti-realism is most marked for sciences which make claims about non-observable entities. Scientific realism holds that the entities described by scientific theories exist independently of our knowledge of them. According to this view, successful scientific theories reveal truths about the world. In contrast, anti-realism argues that scientific theories are useful tools for predicting and organizing observations, but we should not necessarily believe that non-observable entities like electrons or gravitational waves are real. For anti-realists, the purpose of science is not to describe an independent reality, but rather to provide models that help us navigate and predict phenomena.

Scientists often seek to classify the objects they study into general kinds or categories. A key goal of classification is to convey meaningful information, enabling us to better understand and navigate the complexities of the natural world. However, this process raises several intriguing philosophical questions. Do the categories used in science represent real, essential divisions in nature, or are they merely human-made constructs designed to impose order on a complex and often ambiguous reality (realism vs. anti-realism)? Since any set of objects can, in principle, be classified in various ways, how should we determine the most appropriate approach? Is there a 'correct' way to classify, or are all classification systems fundamentally arbitrary? Some philosophers argue that *natural kinds*, groups that correspond to divisions

genuinely existing in the world, do exist, as opposed to merely reflecting human interests. Understanding whether and how such natural kinds exist can provide valuable insights into the structure of reality and inform scientific inquiry.

### G.3 Observation in Science

Scientific knowledge has traditionally been granted a special epistemic status, in part because it is thought to rest on a secure foundation of solid facts firmly established by observation. On this view, facts are taken to be direct, theory-independent deliverances of the senses, immediately accessible to any careful and unprejudiced observer. They are assumed to precede theory and to provide a firm and reliable basis upon which scientific knowledge can be built.

However, there are compelling reasons to question the straightforwardness and security of facts acquired by observation and experiment. What an observer perceives when viewing an object or scene is not determined solely by the images projected onto the retina. Rather, perception is shaped by the observer's prior experience, knowledge, and expectations. One must "learn" to be a competent observer in science, and the perceptual experience of an expert is often markedly different from that of an untrained novice faced with the same situation. This observation challenges a literal understanding of the claim that facts are simply given via the senses. Even when multiple observers view the same scene from the same vantage point, they may interpret what they see in different ways.

Before an observer can formulate and assent to an observation statement, they must already possess an appropriate conceptual framework and know how to apply it. Observation statements presuppose considerable background knowledge. Our search for relevant facts is guided by our current state of knowledge, and this does not undermine the idea that knowledge should ultimately be grounded in facts confirmed by observation. However, it does mean that if the knowledge providing the categories through which we interpret our observations is flawed, the resulting observation statements will be similarly flawed. Judgments about what counts as an observable fact depend on the background of knowledge against which they are made. Indeed, scientific revolutions involve not only the progressive transformation of scientific theories but also a reconfiguration of what are regarded as the observable facts themselves.

This leads to the recognition that facts are fallible and subject to correction. Perceptions are influenced by the expectations and prior knowledge of the observer, so what appears as a straightforward fact to one person may

not be so to another. Judgments about the truth of observation statements depend on what is already known or assumed, rendering observable facts as fallible as the presuppositions that underlie them. Everyday observation is far from passive: observers perform numerous actions, often automatically or unconsciously, to establish the validity of what they perceive. To serve as a foundation for scientific knowledge, observation statements must be formulated in ways that allow their validity to be tested through routine, objective procedures that minimize the need for subjective judgment.

Observations that can function as a basis for science must therefore be both objective and fallible. They are objective to the extent that they can be publicly tested by straightforward procedures, and they are fallible to the extent that they may be overturned by new tests enabled by advances in science and technology. Moreover, science does not require just any facts, but relevant facts. What counts as relevant depends on the current state of development of a science. The world around us involves many interwoven processes that are difficult to disentangle through mere observation of naturally occurring events. To obtain facts relevant for identifying and characterizing particular processes, it is generally necessary to intervene practically: to conduct experiments designed to isolate the process under investigation and eliminate the influence of other processes. If there are facts that constitute the foundation of science, they come primarily in the form of experimental results rather than unstructured observational data.

Yet experimental results are themselves far from straightforwardly given. Their production involves complex practical work, technical know-how, trial and error, and the exploitation of available technologies. Judgments about their adequacy are not simple either: experimental results can be faulty if the knowledge informing their design or interpretation is flawed. They can become obsolete as technology advances, can be rejected when new understanding reveals an experimental set-up to be inadequate, or can be ignored as irrelevant after theoretical shifts. Like observation statements, experimental results are fallible and revisable. Furthermore, they must not only be adequate but also significant: what counts as an adequate or significant experiment depends heavily on the prevailing theoretical and practical context. All experiments presuppose the truth of some theories to judge whether the set-up is appropriate and the instruments are providing the intended readings.

This recognition has led to attempts to formalize the relationship between scientific knowledge and the facts on which it rests. The empiricists maintained that all knowledge should be derived from ideas implanted in the mind through sense perception, while the positivists developed a broader conception of what counts as a fact. Logical positivists, in particular, sought

to rigorously analyze the logical structure of the relationship between scientific knowledge and the observational and experimental facts that are thought to underpin it.

In sum, while observation plays a crucial role in science, its role is more complex and less direct than traditionally assumed. Observational and experimental facts are not simply given; they are theory-laden, constructed through skilled practice, and embedded within conceptual frameworks. They provide a fallible yet indispensable foundation for scientific knowledge, one that must continually be tested, refined, and sometimes replaced as science advances.

#### G.4 Scientific Representation

Science helps us understand the natural world by using different kinds of representations of research entities. These representations include measurements from scientific instruments, descriptions of observations, digital images like X-rays or MRI scans, and more. Scientific practice also often considers mathematical equations, models, and theoretical constructs as valid forms of representation. The challenge of *scientific representation* is to identify the conditions that make a representation scientific and determine what makes an effective representation. The main issues discussed in this area of philosophy include:

*Scientific Representation Problem:* The scientific representation problem is about figuring out the necessary and sufficient conditions that make a representation valid in science. It explores whether these conditions are the same across all scientific fields or if they vary depending on the discipline or research context. For example, what qualifies as a valid representation in physics, which often uses mathematical models, might be different from what is used in biology, where visual and descriptive representations are common. This problem also questions whether scientific representations need to be adapted to specific research goals to be considered valid.

*Representational Demarcation Problem:* The representational demarcation problem looks at whether scientific representations are fundamentally different from other kinds of representations, like those found in art or everyday life. It examines what makes scientific representations unique, focusing on their purpose, accuracy, and the methods used to create them. Unlike artistic representations, which may emphasize subjective interpretation or aesthetic value, scientific representations are generally held to standards of precision, reliability, and empirical adequacy. Understanding these differences helps clarify the specific role that scientific representations play in knowledge production.

*Problem of Style:* The problem of style addresses the fact that the same entity can be represented in different ways, depending on the goals and methods of the research. Different styles of representation—such as diagrams, mathematical equations, physical models, or computer simulations—each have unique characteristics, including the intended audience, level of abstraction, and type of information conveyed. This issue also asks whether these styles are fixed or if new styles can be invented to meet emerging scientific needs. The flexibility of representation styles is crucial because it allows for new insights and different ways of understanding scientific phenomena.

*Standard of Accuracy:* The standard of accuracy problem is about determining what makes a scientific representation accurate. It involves figuring out how to distinguish between accurate and inaccurate representations by considering factors like how well the representation matches empirical data, captures important features of the phenomenon, and its ability to make predictions. This issue also explores whether accuracy should be seen as an objective standard or if it depends on the specific aims and context of the research. For example, a simplified model might still be considered accurate if it effectively serves its purpose, such as making predictions or providing explanations.

*Problem of Ontology:* The problem of ontology in scientific representation deals with the nature of the entities that can serve as representations. It asks whether representations need to be concrete, like physical models or graphs, or if they can also be abstract, like mathematical equations or theoretical constructs. This issue also questions whether representations must be realistic or if more abstract, idealized forms can still be effective in scientific inquiry. Understanding these ontological aspects helps define the types of entities that are allowed in scientific discourse and how they relate to the real-world phenomena they represent.

There are also five conditions of adequacy that a scientific representation should satisfy to be considered effective and reliable:

*Requirement of Directionality:* The requirement of directionality examines the relationship between representations and the real world. Representations are meant to describe entities in the real world, but this condition raises the question of how, if at all, real-world entities might describe their representations. It challenges us to think about the direction of influence between the representation and the entity it aims to depict.

*Surrogate Reasoning:* Surrogate reasoning addresses how scientific representations allow researchers to generate hypotheses about the entities they represent. This condition explores how using a representation as a surrogate can lead to new insights or predictions about the target, effectively using the representation to stand in for the real-world entity during reasoning

and analysis.

*Applicability of Mathematics:* The applicability of mathematics condition is concerned with how mathematical models can be used to represent the real world. It questions how abstract mathematical constructs can effectively describe complex physical systems and whether the success of mathematical representation depends on any special features of the target phenomena. This condition highlights the central role of mathematics in developing and understanding scientific theories.

*Possibility of Misrepresentation:* The possibility of misrepresentation addresses whether representations that are not fully accurate can still be considered valid scientific representations. It considers situations where simplifications or approximations are necessary and whether these less-than-perfect representations can still contribute valuable understanding of a phenomenon. This condition is important for understanding how idealizations and abstractions function in scientific practice.

*Targetless Models:* The targetless models condition explores whether we can allow representations that do not have a direct real-world counterpart. It questions if a model that does not represent any existing entity can still be useful in scientific inquiry, perhaps as a way to explore theoretical possibilities or to understand potential scenarios. This condition emphasizes the creative and exploratory aspects of scientific modeling.

There have been multiple proposals to formally define the concept of scientific representation. Unfortunately, none of these proposals can provide a convincing answer to the questions and conditions of adequacy described above. In the rest of this section, we describe some of these proposals, identifying their advantages and drawbacks. To compare these proposals, we will present them as: "A scientific model  $M$  represents a target system  $T$  if, and only if ...".

*Stipulative Fiat:* The stipulative fiat proposal states that "a scientific model  $M$  represents a target system  $T$  if, and only if, a scientist stipulates that  $M$  represents  $T$ ." The main problem with this interpretation is that, since anything can be a representation if a scientist says so, it is difficult to guarantee the surrogate reasoning condition. If any model can be deemed a representation by simple stipulation, it becomes challenging to determine which representations are genuinely useful for making scientific inferences. Proponents of this theory acknowledge that while all representations may be stipulated, some are undeniably more useful than others.

*Similarity Conception:* The similarity conception proposes that "a scientific model  $M$  represents  $T$  if, and only if,  $M$  and  $T$  are similar." This conception addresses the surrogate reasoning condition since similarity between the model and the target allows us to derive similar properties. How-

ever, it introduces new challenges, particularly regarding the problem of style. The concept of similarity is often vague: in what sense are  $M$  and  $T$  similar? This vagueness can lead to issues with directionality and accuracy, as different aspects of similarity may not always align with what is relevant for scientific representation.

*Structuralist Conception:* The structuralist conception is based on the idea of isomorphism. According to this view, a scientific model  $M$  represents a target system  $T$  if the structure of  $M$  is isomorphic to the structure of  $T$ . In other words, there is a one-to-one correspondence between the elements and relationships in both  $M$  and  $T$ . This approach justifies surrogate reasoning because having the same structure implies that properties and relations in the model correspond to those in the target. Furthermore, since mathematics is fundamentally concerned with the study of structures, this conception also supports the applicability of mathematics in representing natural systems.

*Inferential Conception:* The inferential conception proposes that a model  $M$  is an epistemic representation of a target  $T$  if, and only if, the user adopts an interpretation of  $M$  in terms of  $T$ . This view emphasizes the role of the user in giving meaning to the model, suggesting that representation is not an inherent property of the model itself but arises through its use in making inferences about the target system. This conception underscores the importance of context and interpretation in determining whether a model effectively represents its target.

*Fiction View of Models:* According to the fiction view of models,  $M$  represents  $T$  if and only if  $M$  functions as a prop in a game of make-believe that prescribes imagining certain things about  $T$ . This view draws an analogy between scientific modeling and storytelling, where models are treated as fictional constructs that facilitate imaginative engagement with the target system. Although this approach highlights the creative aspects of modeling, it raises questions about how such fictional constructs can be rigorously linked to real-world entities.

*Representation-As:* The representation-as approach suggests that a scientific model represents a target system as something, emphasizing that representation involves highlighting certain features of the target while downplaying others. This conception focuses on the interpretive aspect of modeling, where the modeler selects specific attributes of the target to represent, depending on the research goals. This approach allows for a flexible understanding of representation that can accommodate different styles and purposes, but it also implies that the usefulness of a representation is contingent on how well the modeler captures the relevant aspects of the target.

Each proposal has its strengths and weaknesses, highlighting the complexity of what it means for a model to effectively represent a target in

scientific inquiry. Understanding these various perspectives is crucial for advancing our comprehension of the role of representation in science.

## G.5 Scientific Discovery

Scientific discovery refers to the process through which new knowledge, ideas, or principles are uncovered within science. Unlike the systematic procedures associated with justification, discovery often involves creativity, intuition, and inspiration. While some discoveries arise from planned experiments or systematic observation, others occur unexpectedly, challenging existing paradigms or opening new fields of inquiry. The general agreement among philosophers is that the creative process of conceiving a new idea is a non-rational process that cannot be formalized as a set of steps. Understanding discovery is crucial for appreciating how science evolves and adapts, as it reveals the dynamic and often unpredictable nature of scientific progress.

The following two proposals assume that a domain-neutral logic of discovery can be formalized, offering attempts to develop such a framework.

- *Discovery as abduction*: Abductive reasoning is a mode of discovery that begins with surprising or anomalous phenomena and seeks to generate plausible hypotheses to explain them. This process is conceptualized as follows: (i) some unexpected data, such as  $p_1, p_2, \dots, p_n$ , is encountered; (ii) these data would be less surprising if a hypothesis of type  $H$  were true; and (iii) therefore, there is justification to develop a hypothesis of type  $H$ . Two types of abduction are distinguished: *selective abduction*, which involves choosing from known hypotheses, and *creative abduction*, which generates entirely new hypotheses. Abduction present some limitations. First, multiple hypotheses may explain the same phenomena, making additional criteria necessary to evaluate their merit. Second, the schema of abductive reasoning does not account for the act of conceiving a hypothesis itself.
- *Heuristic programming* is a computational approach designed to simulate and assist the creative aspects of human problem-solving. These programs operate as searches within a defined problem space, which includes all possible configurations for a given domain. Each configuration represents a specific state within the problem space, with two key states being the initial state, the starting point of the search, and the goal state, which represents the desired outcome. Operators define the moves that transition between states, while path constraints limit permissible moves within the problem space. Problem-solving in this context involves finding a sequence of operations that connects the initial state to the goal state. The core aim of this approach is to de-

velop heuristics (practical rules or strategies) to efficiently navigate and solve complex problems. Heuristic programming has its limitations, scientific problem spaces are often ill-defined, and computer programs rely on experimental data, meaning that simulations frequently cover only specific aspects of scientific discovery.

Many philosophers argue that discovery is an important topic within the philosophy of science, even as they move away from the idea of a formal logic of discovery. A highly influential perspective is Thomas Kuhn's examination of how new facts and theories emerge in the so-called *paradigm shifts*. According to Kuhn, discovery is not a single event but rather a complex and prolonged process that often results in paradigm shifts. Paradigms consist of shared generalizations, theoretical commitments, values, and exemplars that unify a scientific community and shape its research practices. During periods of normal science, research focuses on expanding and refining the existing paradigm rather than pursuing novelty. Discovery typically begins with the recognition of anomalies—phenomena that defy the expectations established by the current paradigm. This process includes observing and conceptualizing the anomaly, followed by revising the paradigm to accommodate it. During paradigm crises, theory-driven discoveries may occur as scientists propose speculative theories, develop new expectations, and conduct experiments or observations to test these ideas. Ultimately, a new paradigm emerges, transforming the once-anomalous phenomena into standard expectations.

Scientific discovery can be also viewed as inherently tied to creativity, with philosophers drawing from cognitive science, neuroscience, computational research, and environmental and social psychology to better understand how new ideas emerge. This perspective aims to demystify the mental processes behind creative thought, emphasizing that scientific creativity can be analyzed and understood philosophically. Central to this analysis are two pivotal cognitive mechanisms: analogies and mental models, which serve as fundamental tools in the generation of innovative ideas and the advancement of knowledge.

- *Analogy*: Analogies play a crucial role in scientific discovery by enabling the transfer of ideas from one domain to another. Philosophers distinguish between three types of analogies: positive, negative, and neutral. Positive analogies involve properties that are shared by both the model (the well-understood domain) and the target domain (the new domain). Negative analogies include properties that belong solely to the model and do not apply to the target domain. Neutral analogies are the most intriguing because they consist of properties whose relevance to the target domain remains uncertain. These neutral analogies

are significant as they often lead to new insights and hypotheses about the less familiar domain. Additionally, there is a distinction between horizontal and vertical analogies. Horizontal analogies connect two domains at the same level of abstraction, whereas vertical analogies involve relationships between different levels of abstraction within the same domain.

- *Model-based reasoning:* The concept of model-based reasoning suggests that much of human thought, including probabilistic and causal reasoning as well as problem-solving, relies on mental models rather than formal logic or strict methodological rules. In this approach, the mind constructs structural representations of real-world or imagined situations and manipulates these models to explore possibilities and generate insights. Conceptual structures are viewed as models, and conceptual innovation involves creating new models using various operations. Analogical reasoning, or analogical modeling, is one of the primary forms of model-based reasoning, alongside visual modeling and simulative modeling, such as through experiments.

## G.6 Scientific Explanation

In philosophy of science it is often made the assumption that there exists a single, distinct type of explanation that qualifies as "scientific." The concept of "scientific explanation" suggests at least two key elements: first, a contrast between explanations characteristic of science and those that are not, and second, a contrast between "explanations" and other forms of discourse, such as mere "descriptions." It is important to note that a set of claims can be true, accurate, and supported by evidence while still failing to qualify as explanatory.

Good scientific explanations are typically evaluated based on several key characteristics. While different philosophical frameworks might emphasize different criteria, the following characteristics are broadly agreed upon:

*Empirical Adequacy:* A good explanation must align with observed and experimental evidence. It should accurately describe the phenomena being explained, providing a detailed account of how the observed data supports the explanation. Additionally, it should integrate well with existing empirical findings, ensuring reliability and fostering broader scientific understanding.

*Logical Coherence:* The explanation should be logically consistent and free of contradictions. Its components should interconnect in a structured and harmonious way, enabling clear and valid reasoning. This coherence ensures that the explanation aligns with established logical principles and provides a solid foundation for understanding the phenomena under investigation.

*Causal Relevance:* Good explanations often identify the causal mechanisms responsible for the phenomenon. They should clarify how and why the phenomenon occurs, detailing the specific interactions and processes involved. By establishing a clear causal link, these explanations provide a framework for understanding not just what happens, but the underlying reasons and mechanisms driving the event. This depth of causality enables predictions, interventions, and broader scientific application.

*Generality:* Explanations that apply to a broader range of phenomena are considered more valuable. They should transcend individual instances to reveal patterns or principles that connect seemingly disparate phenomena. By addressing a wider scope, such explanations enhance our ability to generalize knowledge, foster predictive accuracy, and provide a unifying perspective across different domains of inquiry.

*Simplicity (Parsimony):* A good explanation should not be unnecessarily complex. Among competing explanations, the one that makes the fewest assumptions while still accounting for the phenomena is often preferred (Occam's Razor). This principle emphasizes clarity and efficiency in reasoning, ensuring that explanations avoid superfluous details or unwarranted complexity. Simpler explanations are easier to test, communicate, and apply, fostering a more practical and streamlined understanding of the phenomena.

*Explanatory Depth:* Good explanations go beyond surface descriptions to provide a deeper understanding of the underlying mechanisms, principles, or causes. They delve into the fundamental elements that drive the phenomenon, offering insights into its origins and interconnections with related concepts. By unraveling these deeper layers, such explanations help uncover not just the "how," but the "why," enriching our comprehension and enabling a more profound application of knowledge.

*Testability:* The explanation must allow for predictions that can be tested and potentially falsified. This criterion ensures that the explanation is scientifically meaningful and subject to empirical scrutiny. By being testable, the explanation invites rigorous examination and challenges, which help to confirm its validity or expose its weaknesses. Testability also connects scientific explanations to the broader experimental process, enabling continuous refinement and adaptation in light of new evidence, thereby fostering scientific progress and reliability.

*Unification:* A good explanation often unifies disparate phenomena under a single framework or theory, showing how they are related. It highlights the connections and underlying principles that bring coherence to seemingly unrelated phenomena. By doing so, unification not only simplifies our understanding but also enhances the explanatory power of a theory, allowing for a more integrated and comprehensive perspective on the natural world.

*Use of Laws:* Good explanations often incorporate established scientific laws or theories to provide a robust foundation for their claims. By grounding explanations in well-established principles, they ensure credibility and consistency with the broader scientific framework. This approach not only strengthens the explanatory power of the claims but also facilitates their integration into existing knowledge systems, thereby fostering coherence, predictability, and utility in advancing scientific understanding.

*Practical Applicability:* In some cases, the usefulness of an explanation in solving problems, guiding further research, or applying knowledge in practical ways adds to its value. An explanation with practical applicability not only enhances theoretical understanding but also bridges the gap between abstract knowledge and real-world implementation. It can inform decision-making, inspire technological innovations, and address pressing societal challenges. By demonstrating utility in diverse contexts, such explanations underscore the relevance of scientific inquiry to everyday life and future advancements.

*Asymmetry:* The requirement of asymmetry in scientific explanation is a widely discussed topic in the philosophy of science. According to most philosophical accounts, explanation is indeed considered to be an asymmetric relation, meaning that if X explains Y, it should not also be true that Y explains X under the same laws and additional facts. However, there are debates and complications surrounding this idea. Let's examine the arguments for and against asymmetry in scientific explanations.

*Explanation as prediction:* When we provide a covering law explanation for a phenomenon, the laws and specific facts we reference could have allowed us to predict the phenomenon's occurrence. This highlights that a scientific explanation is inherently a potential prediction. Conversely, every reliable prediction is inherently a potential explanation, illustrating that explanation and prediction are structurally symmetric in their foundational principles.

The following paragraphs briefly introduce the most relevant proposals to characterize what is a scientific explanation.

The *Deductive-Nomological model*, or DN model, emphasizes the importance of deductive reasoning and general laws. According to this model, a phenomenon is explained by demonstrating how it logically follows from a general law combined with specific initial conditions. This explanation comprises two main components: the *explanans*, which includes the general laws and initial conditions, and the *explanandum*, which is the phenomenon to be explained. For the explanation to be valid, the explanans must be true and logically entail (see logical deduction at Section G.7) the explanandum, meaning the phenomenon can be deduced from the laws and conditions

provided. A DN model answers the question "Why did the explanandum occur?" by showing that the phenomenon resulted from specific circumstances  $C_1, C_2, \dots, C_i$ , in conjunction with laws  $L_1, L_2, \dots, L_j$ . For example, the motion of a particular pendulum can be explained by applying Newton's laws of motion (general laws) along with specific details such as the pendulum's length and initial displacement (specific initial conditions). Alternatively, the general behavior of pendulums can also be explained using the same laws and reasoning, since the DN model can be applied to both particular occurrences and general patterns.

The *Statistical-Relevance model*, or SR model focuses on explaining phenomena through statistical relationships rather than strict deductive reasoning. Unlike the Deductive-Nomological model, which requires logical entailment from general laws, the SR model emphasizes the identification of statistically relevant factors that significantly influence the likelihood of a phenomenon. In this model, given some class or population  $A$ , an attribute  $C$  is *statistically relevant* to another attribute  $B$  if and only if  $P(B|A, C) \neq P(B|A)$ . This means that  $C$  affects the probability of  $B$  within the context of  $A$ . An explanation involves identifying such statistically relevant factors and evaluating their impact within a reference class (a group of events or entities sharing common characteristics). For example, in explaining the likelihood of developing a particular disease, an SR explanation might highlight factors such as age, genetic predisposition, or lifestyle choices, showing how these variables alter the probability of the disease occurring. By uncovering these statistical relationships, the SR model provides a method for explaining probabilistic phenomena that cannot be addressed deterministically.

The *Causal-Mechanical model*, or CM model, of scientific explanation emphasizes understanding phenomena by uncovering the underlying causal mechanisms that produce them (see causality at Section G.7). This model asserts that explanations are not just about identifying laws or statistical relationships but about revealing the actual processes and interactions that link causes to effects. A causal-mechanical explanation requires tracing a continuous causal chain, often through detailed physical or biological processes, to show how an event is brought about. For example, explaining the boiling of water involves identifying the causal mechanism: heat energy transfers to the water molecules, increasing their kinetic energy until intermolecular bonds are overcome, leading to a phase change from liquid to gas. The CM model prioritizes clarity in how individual components interact and influence each other, providing a deeper understanding of the phenomenon by grounding it in observable and empirically testable mechanisms. This approach is particularly effective in fields like biology, physics, and engineering, where complex systems and their interactions play a central role in explanation.

The *unificationist account* of scientific explanation, emphasizes the power of explanation through the unification of diverse phenomena under a single, coherent framework of principles and patterns. According to this model, the primary aim of scientific explanation is to reduce the number of independent assumptions and derive a wide range of phenomena from a minimal, consistent set of explanatory patterns. An explanation is considered successful if it contributes to this unifying framework by connecting seemingly disparate observations through common principles. For instance, Newtonian mechanics unifies the motions of celestial bodies and terrestrial objects under the same laws of motion and gravitation. The unificationist approach highlights the importance of simplicity, generality, and coherence in scientific theories, proposing that the value of an explanation lies in its ability to integrate knowledge into an organized, explanatory schema. By offering a comprehensive understanding of diverse phenomena, this account showcases the interconnectedness and systematic nature of scientific inquiry.

The *pragmatic theories* of scientific explanation emphasize the context-dependent and audience-specific nature of explanations, focusing on their purpose and practical utility rather than strict formal structures. These theories argue that explanations are answers to "why" questions posed within a specific context, and their adequacy depends on how well they address the interests and background knowledge of the audience. A scientific explanation, therefore, is not inherently tied to a universal standard but varies depending on the explanatory goals, such as prediction, understanding, or control. For instance, explaining why a bridge collapsed might involve detailed structural analysis for engineers, whereas a simplified account focusing on the immediate cause, like high winds, might suffice for the general public. Pragmatic approaches recognize that explanatory demands can differ across disciplines, situations, and audiences, making the effectiveness of an explanation contingent on its relevance, clarity, and alignment with the inquirer's needs. This perspective underscores the interplay between scientific knowledge and its communication within varied practical contexts.

## G.7 Scientific Justification

Scientific knowledge is often said to be justified by appeal to evidence. However, this justification is not as straightforward or conclusive as it might first appear. It has long been recognized that scientific knowledge can neither be conclusively proved nor conclusively disproved. The role of justification in science is therefore complex: it is always partial, provisional, and open to revision, rather than absolute or final.

Logicians distinguish between two fundamental kinds of reasoning: *de-*

*duction and induction.* Deductive reasoning moves from general premises to conclusions that must be true if the premises are true. Inductive reasoning moves from particular observations to general conclusions, and can lead from true premises to false conclusions. Scientists make use of induction whenever they generalize from a finite number of observations to broader laws, which they do constantly.

This reliance on induction gives rise to what is known as Hume's problem of induction. Any inductive inference assumes that the unexamined cases will resemble the examined ones, an assumption often called the "uniformity of nature." But this assumption cannot itself be proven, and so the rational justification of induction seems to rest on circular reasoning. Karl Popper responded to this difficulty by arguing that induction is not needed at all: science, he claimed, progresses by proposing bold conjectures and then trying to refute them through deductive tests. Although no amount of evidence can prove a scientific theory true, a single well-corroborated counterexample can, in principle, refute it. From this perspective, scientific theories are never justified as true, only as not-yet-falsified.

In practice, scientists also often justify theories by appealing to what is known as inference to the best explanation. When confronted with a body of evidence, they favor the theory that best explains it, typically one that is simpler, more coherent, and unifies disparate observations. This is an inherently comparative and evaluative notion of justification: a theory is considered justified not because it is proven true, but because it currently explains the data better than its rivals.

Much of modern science can be described as following the *hypothetico-deductive method*. A theory is proposed, from which testable predictions are deduced. These predictions are then confronted with observations or experimental results. If the results conflict with the predictions, the theory may need to be rejected or revised; if the results agree, the theory is supported, though never conclusively proven. This illustrates a fundamental feature of scientific justification: positive results support a theory only tentatively, while negative results can count against it. Yet even refutation is not straightforward, because failed predictions may stem from auxiliary assumptions, background theories, or experimental errors rather than from the core theory itself.

This underdetermination of theory by evidence means that evidence alone cannot logically force the acceptance or rejection of a theory. Instead, scientists must weigh multiple considerations when deciding which theory is best justified, including simplicity, explanatory power, and coherence with established knowledge.

Alongside these logical models, science often relies on probabilistic justification. Statistical inference allows scientists to estimate how likely

their data would be if a given hypothesis were true. The Bayesian approach refines this by providing a formal framework for updating the degree of belief in a hypothesis when new evidence is obtained. Each new observation shifts the credibility of a hypothesis up or down, never proving it true but adjusting its level of justification relative to alternatives.

This view makes clear that justification in science is gradual and comparative: theories become more or less justified over time as evidence accumulates. However, it also shows the limits of justification. Bayesian methods assume that all relevant hypotheses are already under consideration, and cannot by themselves account for genuinely novel theories. They also depend on prior assumptions that may be subjective or biased. In large part, then, what counts as “justified” depends on what the scientific community currently considers possible or plausible.

Thomas Kuhn highlighted that justification is not theory-neutral. What counts as evidence, and how evidence is interpreted, depends on the conceptual framework or paradigm within which scientists are working. During periods of normal science, researchers largely take the paradigm for granted and evaluate theories in terms of how well they solve puzzles defined by it. When anomalies accumulate, a period of crisis can lead to a scientific revolution in which the old paradigm is replaced by a new one. Kuhn argued that competing paradigms are often incommensurable, lacking a common standard by which to compare them directly.

This means that the standards of justification themselves can shift when paradigms change. What once counted as strong evidence may later be dismissed as irrelevant, and what was once regarded as speculative may later be seen as well justified. This view challenges the traditional separation between the context of discovery and the context of justification: the criteria for what justifies a theory are historically and socially situated, not timeless or purely logical.

There is no universal algorithm that can mechanically determine which theory is best justified. Scientific virtues such as empirical fit, simplicity, explanatory power, and coherence often trade off against one another, and reasonable scientists may weigh them differently. As Paul Feyerabend argued, the history of science shows that methodological pluralism—even breaking the accepted rules—has sometimes been necessary for progress. Scientific justification is therefore not purely mechanical or formal, but involves the exercise of judgment, creativity, and even a degree of risk-taking.

Finally, justification in science often depends on establishing causal claims. This is usually done not through passive observation but through carefully designed experiments that isolate variables and eliminate confounding factors. Experimental results provide stronger justification because they

actively create the conditions needed to test causal hypotheses. Yet they, too, are interpreted through theoretical assumptions, and their justificatory power depends on the soundness of those assumptions. Even causal claims are therefore not justified by observation alone, but by their integration into a broader theoretical framework.

## G.8 The Limits of Science

Science has been one of the most successful human enterprises, providing powerful tools for understanding and transforming the world. Yet its scope is not unlimited. There are questions science is ill-suited to answer, boundaries it may never cross, and domains where its methods cease to apply. Recognizing these limits is not to diminish science, but to place it in the broader landscape of human inquiry.

Philosophy, for example, pursues questions beyond the reach of empirical investigation. It employs methods such as logical reasoning, thought experiments, and conceptual analysis, which seek clarity about the meaning and boundaries of our concepts rather than empirical predictions. These methods can reveal types of truths—about logic, meaning, or ethics—that lie outside science’s domain. While science asks how the world works, philosophy can ask what existence itself means, or why there is something rather than nothing. Such metaphysical questions appear to fall beyond the scope of empirical methods.

Similarly, science can describe the consequences of actions, but it cannot tell us what we ought to value or how we ought to act. Moral and aesthetic judgments involve norms, not facts. Scientific results can inform ethical debates, but cannot settle them. This highlights a central boundary of science: it can describe what is, but not what ought to be.

Even within its own domain, science cannot explain everything. Any scientific explanation must rely on some fundamental laws or principles; yet those laws cannot themselves be explained by deeper laws without falling into infinite regress. There will therefore always be a point at which science must stop and accept certain principles as unexplained. This does not mean we can say in advance which questions are forever unanswerable, only that some will inevitably be so.

Moreover, science is constrained by what can be observed or indirectly detected. If an entity or phenomenon leaves no observable trace, then it is beyond the reach of empirical science. This raises subtle questions about the ontological status of unobservable entities: scientific theories may refer to them, but their existence can never be confirmed directly.

Another limit of science arises from its division into disciplines. Each

scientific field is designed to explain a particular range of phenomena using its own methods. Physics, for example, studies the fundamental structure of matter, while sociology studies the structure of human societies. It is often assumed that the more fundamental sciences can, in principle, subsume the less fundamental ones—a view called reductionism. Yet many phenomena studied by the higher-level sciences are multiply realized at the physical level, and their behavior emerges from complex interactions not captured by lower-level laws alone. Holism emphasizes that some properties can only be understood at the level of the whole system. This suggests that even complete knowledge of physics would not automatically yield complete knowledge of biology, psychology, or culture.

A further limitation concerns the very nature of scientific knowledge. Scientific realists argue that science aims to describe the world as it truly is, including unobservable entities. Anti-realists reply that theories should be seen as instruments for predicting observable phenomena, not as literal descriptions of reality. One argument for anti-realism is the problem of underdetermination: the same body of empirical data can often be explained by many different, mutually incompatible theories. If evidence underdetermines theory, then perhaps we should suspend judgment about the reality of unobservable entities. Realists respond that not all explanations are equally good, and that virtues like simplicity, coherence, and explanatory power can guide rational choice. Yet the debate shows that even the status of scientific knowledge itself is contested.

Science is often described as objective, yet it is carried out by human beings embedded in cultural and social contexts. Research agendas, funding priorities, values, and personal biases all shape what questions are asked, how data are interpreted, and which results are accepted. This does not mean that science is purely subjective: its methods aim to correct for individual bias through replication, peer review, and open criticism. But recognizing these social influences reveals that scientific objectivity is an ideal rather than an automatic fact. The self-correcting nature of science works only because its claims remain open to challenge and revision.

Another important boundary is the demarcation between science and non-science. Popper famously proposed falsifiability as a criterion: a theory is scientific if it makes predictions that could in principle be shown false. This criterion has been influential, but not universally accepted, as some fields (such as string theory or evolutionary psychology) are difficult to test in practice yet still considered scientific by many. The demarcation problem shows that there is no single, universally agreed rule for what counts as science, and that the borders of science are themselves somewhat fluid and historically contingent.

Finally, some questions may remain permanently beyond science's reach. The ultimate origin of the universe, the nature of dark matter, or the final fate of reality may elude definitive answers. Science depends on its tools, and some phenomena might require conceptual frameworks or observational methods we cannot yet—and perhaps can never—develop. This does not make such questions meaningless, but it shows that science is only one strand in the broader human quest for knowledge.

Even where science can answer a question, it cannot tell us whether answering it is good or bad. The same knowledge that enables genetic engineering or artificial intelligence also raises profound ethical dilemmas about how such technologies should be used. Science can provide the means to achieve our ends, but it cannot choose the ends themselves. For this, we must appeal to ethical and political reasoning beyond the scope of empirical science.

## References

[Cha13]: Chalmers provides a clear and widely used introduction to the philosophy of science, discussing induction, falsifiability, paradigms, and the limits of scientific reasoning. It is ideal as a general reference for framing the main questions addressed in this chapter.

[Pop14]: Popper introduces falsifiability as the demarcation criterion for science and develops his vision of knowledge growth through bold conjectures and refutations. This is central for your treatment of scientific progress and the demarcation problem.

[Van80]: Van Fraassen defends constructive empiricism, arguing that science aims for empirical adequacy rather than truth. This perspective supports your discussion on the limits of science and the interpretation of scientific representations.

[Sup02]: Suppes develops a structuralist account of scientific theories, emphasizing how scientific knowledge can be formalized in terms of models and invariants. This is directly relevant to your sections on scientific representation.

[Mos16]: Mosterín explores the logical and conceptual structure of scientific theories, focusing on how concepts evolve and how theories are built. This supports your account of entities, theories, and the evolution of knowledge.

[LSW13]: Latour's ethnographic study highlights the social processes behind the making of scientific knowledge. Including this reference enriches your treatment of science as a human practice and its limits.

[Nag14]: Nagel gives a concise overview of epistemological issues such

as justification, belief, and truth. This is useful background for connecting philosophy of science with broader epistemology.

?? contains an interesting review of the concept of science, the scientific method, and the role that technology plays in our society. The author proposes that the goal of science should be quality, although the concept of quality is left undefined, and how to reconcile the rational and romantic points of view in science. The book also contains some advice about which is the right state of mind to pursue a scientific problem, and how to deal with the inevitable failures.

# Appendices

H	About Quotes and Photos .....	479
I	Notation .....	493
	Bibliography .....	497
	Index .....	503





## H. About Quotes and Photos

*What we know is little,  
combined with tenacious concentration on a subject  
and what we are ignorant of is immense.*

Pierre-Simon Laplace

This appendix gathers the quotes and photographs that appear at the beginning of each chapter, offering a brief explanation of their origin and meaning. These elements were carefully chosen for their deep symbolic connection to the central themes of the theory of nescience. The quotations capture, in a few words, timeless insights about knowledge, ignorance, discovery, and the creative power of questioning; some of them serving as true sources of inspiration in the development of the theory itself.

The photographs, likewise, are not mere illustrations: each has been selected as a visual metaphor for a key concept discussed in the book, and has been pre-processed to ensure a coherent visual style. Together, these quotes and images form a symbolic framework that accompanies the reader throughout the book, signposting the intellectual journey from the unknown to the known.

## H.1 Chapter Quotes

The quotations collected in the book have been carefully selected for their profound relevance to the theory of nescience. Each captures, in a few words, a key idea about knowledge, ignorance, discovery, or the creative act of questioning, all central to the themes explored throughout this book. Some of these quotes have been a genuine source of inspiration in the development of the theory itself, shaping its concepts and guiding its spirit. They are presented here as intellectual signposts, inviting the reader to reflect on the timeless wisdom that underlies our ongoing quest to illuminate the unknown.

“Perfection is achieved not when there is nothing more to add, but when there is nothing left to take away.”

*Antoine de Saint-Exupéry*

French aviator and author of *The Little Prince*, Saint-Exupéry combined technical discipline with poetic clarity. This quote celebrates the art of refinement: true perfection emerges not from accumulation, but from stripping away the superfluous to reveal the essential. It resonates with the theory of nescience, which seeks to reduce surfeit, eliminating redundant information until only the irreducible core of knowledge remains.

“If presented with a choice between indifferent alternatives, then one ought to select the simplest one.”

*William of Ockham (Occam's Razor)*

The medieval philosopher William of Ockham championed conceptual economy in an age of scholastic excess. His Razor urges us to prefer simplicity when multiple explanations fit the same evidence. This principle aligns with nescience by warning against needless complexity, guiding us to seek the most compact and accurate representations of knowledge.

“We are all agreed that your theory is crazy. The question which divides us is whether it is crazy enough.”

*Niels Bohr*

Bohr, a founder of quantum theory, embraced the paradoxes that reshaped modern physics. His quip reminds us that revolutionary ideas often seem absurd at first. It underscores the need for bold conceptual leaps to break through entrenched assumptions and reveal what lies hidden in the unknown.

“All great work is the fruit of patience and perseverance, combined with tenacious concentration on a subject over a period of months or years.”

*Santiago Ramón y Cajal*

The father of modern neuroscience, Ramón y Cajal spent decades meticu-

lously mapping the nervous system. His words highlight the slow, deliberate nature of deep discovery. For nescience, they affirm that overcoming ignorance is rarely sudden; it demands sustained, focused engagement with the unknown.

“A little inaccuracy sometimes saves tons of explanation.”

*Saki (H. H. Munro)*

Known for his concise, satirical stories, Saki understood the power of brevity. This remark suggests that relentless precision can hinder clarity, and that small simplifications can reveal the bigger picture. It speaks to nescience by acknowledging that small sacrifices in exactness can accelerate the broader reduction of ignorance.

“Everything should be made as simple as possible, but not simpler.”

*Albert Einstein*

Einstein revolutionized physics through elegant, minimalist theories, yet he warned against oversimplification. His maxim captures the delicate balance at the heart of nescience: reducing the length and complexity of our descriptions without introducing errors that distort reality.

“There are known knowns... There are things we don’t know we don’t know.”

*Donald Rumsfeld*

Though a political figure, Rumsfeld articulated a taxonomy of knowledge that became famous beyond its context. He distinguished what we know, what we know we don’t know, and what we don’t even realize we don’t know. This hierarchy mirrors the structure of nescience, which measures progress by transforming unknown unknowns into known unknowns, and eventually into knowns.

“It is not the answer that enlightens, but the question.”

*Eugène Ionesco*

A pioneer of the Theatre of the Absurd, Ionesco used paradox to challenge conventional thinking. His statement celebrates the generative power of questions, which open new conceptual spaces. Nescience places similar emphasis on asking fruitful questions as the first step toward uncovering the hidden unknown.

“There are no difficult problems, only lack of imagination.”

*Antonio García*

A Spanish engineer, García championed creativity as the key to problem-solving. His observation reframes obstacles as failures of perspective rather

than inherent difficulty. This spirit of imaginative thinking is vital to finding new ways to approach what seems unknowable.

“Science may be regarded as the art of data compression.”

*Ming Li & Paul Vitányi*

Pioneers of algorithmic information theory, Li and Vitányi showed that scientific theories compress vast datasets into concise descriptions. Their insight directly underpins science: progress is measured by how much shorter and more accurate our descriptions become as we transform scattered facts into structured knowledge.

“To be surprised, to wonder, is to begin to understand.”

*José Ortega y Gasset*

A leading Spanish philosopher, Ortega y Gasset saw wonder as the spark of understanding. This quote marks the moment when the familiar becomes strange enough to question. That moment, recognizing something as unknown, is the first step in reducing it.

“Mathematics may be defined as the subject in which we never know what we are talking about, nor whether what we are saying is true.”

*Bertrand Russell*

Logician, philosopher, and Nobel laureate, Russell used humor to reveal the abstract nature of mathematics. His quip highlights how mathematical structures often float free of meaning. This echoes nescience’s concern that our representations may be precise yet disconnected from reality, requiring continual testing to ground them.

“The purpose of models is not to fit the data but to sharpen the questions.”

*Samuel Karlin*

Karlin, a mathematician who bridged probability and biology, valued models as tools for inquiry. His statement reframes models not as answers but as catalysts for better questions. This perspective resonates with science, where models are judged not just by accuracy but by how they illuminate unexplored unknowns.

“Wanderer, there is no road; the road is made by walking.”

*Antonio Machado*

One of Spain’s most beloved poets, Machado captured the spirit of exploration. His verse suggests that progress comes from forging paths rather than following them. Reducing ignorance often means inventing entirely new conceptual routes through uncharted intellectual terrain.

“Information is the resolution of uncertainty.”

*Claude Shannon*

The father of information theory, Shannon redefined knowledge as the reduction of uncertainty. This precise definition transformed communication and computing. It also aligns perfectly with nescience, which measures discovery by how much it reduces inaccuracy and brings clarity to the unknown.

“An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem.”

*John Tukey*

A trailblazing statistician, Tukey emphasized exploratory thinking over rote analysis. His maxim warns that asking the wrong question wastes precision. It affirms that choosing the right unknown to explore matters more than perfectly solving a trivial one.

“Some mathematical statements are true for no reason; they’re true by accident.”

*Gregory Chaitin*

Chaitin, a founder of algorithmic information theory, exposed the limits of provability. His remark highlights that some truths are irreducible and beyond explanation. This insight sets a boundary for science: not all unknowns can be conquered, some are unknowable by nature.

“To go where you don’t know, you have to go by a way you don’t know.”

*Saint John of the Cross*

The Spanish mystic and poet portrayed spiritual growth as a journey through darkness. This line captures the essence of venturing beyond the familiar. Reaching the unknown unknown requires abandoning well-worn methods and daring to step into paths yet unimagined.

## H.2 Chapter Pitures

The photographs presented at the beginning of each chapter have been carefully selected for their symbolic significance and their deep connection to the central themes of the theory of nescience. Each image embodies a key idea explored in the book, such as the pursuit of knowledge, the struggle against ignorance, the propagation of understanding across generations, and the continual expansion of the frontiers of the known. Rather than serving as mere decoration, these images act as visual metaphors, inviting the reader to reflect on the broader meaning of the concepts discussed in each chapter.

All the photographs included here are believed to be royalty-free (or at least publicly usable according to their appearance on Google Images). In

addition, every image has been pre-processed using GIMP<sup>1</sup>, the GNU Image Manipulation Program, to enhance their clarity, contrast, and compositional balance, ensuring a consistent visual style throughout the book. This curated selection aims to enrich the reading experience, providing a visual narrative that parallels the intellectual journey from ignorance to understanding.

### The Torch Bearers



The Torch Bearers is an aluminum sculpture by the American artist Anna Hyatt Huntington, donated to the city of Madrid in 1955 and now located on the campus of the Universidad Complutense. The work depicts an old, dying man passing a flaming torch to a vigorous young rider, who carries it forward with determination. The torch symbolizes the light of knowledge, handed from one generation to the

next in humanity's endless pursuit of understanding. As each generation advances the frontier of the known, they also illuminate new regions of the unknown, leaving an enduring legacy for their successors to explore. Huntington also created bronze casts of the sculpture, which can be found in Valencia, Havana, and at several cultural institutions across the United States.

### The Allegory of the Cave



The Allegory of the Cave, conceived by the Greek philosopher Plato, illustrates humanity's struggle to move from illusion to understanding. Prisoners are chained in a cave, able to see only shadows cast on a wall, mistaking these fleeting silhouettes for reality. When one prisoner breaks free and ascends

to the surface, he is at first blinded by the light, but gradually perceives the true forms of things under the sun. This image embodies the essence of the theory of nescience: our initial grasp of the world is partial and distorted, yet through effort, questioning, and discovery we can escape the cave of ignorance and illuminate what was once unknown.

---

<sup>1</sup><http://www.gimp.org>

## Ars Magna



Ars Generalis Ultima, also known as the Ars Magna ("The Ultimate General Art"), was a groundbreaking work published in 1305 by the Spanish philosopher Ramon Llull. In it, Llull described a mechanical method for reasoning, composed of concentric rotating wheels inscribed with fundamental concepts.

By systematically combining these elements, the device aimed to generate all possible logical truths about a given topic (especially to defend and explain Christian beliefs through rational argument). Although primitive by modern standards, Llull's invention represents one of the earliest attempts to use formal logic as a tool for discovery. It stands as a distant ancestor of modern computational thinking and an early expression of the human desire to reduce ignorance by unveiling hidden relations between ideas.

## Galileo's Telescope



Galileo Galilei was an Italian astronomer, physicist, engineer, philosopher, and mathematician who helped ignite the scientific revolution of the seventeenth century. He was among the first to proclaim that the language of nature is written in mathematics, and that its laws can be unveiled through systematic observation and reasoning.

In 1610, Galileo built a telescope and turned it toward the heavens, revealing mountains on the Moon, the moons of Jupiter, and countless stars invisible to the naked eye. These discoveries shattered long-held beliefs and expanded the boundaries of the known cosmos. His telescope thus became a symbol of humanity's enduring drive to pierce the veil of ignorance.

## Athena's Owl



This ancient silver coin depicts the owl of Athena, the Greek goddess of wisdom. Revered as a symbol of knowledge, insight, and scholarly pursuit throughout the Western tradition, the owl embodies the capacity to see through darkness, just as understanding allows us to pierce the unknown.

The German philosopher Hegel famously wrote that "the owl of Athena spreads its wings only with the falling of the dusk," suggesting that philosophy comprehends the meaning of an era only as it fades into the past. In this sense, the owl reminds us that knowledge often arrives after the fact, illuminating what once was obscure.

### Escher's Infinite Stairs



This artwork, known as "Relativity" by M. C. Escher, depicts figures endlessly walking along staircases that defy the laws of geometry and gravity. Each group of figures moves consistently within its own frame of reference, yet the perspectives are incompatible with one another, creating an impossible structure. This paradoxical scene symbolizes the essence of miscoding: when our representations of reality are flawed, they can lead to coherent but fundamentally incorrect interpretations. Like the figures trapped in Escher's world, we may continue along logical paths that never converge on truth, illustrating how mistaken encodings can imprison thought within self-consistent illusions, a central concern in the theory of nescience.

### Train Wreck at Montparnasse



This historic photograph captures the train wreck at Montparnasse Station in Paris in 1895, when an express locomotive overran the buffer stop, crashed through the station wall, and plunged onto the street below. The accident, caused by a combination of mechanical failure and human miscalculation, serves as a striking metaphor for inaccuracy. Even small errors in the models we use to predict and control complex systems can propagate into catastrophic failures when reality does not behave as expected. This image reminds us that inaccurate descriptions of the world not only mislead our understanding but can also derail our reasoning, sometimes with dramatic consequences.

### Alexander Cuts the Gordian Knot



This painting depicts the legendary moment when Alexander the Great confronted the Gordian Knot, an impossibly tangled cord tied to an ancient chariot in the city of Gordium. According to prophecy, whoever could untie the knot would become ruler of Asia. Rather than struggling to unravel its countless coils, Alexander drew his sword and cut through it in a single stroke. This bold act symbolizes the triumph of simplicity over unnecessary complexity, an apt metaphor for surfeit, the burden of excessive and redundant information that obscures true understanding. This image reminds us that insight often comes not from adding more complexity, but from daring to strip it away to reveal the underlying essence of a problem.

### Creation of Adam



This detail from Michelangelo's The Creation of Adam, painted on the ceiling of the Sistine Chapel, captures the moment when the hand of God reaches out to spark life, and with it, the potential for knowledge, into humanity. The near-touching fingers symbolize the latent connection between the divine source of truth and the human capacity to grasp it.

This image embodies the idea that all knowledge exists in principle, yet remains inaccessible until we reach for it. What separates ignorance from understanding is the daring act of extending our minds toward the unknown, bridging the gap between what is given and what is comprehended.

### Rodin's The Thinker



This sculpture, The Thinker by Auguste Rodin, portrays a solitary figure immersed in deep contemplation, his entire body tense with the effort of thought. Far from passive, his posture reveals thinking as an act of creation, shaping new possibilities within the mind. This image symbolizes the vital activity of

discovering interesting questions: reaching beyond what is known, probing the boundaries of understanding, and daring to imagine what has not yet been asked. Every breakthrough begins not with an answer, but with the courage to pose a question that opens a new path into the unknown.

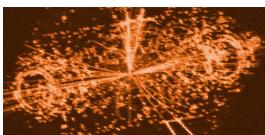
### Rossum's Universal Robots



This photograph shows a scene from R.U.R. (Rossum's Universal Robots), the 1920 play by Karel Čapek that introduced the word robot to the world. In the play, artificial beings are created to perform human labor, eventually surpassing their makers. This early vision of automation resonates

with the modern ambition of machine learning: building systems capable of discovering patterns and generating knowledge with minimal human intervention. The image symbolizes the quest to construct machines that can autonomously reduce our ignorance, relentlessly searching through the unknown to uncover new truths, guided by the principle of minimizing nescience.

### Large Hadron Collider



This image shows the traces of particles produced in a collision inside the Large Hadron Collider (LHC) at CERN, where protons are accelerated to near-light speeds and smashed together to probe the fundamental structure of matter. Each collision generates a burst of data that allows scientists to

test hypotheses, refine theories, and discard false models. This process vividly illustrates how research reduces ignorance: by confronting the unknown through controlled experimentation, science transforms uncertainty into knowledge. Such experiments provide measurable evidence of progress, as each discovery narrows the gap between what is known and what remains to be uncovered.

### Wikipedia Monument



The Wikipedia Monument, located in Ślubice, Poland, was created by the Armenian sculptor Mihran Hakobyan and unveiled in October 2014 as the world's first monument dedicated to the online encyclopedia. The statue honors the countless anonymous editors who, across all cultures and languages, have voluntarily contributed to building one of humanity's most ambitious collective knowledge projects. The inscription celebrates Wikipedia as a symbol of global collaboration beyond political, religious, and cultural boundaries. A foundation for a knowledge society capable of fostering sustainable development, social justice, and peace among nations. This monument stands as a testament to the creative power of many minds working together to illuminate the unknown, piece by piece, like assembling the fragments of an ever-expanding puzzle.

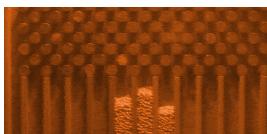
### Königsberg Bridges



The old city of Königsberg in Prussia (now Kaliningrad, Russia) was built on both banks of the Pregel River, with two large islands linked to each other and to the mainland by seven bridges. The famous Seven Bridges of Königsberg problem asked whether it was possible to devise a walk that crossed

each bridge exactly once, starting and ending at any point. In 1736, the Swiss mathematician Leonhard Euler proved that such a walk was impossible. His elegant solution marked the birth of graph theory, a new branch of mathematics that studies the structure of connections, a foundational concept in discrete mathematics and in the formal representation of knowledge.

### Galton's Box



This device, known as a Galton box or quincunx, was invented by the English scientist Francis Galton to demonstrate the principles of probability and the emergence of statistical regularities. Small balls dropped from the top bounce randomly off a series of pegs, eventually accumulating in bins at the

bottom to form the familiar bell-shaped curve of the normal distribution. Although each individual path is unpredictable, the collective outcome is remarkably stable, revealing the hidden order that emerges from chance. In the context of probability theory, the Galton box illustrates how randomness can give rise to predictable patterns, transforming uncertainty into measurable knowledge.

### The Turing Machine



In 1936, the British mathematician Alan Turing introduced a formal model of a hypothetical computing device (later known as the Turing machine) and argued that it could perform any calculation that a human could carry out by following an algorithm.

Turing's model was strikingly simple yet powerful enough to enable rigorous mathematical analysis of computation. Remarkably, many of the principles embedded in his abstract design anticipated the architecture of real computers built a decade later. The Turing machine stands as one of the rare moments in the history of science when theory not only preceded practice, but made it possible, laying the foundations of modern computer science.

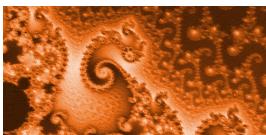
### Morse Key



This Morse key is a manual switching device used to send electrical pulses along a telegraph wire using the Morse code, named after Samuel F. B. Morse, the inventor of the telegraph. Morse code represents letters, numbers, and punctuation through standardized sequences of short and long

signals called dots and dashes. Although not strictly binary (since it also includes symbols for spacing between letters and words) it is remarkably efficient: the average bit length per character in English is about 2.53 bits. This is an extraordinary achievement considering that Morse designed his code intuitively, long before the formal principles of coding theory were developed, making it one of the earliest practical examples of efficient symbolic encoding.

## Mandelbrot Set



The Mandelbrot set is generated by sampling complex numbers and testing, for each point, whether repeatedly iterating a simple mathematical function causes the values to diverge to infinity. By interpreting the real and imaginary parts of each number as image coordinates, pixels are colored according

to how quickly their sequences diverge, while points that never diverge are shown in black. The resulting image reveals an intricate boundary that unfolds into endlessly finer, self-similar patterns under magnification. Despite its visual richness, the Mandelbrot set has very low Kolmogorov complexity, as it arises from an extremely short algorithm, illustrating how simple rules can produce extraordinary complexity in appearance while remaining simple in description.

## Terra Australis Incognita



This antique map depicts a vast southern landmass labeled Terra Australis Incognita, the unknown southern land, a hypothetical continent once believed to balance the known continents of the Northern Hemisphere. Cartographers filled its empty expanse with speculative coastlines and imagined

regions, a testament to how human minds try to complete what they do not yet understand. This map stands as a powerful allegory of the unknown: the regions of knowledge we do not even realize we are missing. Like explorers staring at blank spaces on their charts, we must first recognize that these voids exist before we can begin to illuminate them.

## The School of Athens



This fresco, The School of Athens by Raphael, portrays an idealized gathering of the great thinkers of classical antiquity (Plato, Aristotle, Socrates, Pythagoras, Euclid, and many others) engaged in animated debate and shared inquiry. Painted on the walls of the Vatican in the early 16th century, it

celebrates the collective pursuit of wisdom through dialogue, reasoning, and teaching. The scene embodies the essence of learning as a cumulative human endeavor: knowledge grows not in isolation, but through the exchange of ideas across generations. It symbolizes the collaborative effort to diminish ignorance, as each mind contributes a fragment of understanding to the expanding edifice of human knowledge.

## Ouroboros



The Ouroboros, an ancient symbol depicting a serpent devouring its own tail, represents the eternal cycle of destruction and creation, of endings becoming beginnings. Often associated with alchemy, philosophy, and mysticism, it embodies the self-renewing nature of knowledge, how each resolution

gives rise to new questions, and each discovery reshapes what remains unknown. The Ouroboros serves as a powerful metaphor for the dynamic process of inquiry: as we consume the known, we expose the unknown beyond it, endlessly renewing the pursuit of understanding.

## Gutenberg's Printing Press



The Gutenberg printing press, invented by Johannes Gutenberg in the mid-15th century, revolutionized the production and dissemination of knowledge. By introducing movable type and mechanical printing, Gutenberg transformed books from rare, hand-copied artifacts into mass-produced vehicles of

ideas, enabling knowledge to spread across continents and generations. This invention ignited an unprecedented expansion of learning, accelerating the scientific revolution and reshaping human thought. The printing press symbolizes a turning point where the known began to outpace the unknown, an enduring reminder that advances in how we transmit knowledge can profoundly accelerate our conquest of ignorance.

## Great Library of Alexandria



The Great Library of Alexandria, founded in the 3rd century BCE in ancient Egypt, was one of the most ambitious centers of learning in human history. It is said to have housed hundreds of thousands of scrolls, gathering the accumulated knowledge of the known world under one roof. Though ultimately

destroyed, its legend endures as a symbol of humanity's longing to preserve and unify all understanding. The Library of Alexandria represents both the grandeur and fragility of our efforts to conquer ignorance, an eternal reminder that the pursuit of knowledge is vast, precious, and always vulnerable to loss.

### Tree of Knowledge



The Tree of Knowledge has appeared in many cultural, religious, and philosophical traditions as a symbol of the branching structure of human understanding. From medieval "arbores scientiarum" to moral trees and the Kabbalistic "Etz Chaim", such diagrams attempt to organize all domains of thought

as growing from shared roots and unfolding into diverse branches. They express the idea that knowledge is not a collection of isolated facts, but an interconnected living system where new ideas grow from older ones. The Tree of Knowledge embodies humanity's ongoing effort to map the known while hinting at the vast, unseen forest of the unknown that still lies beyond its outermost branches.



# I. Notation

*When academics encounter a new idea that doesn't conform to their preconceptions,*

*there's often a sequence of three reactions:  
first dismiss, then reject, and finally declare it obvious.*

S. Sloman and P. Fernbach

$\mathbb{N}$  set of natural numbers (including 0)

$\mathbb{Z}$  set of integers

$\mathbb{Z}^+$  set of positive integers

$\mathbb{Q}$  set of rational numbers

$\mathbb{R}$  set of real numbers

$\mathbb{R}^+$  set of positive real numbers

$x \in A$   $x$  is a member of  $A$

: set formation

$A \subset B$   $A$  is a subset of  $B$

$A \subseteq B$   $A$  is a subset or equal to  $B$

$\emptyset$  empty set

$d(A)$  cardinality of  $A$

$A \cup B$  union of  $A$  and  $B$

$A \cap B$  intersection of  $A$  and  $B$

$A \setminus B$	set difference
$\bar{A}$	complement of $A$
$\mathcal{P}(A)$	power set
$(x, y)$	ordered pair
$A \times B$	cartesian product
$A^n$	$n$ -fold cartesian product
$R$	binary relation
$\leq$	total order
$\max(A)$	maximum
$\min(A)$	minimum
$f(x) = y$	function
$f(x) = \infty$	undefined element
$I_A$	identity
$f^{-1}$	inverse function
$f \circ g$	composition
$1_A$	characteristic function
$\text{abs}(x)$	absolute value
$\lceil x \rceil$	ceil
$\lfloor x \rfloor$	floor
$l(s)$	length of string
$\lambda$	empty string
$s^R$	reverse string
$\mathcal{S}^n$	set of strings of length $n$
$\mathcal{S}^+$	set of all finite strings
$\mathcal{S}^*$	set of all finite strings including the empty string
$<_p$	prefix
$\bar{s}$	Self delimited string
$\langle O \rangle$	Encoding as string of $O$
$G = (V, E)$	graph
$\deg(v)$	degree of a vertex
$N(v)$	neighborhood of a vertex
$\text{indeg}(v)$	in-degree of a vertex
$\text{outdeg}(v)$	out-degree of a vertex
$\Omega$	sample space
$P(x)$	probability of $x$
$E(X)$	expectation of $X$
$T$	Turing machine
$Q$	set of states
$\Gamma$	set of tape symbols
$\sqcup$	blank symbol

---

$\Sigma$	input symbols
$q_o$	initial state
$q_f$	final state
$\tau$	transition function
$C$	configuration
$C_o$	initial configuration
$C_f$	final configuration
$U$	universal Turing machine
$\mathcal{E}$	Set of entities
$\mathcal{O}_{\mathcal{E}}$	Representation oracle
$\delta_{\mathcal{E}}$	Representation function
$\mathcal{R}^*$	Set of perfect representations (short for $\mathcal{R}_{\mathcal{O}_{\mathcal{E}}}^*$ and $\mathcal{R}_{\mathcal{E}}^*$ )
$\mathcal{R}_e^*$	Set of perfect representations for $e \in \mathcal{E}$
$\mathcal{R}$	Set of representations (short for $\mathcal{R}_{\mathcal{E}}$ )
$\mathcal{R}_e$	Set of representations for $e \in \mathcal{E}$
$\hat{\mathcal{R}}$	Set of effective representations (short for $\hat{\mathcal{R}}_{\mathcal{E}}$ )
$\hat{\mathcal{R}}_e$	Set of effective representations for $e \in \mathcal{E}$
$\hat{\mathcal{E}}$	Set of effectively known entities
$t \in \mathcal{T}$	Research topic
$d \in \mathcal{D}_t$	Description of a topic
$\mathcal{D}$	Set of descriptions
$\mathcal{D}_{\mathcal{T}}$	Set of valid descriptions of $\mathcal{T}$
$\mathcal{D}_t$	Set of descriptions of $t \in \mathcal{T}$
$\delta$	Description function
$d_t^*$	Perfect description of $t \in \mathcal{T}$
$d_{t,s}$	Joint description of $t, s \in \mathcal{T}$
$\mathcal{D}_{t,s}$	Set of joint descriptions of $t, s \in \mathcal{T}$
$d_{t,s}^*$	Perfect joint description of $t, s \in \mathcal{T}$
$d_{t s^*}$	Conditional description of $t$ given $s, t, s \in \mathcal{T}$
$\mathcal{D}_{t s^*}$	Set of conditional descriptions of $t$ given $s, t, s \in \mathcal{T}$
$d_{t s^*}^*$	Perfect conditional description of $t$ given $s, t, s \in \mathcal{T}$
$A \subset \mathcal{T}$	Research area
$\hat{A}$	Know subset of the area $A \subset \mathcal{T}$
$\mathcal{D}_{\hat{A}}$	Description of the area $A$ given the known subset $\hat{A}$
$\mathcal{D}_{\hat{A}}$	Set of descriptions of the area $A$ given the known subset $\hat{A}$
$d_{\hat{A}}^*$	Perfect description of the area $A$ given the known subset $\hat{A}$
$RG$	relevance graph
$R_t$	relevance of topic $t$
$IP_t$	interestingness of topic $t$ as a problem
$M_t$	maturity of topic $t$

$AG$  applicability graph

$A_t$  applicability of topic  $t$

$IT_t$  interestingness of topic  $t$  as a tool

$T'$  set of known topics

$Q_{t_1 \rightarrow t_2}$  interesting question

$IQ_{t_1 \rightarrow t_2}$  interestingness of question  $Q_{t_1 \rightarrow t_2}$

$\mathbb{F}$  unknown frontier

$\mathbb{S}$  new topics area

$S_{\{t_1, t_2\}}$  new topic

$IS_{\{t_1, t_2\}}$  interestingness of a new topic

$IT_A$  interestingness of an area as tool

$IP_A$  interestingness of an area as problem

$\hat{i}(\hat{\mathbf{y}}, \mathbf{y})$  inaccuracy of predicted values

# Bibliography

## Books

- [Abr63] Norman Abramson. *Information theory and coding*. 1963 (cited on pages 130, 393).
- [Cal02] Cristian S Calude. *Information and randomness: an algorithmic perspective*. Springer Science & Business Media, 2002 (cited on pages 154, 414).
- [Cha13] Alan F Chalmers. *What is this thing called science?* Hackett Publishing, 2013 (cited on pages 48, 172, 254, 280, 475).
- [Chi13] Timothy Childers. *Philosophy and probability*. Oxford University Press, USA, 2013 (cited on page 351).
- [Coo03] S Barry Cooper. *Computability theory*. CRC Press, 2003 (cited on page 369).
- [Cor+90] Thomas H Cormen et al. *Introduction to algorithms mit press*. MIT Press, 1990 (cited on page 299).
- [CT12] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012 (cited on pages 48, 154, 393, 414).

- [DeG+86] Morris H Morris H DeGroot et al. *Probability and statistics*. 04; QA273, D4 1986. 1986 (cited on page 351).
- [Fer09] Maribel Fernández. *Models of Computation: An Introduction to Computability Theory*. Springer Science & Business Media, 2009 (cited on page 369).
- [GG12] Allen Gersho and Robert M Gray. *Vector quantization and signal compression*. Volume 159. Springer Science & Business Media, 2012 (cited on page 393).
- [GV13] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013 (cited on page 299).
- [Grü07] Peter D Grünwald. *The minimum description length principle*. MIT press, 2007 (cited on pages 154, 221, 449).
- [HH10] Ifan Hughes and Thomas Hase. *Measurements and their uncertainties: a practical guide to modern error analysis*. OUP Oxford, 2010 (cited on page 116).
- [Jam+13] Gareth James et al. *An introduction to statistical learning*. Volume 112. Springer, 2013 (cited on page 449).
- [Jec13] Thomas Jech. *Set theory*. Springer Science & Business Media, 2013 (cited on page 90).
- [Joh09] Richard Johnsonbaugh. *Discrete mathematics*. Pearson, 2009 (cited on page 299).
- [LV13] Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer Science & Business Media, 2013 (cited on pages 47, 90, 153, 254, 280, 393, 414).
- [Mie12] Kaisa Miettinen. *Nonlinear multiobjective optimization*. Volume 12. Springer Science & Business Media, 2012 (cited on pages 154, 450).
- [Mos16] Jesús Mosterín. *Conceptos y teorías en la ciencia*. Alianza, 2016 (cited on pages 254, 475).
- [Nag14] Jennifer Nagel. *Knowledge: A very short introduction*. OUP Oxford, 2014 (cited on page 475).
- [PM19] Judea Pearl and Dana Mackenzie. *The Book Of Why: The New Science of Cause and Effect*. Penguin Science, 2019 (cited on page 172).
- [Pop14] Karl Popper. *Conjectures and refutations: The growth of scientific knowledge*. routledge, 2014 (cited on pages 117, 154, 172, 254, 280, 475).

- 
- [Rob15] Borut Robič. *The foundations of computability theory*. Springer, 2015 (cited on page 90).
- [Sip12] Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 2012 (cited on pages 47, 299, 369).
- [Soa16] Robert I Soare. *Turing computability*. Springer, 2016 (cited on page 369).
- [Van80] Bas C Van Fraassen. *The scientific image*. Oxford University Press, 1980 (cited on pages 102, 173, 254, 475).
- [Wal05] Christopher S Wallace. *Statistical and inductive inference by minimum message length*. Springer Science & Business Media, 2005 (cited on pages 221, 450).

## Articles

- [CAO+05] Manuel Cebrián, Manuel Alfonseca, Alfonso Ortega, et al. “Common pitfalls using the normalized compression distance: What to watch out for in a compressor”. In: *Communications in Information & Systems* 5.4 (2005), pages 367–384 (cited on pages 255, 280).
- [Cha69] Gregory J Chaitin. “On the simplicity and speed of programs for computing infinite sets of natural numbers”. In: *Journal of the ACM (JACM)* 16.3 (1969), pages 407–422 (cited on pages 154, 414).
- [Cha95] Gregory J Chaitin. “The berry paradox”. In: *Complexity* 1.1 (1995), pages 26–30 (cited on pages 90, 154).
- [Göd31] Kurt Gödel. “Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I”. In: *Monatshefte für mathematik und physik* 38.1 (1931), pages 173–198 (cited on page 90).
- [Hua13] Keguo Huang. “Three hundred years of the St. Petersburg paradox”. In: (2013) (cited on page 352).
- [Ioa05] John PA Ioannidis. “Why most published research findings are false”. In: *PLoS medicine* 2.8 (2005), e124 (cited on pages 103, 254, 280).
- [Kol65] Andrei N Kolmogorov. “Three approaches to the quantitative definition of information”. In: *Problems of information transmission* 1.1 (1965), pages 1–7 (cited on pages 154, 413).

- [LSW13] Bruno Latour, Jonas Salk, and Steve Woolgar. “Laboratory life: The construction of scientific facts”. In: (2013) (cited on pages 103, 254, 280, 475).
- [Li+04] Ming Li et al. “The similarity metric”. In: *IEEE transactions on Information Theory* 50.12 (2004), pages 3250–3264 (cited on pages 254, 280).
- [Llo82] Stuart Lloyd. “Least squares quantization in PCM”. In: *IEEE transactions on information theory* 28.2 (1982), pages 129–137 (cited on page 393).
- [McM56] Brockway McMillan. “Two inequalities implied by unique decipherability”. In: *IRE Transactions on Information Theory* 2.4 (1956), pages 115–116 (cited on page 393).
- [Ped+11] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pages 2825–2830 (cited on page 449).
- [Pos46] Emil L Post. “A variant of a recursively unsolvable problem”. In: *Bulletin of the American Mathematical Society* 52.4 (1946), pages 264–268 (cited on page 369).
- [QR89] J Ross Quinlan and Ronald L Rivest. “Inferring decision trees using the minimum description lenght principle”. In: *Information and computation* 80.3 (1989), pages 227–248 (cited on pages 221, 449).
- [Shm+10] Galit Shmueli et al. “To explain or to predict?” In: *Statistical science* 25.3 (2010), pages 289–310 (cited on pages 172, 221).
- [Sol64] Ray J Solomonoff. “A formal theory of inductive inference. Part I and II”. In: *Information and control* 7.1 (1964), pages 1–22 (cited on pages 154, 414).
- [Sup02] Patrick Suppes. “Representation and invariance of scientific structures”. In: (2002) (cited on pages 102, 154, 254, 280, 475).
- [TAY22] JR TAYLOR. “An introduction to error analysis: The study of uncertainties in physical measurements (3rd ed. edition).” In: (2022) (cited on page 116).
- [Tur36] Alan Mathison Turing. “On computable numbers, with an application to the Entscheidungsproblem”. In: *J. of Math* 58.345–363 (1936), page 5 (cited on pages 154, 368).
- [Tur39] Alan Mathison Turing. “Systems of logic based on ordinals”. In: *Proceedings of the London Mathematical Society* 2.1 (1939), pages 161–228 (cited on page 90).

- [WB68] Chris S Wallace and David M Boulton. “An information measure for classification”. In: *The Computer Journal* 11.2 (1968), pages 185–194 (cited on page 449).
- [WP93] Chris S Wallace and JD Patrick. “Coding decision trees”. In: *Machine Learning* 11.1 (1993), pages 7–22 (cited on page 221).



# Index

- $P \stackrel{?}{=} NP$ , 355  
 $\sigma$ -algebra, 305
- Abductive reasoning, 464  
Absolute frequency, 390  
Absolute value, 288  
Abstract entities, 456  
Adjacency matrix, 296  
Adjacent vertices, 295  
Alphabet, 288  
Analogy, 465  
Ancestors of a vertex, 297  
Anti-Pareto optimal hypothesis, 150  
Anti-realism, 457  
Antisymmetric relation, 286  
Applicability, 161  
Applicability of mathematics, 462  
Axiom of Choice, 53  
Axiom of Comprehension, 53  
Axiom of Separation, 53  
Axiomatic interpretation of proba-  
bility, 304  
Backus-Naur form, 291  
Balanced tree, 298  
Bayes' theorem, 312, 326, 423  
Bayesian inference, 423  
Bayesian interpretation of probabil-  
ity, 303  
Bernoulli distribution, 337  
Bernoulli process, 338  
Bernoulli trial, 338  
Berry paradox, 32, 77  
Bijective function, 287  
Binary relation, 286  
Binary sequence, 288  
Binary tree, 297  
Binomial coefficient, 292  
Binomial distribution, 338  
Bipartite graph, 297  
Bit, 383  
Blank Symbol, 356, 365  
Branches of a tree, 297

- Breadth-first tree traversal, 298  
Burrows-Wheeler algorithm, 229  
bzip2, 229
- Candidate topic combination, 171  
Cantor's theorem, 27, 53  
Cardinality of a set, 284  
Cartesian product, 286  
Categorical attribute, 427  
Causal relevance, 467  
Causal-Mechanical model, 469  
Ceil, 288  
Central limit theorem, 350  
Certain event, 304  
Cervantes, 57  
Characteristic function, 287  
Chebyshev's inequality, 346  
Child of a vertex, 297  
Chomsky hierarchy, 290  
Church-Turing thesis, 354  
Class interval, 391  
Class interval width, 391  
Class limits, 391  
Class mark, 391  
Classical interpretation of probability, 302  
Classification problems, 428  
Code, 372  
Code alphabet, 372  
Code efficiency, 382  
Code extension, 373  
Code word, 372  
Codomain, 287  
Column matrix, 293  
Combinations, 292  
Combinatorics, 291  
Compact code, 380  
Comparable oracle, 60  
Complement of a set, 285  
Complete code, 383  
Complexity classes, 354
- Composition, 287  
Computable function, 363  
Computable procedure, 353  
Computable set, 364  
Computably enumerable set, 364  
Computation, 359  
Computational complexity, 354  
Concrete entities, 456  
Conditional inaccuracy, 109  
Conditional independence, 312  
Conditional independence of random variables, 327  
Conditional Kolmogorov complexity, 404  
Conditional model, 83, 85  
Conditional probability, 309  
Conditional probability mass function, 325  
Conditional surfeit, 125  
Configuration, 358  
Configuration yields configuration, 358  
Connected graph, 296  
Context sensitive grammar, 290  
Context-free grammar, 290  
Continuous attribute, 427  
Convergence in probability, 347  
Correlation, 335  
Countable set, 287  
Countably many set, 287  
Covariance, 334  
CRISPR, 245  
Cumulative distribution function, 317  
Cumulative frequency, 392  
Current best description, 147  
Current best hypothesis, 146  
Cycle, 296
- De Morgan's laws, 285  
Deduction, 471  
Deductive-Nomological model, 468

- Deep learning, 245  
Degree of a vertex, 296  
Degree sum formula, 297  
Demarcation problem, 44, 248  
Depth of a vertex, 297  
Depth-first tree traversal, 298  
Descendant of a vertex, 297  
Description, 26, 32, 77  
Description function, 78  
Diagonal matrix, 294  
Directed graph, 296  
Discrete attribute, 427  
Discrete normal distribution, 340  
Discrete random variable, 314  
Discrete random vector, 320  
Disjoint sets, 285  
Distribution of the sum, 322  
DNA, 30, 56, 64  
Domain, 287  
Dominant, 134  
Dutch book, 303
- Edges of a graph, 295  
Effective representation, 68  
Effectively known, 68  
Effectively unknown, 68  
Eigenvalue, 295  
Eigenvector, 295  
Element of a set, 284  
Empirical adequacy, 466  
Empiricism, 453  
Empty language, 289  
Empty set, 284  
Empty string, 288  
Encoding function, 29  
Endpoints of an edge, 295  
Entities, 52  
Entity, 26, 27  
Entropy, 383  
Epistemology, 54, 454  
Equal sets, 284
- Equivalence class, 286  
Equivalence relation, 286  
Equivalent elements, 286  
Estimate, 418  
Estimator, 418, 429  
Euclidean decision rule, 166  
Event, 304  
Expected length of a code, 380  
Expected value, 328  
explanandum, 468  
explanans, 468  
Explanatory depth, 467  
Extension string, 72  
Extensions of representations, 71
- Fiction view of models, 463  
Field of sets, 285  
Final State, 365  
Fixed-length code, 376  
Floor, 288  
Free monoid, 289  
Frequency distribution, 391  
Frequentist interpretation of probability, 303  
full k-ary tree, 297  
Function, 287
- Generalizability, 453  
Grammar, 290  
Graph, 81, 295  
Graphene, 245  
Gödel numbering, 54
- Halting problem, 354, 361, 362  
Halting State, 356  
Handshaking theorem, 296  
Height of a tree, 297  
Heuristic programming, 464  
Holism, 456  
Hypothesis, 133  
Hypothetico-deductive method, 471  
Ideal perfect knowledge, 144

- Ideal vector, 135  
Identity function, 287  
Identity matrix, 294  
Imbalanced dataset, 192  
Impossible event, 304  
In-degree of a vertex, 296  
In-order tree traversal, 298  
Inaccuracy, 27, 37, 107, 189  
Inaccurate description, 37, 106  
Incident edge, 296  
Inclusion-exclusion principle, 292, 307  
Incompressible string, 412  
Independent and identically distributed, 342  
Independent events, 310  
Independent random variables, 324  
Individuals, 426  
Induction, 471  
Inferential conception, 463  
Infinite graph, 295  
Information distance, 406  
Initial State, 356, 365  
Initial vertex, 296  
Injective function, 287  
Input Symbol, 356, 365  
Integrative depth, 157, 158  
Interdisciplinary new topic, 172  
Interesting question, 167, 168  
Interestingness as a problem, 166  
Interestingness as a tool, 165  
Intersection of sets, 284  
Intradisciplinary new topic, 172  
Invariance Theorem, 398  
Inverse function, 287  
Irrelevant string, 63  
Isolated vertex, 296  
Joint Kolmogorov complexity, 402  
Joint probability distribution, 319  
Joint probability mass function, 320  
Joint representation, 74, 95  
k-ary tree, 297  
Kleene closure, 288  
Knowable entity, 52, 65, 455  
Knowledge frontier, 45, 150  
Known knowns, 148  
Known subset of an area, 86  
Known unknown, 155  
known unknown, 148  
Known unknowns, 167  
Kolmogorov Complexity, 397  
Kolmogorov complexity, 54  
Kolmogorov's axioms, 305, 310  
Kraft's inequality, 377  
Labeled graph, 297  
Lambda calculus, 354  
Language, 289  
Law of Large Numbers, 347  
Leaf vertex, 297  
Leibniz series, 36  
Length of a string, 288  
Likelihood function, 420  
Log-likelihood function, 420  
Logic, 454  
Logical coherence, 466  
Loss function, 419  
Lower class limit, 391  
Luminiferous ether, 69  
Machine State, 356, 365  
Marginal probability mass function, 322  
Markov's Inequality, 346  
Matrix, 293  
Matrix determinant, 295  
Matrix main diagonal, 294  
Matrix rank, 295  
Matrix transpose, 294  
Maturity, 164  
Maximal element, 286

- Maximum element, 286  
Maximum Likelihood Estimator, 420  
McMillan's Inequality, 379  
Median, 330  
MediaWiki, 226  
MediaWiki API, 227  
Metaphysics, 454  
Michelson-Morley experiment, 69  
Minimal element, 286  
Minimal oracle, 61  
Minimum decisions rule, 166  
Minimum element, 286  
Miscoding, 27, 33, 93  
MNIST, 189  
Model, 428  
Model-based reasoning, 466  
Modulo, 288  
Multinomial coefficient, 293  
Multiplication rule, 291, 310  
Multitape Turing machine, 357  
Mutually independent events, 311
- n-tuple, 286  
Nadir vector, 136  
Natural kinds, 457  
Neighborhood of a vertex, 296  
Nescience, 27, 41  
Nescience decision maker, 139  
Neural network, 80  
New research entity, 45  
Newton's second law, 38  
Nominal attribute, 427  
Non-knowable entity, 455  
Non-minimal oracle, 60  
Non-observable entities, 456  
Non-redundant description  
    core, 59  
    existence, 59  
Non-redundant representation, 59  
Non-singular code, 372  
Non-singular matrix, 294
- Non-terminal symbol, 290  
Normalized applicability, 162  
Normalized compression distance, 410  
Normalized information distance, 409  
Normalized maturity, 164  
Normalized weighted integrative depth  
    on a subset, 160
- Observable entities, 456  
Occam's razor principle, 78  
Ontology, 54, 454  
Open class interval, 391  
Oracle  
    minimal  
        characterization, 62  
        existence, 62  
    Oracle computable function, 367  
    Oracle reduction, 60  
    Oracle Turing machine, 30, 36, 55,  
        354, 365  
    Order of a function, 288  
    Ordered pair, 285  
    Ordinal attribute, 427  
    Orthogonal matrix, 294  
    Orthonormal vector, 295  
    Out-degree of a vertex, 296  
    Outcome, 304  
    Overfitting, 190
- P vs. NP, 111  
Padding lemma, 360  
Pair-specific nescience reduction, 167  
Pairwise independent events, 311  
Paradigm shift, 465  
Parameter of a probability distribution, 418  
Parameter space, 418  
Parametric random variable, 418  
Parent of a vertex, 297  
Pareto frontier, 134, 145, 165  
Pareto optimal, 134

- Pareto optimality, 42  
Pareto perfect knowledge, 145  
Pareto point, 115, 129  
Parsimony, 467  
Partial computable function, 363  
Partial function, 287  
Partial order, 286  
Partially ordered set, 241, 286  
Partition of a set, 285  
Path, 296  
Pendant vertex, 296  
Perfect conditional description, 84, 85  
Perfect description, 78  
Perfect description of an area, 88  
Perfect knowledge, 43  
Perfect representation, 63  
Permutations, 292  
Philosophy of science, 451  
Phlogiston, 70  
Pigeonhole principle, 292  
Pleonastic description, 81  
Polywater, 36  
Population, 426  
poset, 286  
Possibility of misrepresentation, 462  
Post Correspondence Problem, 362  
Post-order tree traversal, 298  
Posterior distribution, 423  
Posterior probability, 312  
Potential new topic, 171  
Power set, 285  
Pragmatic theories, 470  
Pre-order tree traversal, 298  
Prefix free set, 289  
Prefix string, 289  
Prefix-free code, 375  
Prefix-free universal Turing machine, 397  
Principle of indifference, 302  
Prior distribution, 423  
Prior probability, 312  
Probability, 305  
Probability mass function, 308, 316  
Probability space, 305  
Problem of ontology, 56, 461  
Problem of style, 57, 461  
Production rule, 290  
Proper subset, 284  
Ptolemy, 66  
Qualitative attribute, 427  
Quantitative attribute, 427  
Question, 167  
Quotient set, 286  
Random sample, 342  
Random string, 413  
Random variable, 314  
Range, 287  
Realism, 457  
Recursive function, 354, 363  
Recursively enumerable grammar, 290  
Reduced representation, 73  
Reductionism, 456  
Redundacy, 123  
Redundancy, 229  
Redundancy closure, 60  
Redundant string, 63  
Reflexive relation, 286  
Regression problems, 428  
Regular grammar, 290  
Relative frequency, 392  
Relative variation of inaccuracy, 112  
Relative variation of surfeit, 127  
Representation, 26, 28, 66, 67  
Representation function, 55  
Representation of an area, 87  
Representation oracle, 55  
Representation-as, 463  
Representational demarcation, 56, 460

- Requirement of directionality, 57, 461  
Research area, 86  
Reverse string, 288  
Root of a tree, 297  
Row matrix, 293  
Russell's paradox, 28, 54
- Sample covariance, 345  
Sample mean, 343  
Sample size, 342  
Sample space, 304  
Sample standard deviation, 345  
Sample variance, 344  
Sardinas-Patterson, 374  
Science, 452  
Science decision maker, 139  
Scientific method, 33, 133, 452  
Scientific representation, 54, 460  
Self delimited string, 289  
Self-correction, 453  
Sequence of symbols, 288  
Set, 284  
Set difference, 285  
Set formation notation, 284  
Set of conditional descriptions, 84  
Set of descriptions, 77  
Set of integers, 284  
Set of natural numbers, 284  
Set of positive integers, 284  
Set of positive reals, 284  
Set of rationals, 284  
Set of real numbers, 284  
Shakespeare, 57  
Shortlex ordering, 289  
Sibling to a vertex, 297  
Sicence problem, 133  
Similarity conception, 462  
Simple cycle, 296  
Simple path, 296  
Singular value decomposition, 295
- Source alphabet, 372  
Source symbol, 372  
Square matrix, 294  
St. Petersburg Paradox, 352  
Standard deviation, 332  
Standard of accuracy, 57, 461  
Standardized discrete random variable, 339  
Start symbol, 290  
State diagram, 357  
Statistic, 418  
Statistical inference, 416  
Statistical model, 417  
Statistical relevance, 469  
Statistical-Relevance model, 469  
Stipulative fiat, 462  
Stirling's formula, 292  
Stock market, 192  
String, 288  
String concatenation, 288  
String quotient, 289  
Strongly connected graph, 296  
Structuralist conception, 463  
Subgraph, 297  
Subjective interpretation of probability, 303  
Submatrix, 294  
Subset, 284  
Substring, 289  
Subtree, 297  
Support of a distribution, 316  
Surfeit, 27, 40, 120, 229  
Surjective function, 287  
Surrogate reasoning, 57, 99, 461  
Symbols, 288  
Symmetric matrix, 294  
Symmetric relation, 286
- Tape Symbol, 356, 365  
Target variable, 427  
Targetless models, 462

- Targetless representation, 99  
Targetless representations, 67  
Terminal symbol, 290  
Terminal vertex, 296  
Testability, 453  
Theoretical framework, 453  
Theory of relativity, 69  
Topic, 58  
Total order, 286  
Total relation, 286  
Totally ordered set, 286  
Trade-off vector, 138  
Training set, 428  
Transition Function, 356, 365  
Transitive relation, 286  
Tree, 297  
Tree traversal, 298  
Turing machine, 30, 32, 353, 355  
Turing's thesis, 359
- Uncountable set, 287  
Unificationist account, 470  
Unified trade-off magnitude, 138  
Uniform distribution, 337  
Union of sets, 284  
Uniquely decodable code, 373  
Universal set, 27  
Universal Turing machine, 354, 361  
Universe of a set, 284  
Unknowable entity, 52, 65  
Unknown unknown, 155  
unknown unknown, 149  
Unknown unknowns, 44, 167  
Upper bound of a function, 288  
Upper class limit, 391  
Utility function, 139  
utility function, 42
- Value function, 139  
Variance, 330  
Variation of inaccuracy, 112, 113  
Variation of surfeit, 126
- Venn diagram, 285  
Vertices of a graph, 295  
Wallis product, 35  
Weakly dominates, 135  
Weakly Pareto frontier, 135  
Weakly Pareto optimal, 135  
Weight of an edge, 297  
Weighted applicability, 162  
Weighted integrative depth, 158  
    restricted to a subset, 159  
Weighted sum decision rule, 166  
Wikipedia, 225  
Wikitextparser, 226  
Words of a language, 289
- Yield, 358
- Zermelo-Fraenkel set of axioms, 53