Probabilistic (Graphical) Models

and inference

Oliver Obst · Autumn 2025

WESTERN SYDNEY
UNIVERSITY

Probabilistic (Graphical) Models and Inference

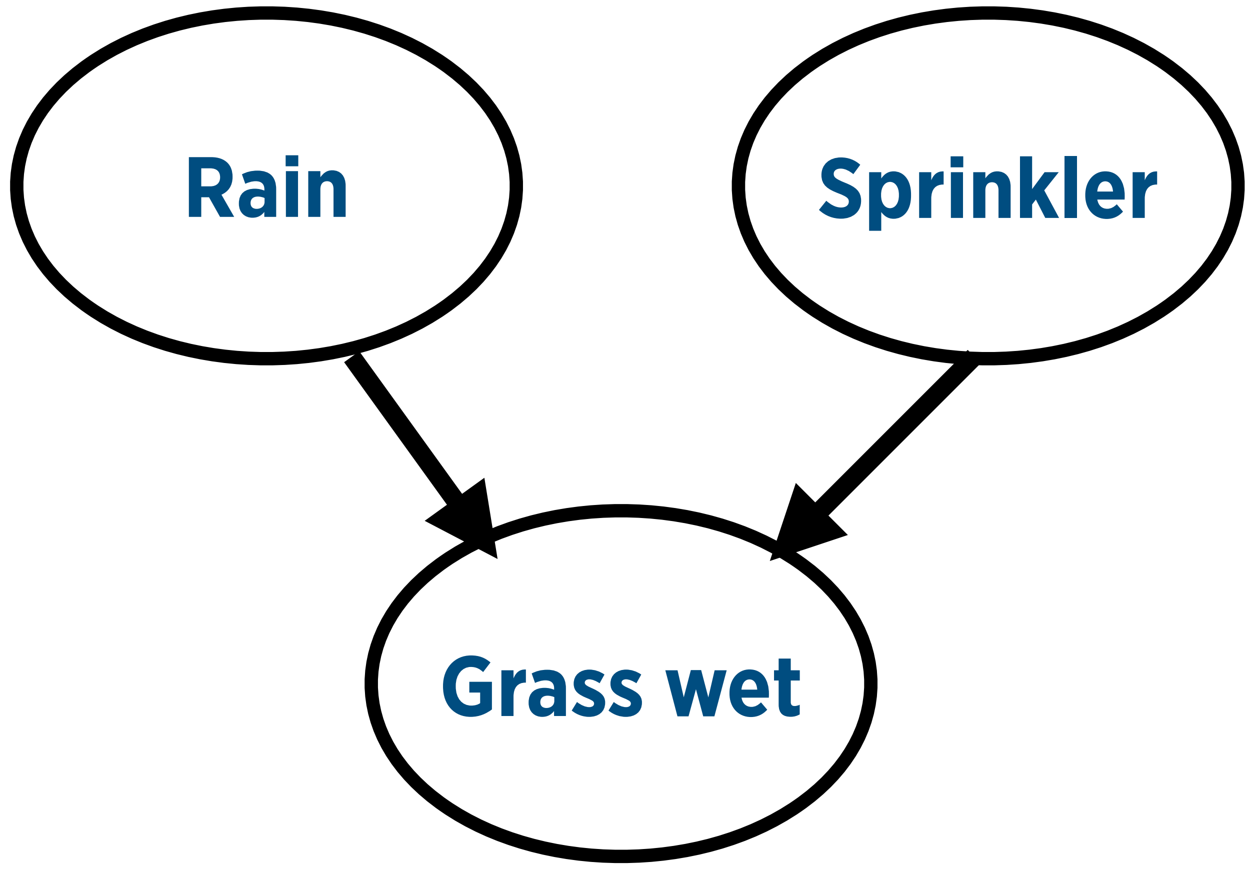
(PGM: Probabilistic Graphical Models: Principles and Techniques by Daphne Koller and Nir Friedman. MIT Press)
(PMLI: Probabilistic Machine Learning: An introduction by Kevin Murphy. MIT Press)

Week	Lecture	Required reading	Assessment
1 Friday, 7 March 2025	Introduction, Probability Theory	PGM Chapter 2, PMLI Chapter 6.1	
2 Friday, 14 March 2025	Directed and undirected networks introduction		Math7017 — Quiz 1
3 Friday, 21 March 2025	Variable elimination		Math3011 — Quiz 1
4 Friday, 28 March 2025	Belief propagation		Math7017 — Quiz 2
5 Friday, 4 April 2025	Message passing & Graph neural networks		Math3011 — Quiz 2 (Census date)
6 Friday, 11 April 2025	Sampling		Math7017 — Quiz 3
7 Friday, 18 April 2025	(no lecture — public holiday)		Math3011 — Quiz 3
8 Friday, 25 April 2025	Mid-term break		
9 Friday, 2 May 2025	Variational inference		Math7017 — Intra-session exam
10 Friday, 9 May 2025	Autoregressive models		Math3011 — Intra-session exam Math7017 — Quiz 4
11 Friday, 16 May 2025	Variational Auto-Encoders		Math3011 — Quiz 4
12 Friday, 23 May 2025	GANs		Quiz 5
13 Friday, 30 May 2025	Energy-based models		
14 Friday, 6 June 2025	Evaluating generative models		Quiz 6
Monday, 17 June 2024			Project due

Some material adapted from
- Sargur Srihari
- David Sontag
- Daphne Koller and Nir Friedman

What are Probabilistic Graphical Models?

PGMs **represent** complex probability distributions using a graph structure.



Purpose:

- Handle **uncertainty** directly in data and models
- **Predict** events and assess how confident we are in those predictions
- Leverage **conditional independence** to simplify computations

Why does this matter:

- Real-world data often involve **many** interacting factors.
- PGMs offer a structured way to make sense of these factors, especially when data is noisy or incomplete.

Today: We review basic probability rules, then see how PGMs put those rules to work in a graph.

Why Probabilistic Graphical Models vs. Deep Learning?

Deep Learning: Strengths

- Powerful for image/speech tasks
- Learns directly from raw data
- Outstanding performance given large labeled datasets

Deep Learning: Challenges

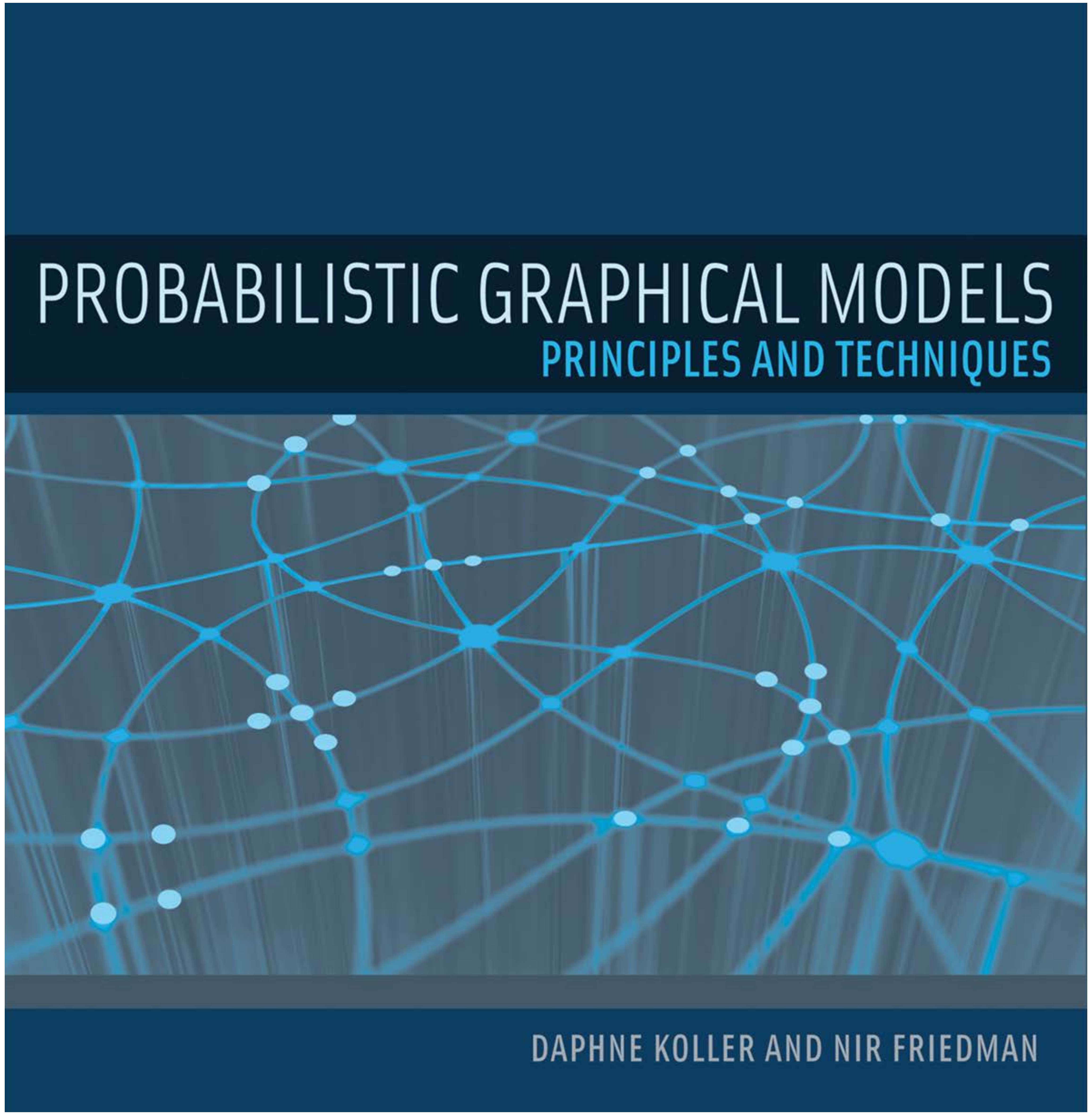
- Overconfident with no built-in measure of uncertainty
- Limited interpretability (“black-box” nature)
- Can amplify data biases
- Often requires massive labeled datasets

PGMs: Advantages

- Explicit uncertainty modelling (probability distributions)
- More transparent structure (conditional dependencies)
- Can incorporate domain knowledge & handle partial data
- Potentially data-efficient thanks to prior information

Why now?

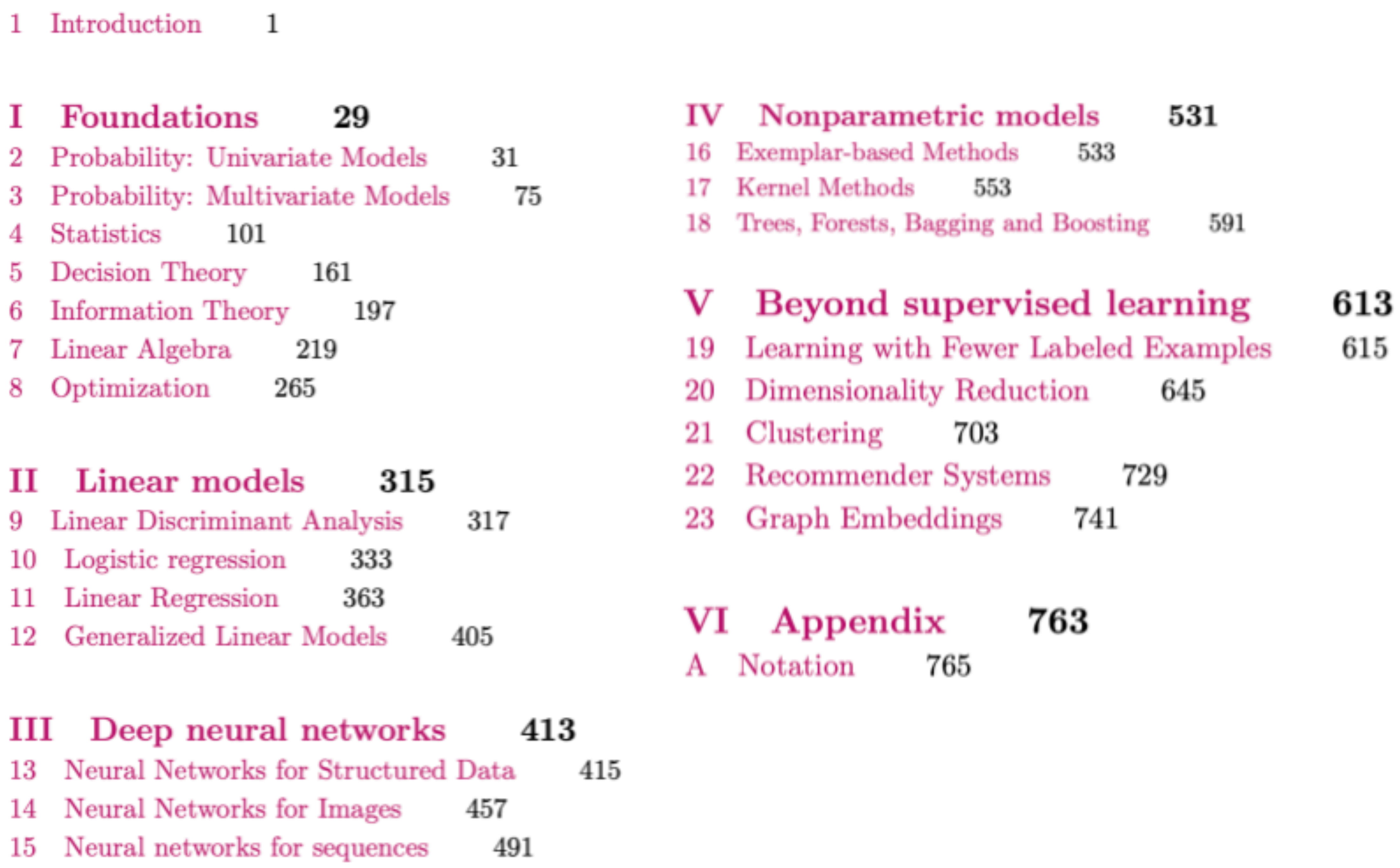
- Industry demands **explainable AI** and **robust** predictions
- PGMs were overshadowed by DL in the past but are making a **comeback** to address interpretability and calibration issues
- Hybrid methods combine best of both worlds

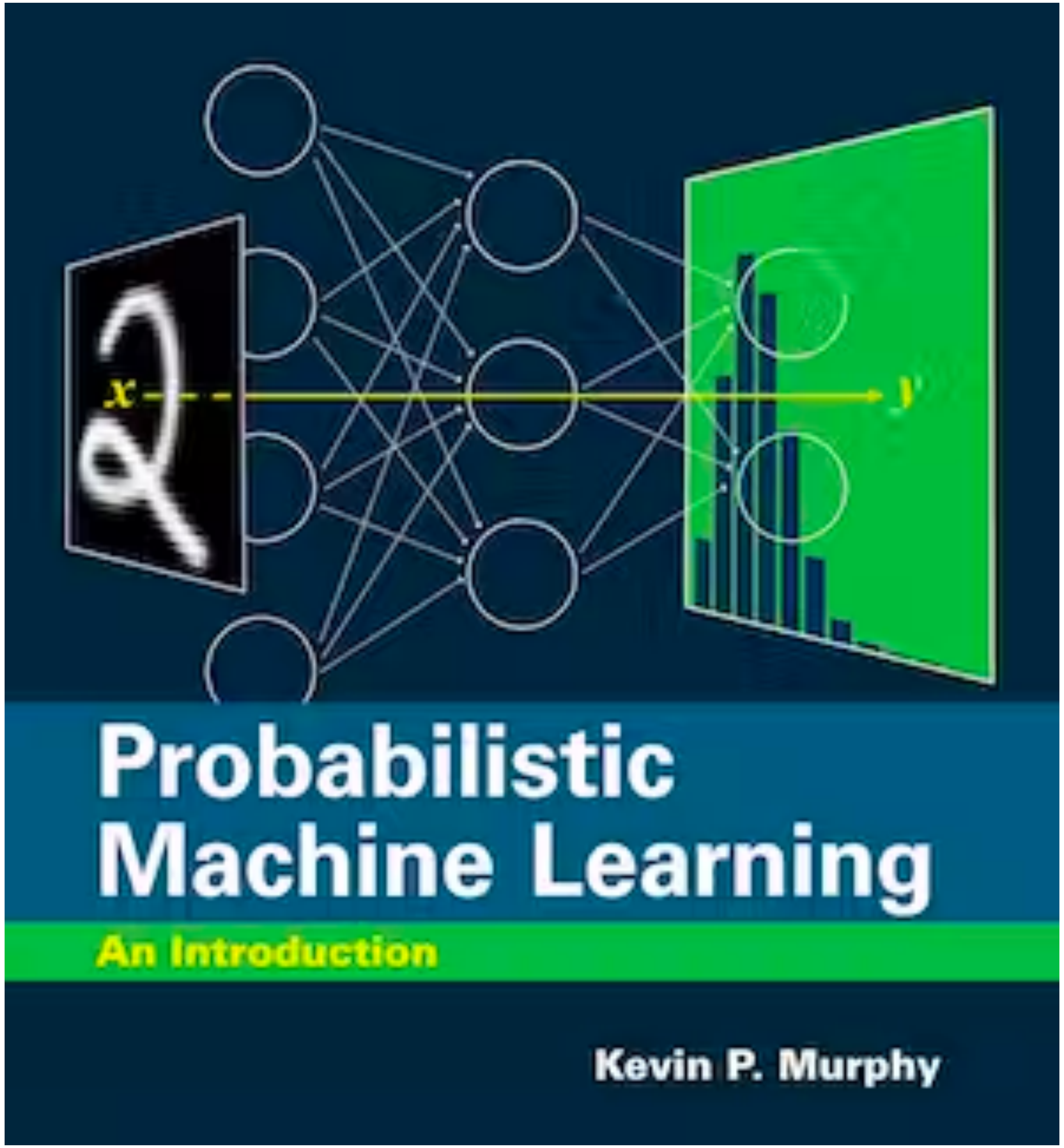

PROBABILISTIC GRAPHICAL MODELS
PRINCIPLES AND TECHNIQUES
DAPHNE KOLLER AND NIR FRIEDMAN

MIT Press, 2009

Another (newer) book

- https://probml.github.io/pml-book/book1.html





Readings

Probabilistic Graphical Models: Principles and Techniques by D. Koller and N. Friedman.
If we refer to "Chapter X", it means the chapter from this book.

Probabilistic Machine Learning: An Introduction by Kevin P. Murphy
https://probml.github.io/pml-book/book1.html

Intro to probability:

Introduction to Probability, 2nd ed (Joseph K. Blitzstein, Jessica Hwang)
Free course and book: https://projects.iq.harvard.edu/stat110/home

Information theory, inference, and learning algorithms. MacKay, D. J. C. (2003). Cambridge University Press.
http://www.inference.org.uk/itila/book.html

Maths refresher:

Mathematics for Machine Learning: https://mml-book.github.io/book/mml-book.pdf.

RTFLG (read the fantastic learning guide)

Videos

Coursera Probabilistic Graphical Models Specialisation

The videos are optional / additional / not a replacement for our lectures.

- https://www.coursera.org/specializations/probabilistic-graphical-models

Programming in Python

- Rich ecosystem for scientific computing (NumPy, pandas, PyTorch, etc.).
- Easy integration with data science workflows.
- Jupyter notebooks for quick, interactive exploration (not usually for production).

Setting Up

- Install Anaconda or use `pip install numpy matplotlib pandas`.
- Choose an environment:
  - **Jupyter** Lab/Notebook for interactive coding
  - An IDE (Spyder, VSCode, PyCharm) for bigger projects

A Note on Learning Programming

We'll provide examples, tips, and pointers, but mastering programming is like learning to swim:

- You can watch demos, but you have to dive in and **practice** to improve.

Example first goals

For this week 1, work on the following things. These should be easy.

- How to install python on your machine?

- Python can make use of extra libraries that are not installed by default. Figure out if NumPy, matplotlib, pandas, torch are installed with python. Figure out how to install them if they are not.

- Write a Python program that
  - creates a random 3 x 3 matrix W, a random 3 x 1 vector x, and then prints the 3 x 1 vector $y = Wx$.
  - solves the practice quiz on vUWS (quiz 0 — unmarked).

- Try this for yourself first. If you get stuck, ask others in this class. It’s a good idea to learn programming together with others.

Other steps till next time

- Revise some of the data science and programming introduction (previous units).
- Revise probabilities (it helps if you took advanced statistical methods)

- This is an advanced subject.
  We require you to know material from earlier units.
- You will have spend time practicing your skills — approx. 10h/week.

Unit *Philosophy*

- “Understanding” assumes some knowledge that you may have from school or earlier classes, mostly linear algebra, calculus, stats.
- “Applying” means that you will have to program (usually short pieces of code). I will use Python for examples and questions, but (unless specifically requested to use python) you can also use R or Matlab if you prefer.
- We will explain and revise some of the programming, but this is not an introductory Maths or Programming class. If you don’t understand or don’t know how to do things at first, persist and try to figure things out.
- (As usual in my classes) we aim to not be overly prescriptive: there are often multiple ways to solve problems. Treat it like a job – solve the problem. Do not just copy/paste!

Assessment Items

	Weight
In-tutorial quizzes (6 overall)	30%
Applied project	40%
Practical exam	30%

To pass, you will need to attempt every item, and achieve at least 50% overall.

This unit is 10 credit points, and will likely be perceived a difficult unit by most. This unit will require 10 hours of study per week. This time includes the time spent within classes during lectures, tutorials / practicals.

I recommend to not take more than 4 units over the semester, and less if you are working full time.

The usual questions

- Will lecture X / topic Y / something we do in a practical be part of the exam?
  Yes, possibly.

- Will the exam be open book?
  You can use your own notes and code, books, websites, no chatGPT, don’t call your mum.

- Will we have to program, can we use our own computer?
  Yes, and most MacOs/Windows/Linux computers should work.

- Can we have a practice exam?
  There will be no “practice” exam to show you what the exam could or will look like. The quizzes will be a preparation for the exam.

What should I do with my time?

(not a usual question, but glad that you asked)

- Read the relevant chapters of the book, each week. Ideally before the lecture, but after the lecture is also good.

- You will get much better at this topic with practice, so .. please practice.

- There are heaps of resources (tutorials, data sets, competitions) online

- Find practice problems for yourself to solve

- I will suggest online material from time to time

How can we gain **global insight** based on **local observations?**

(David Sontag)

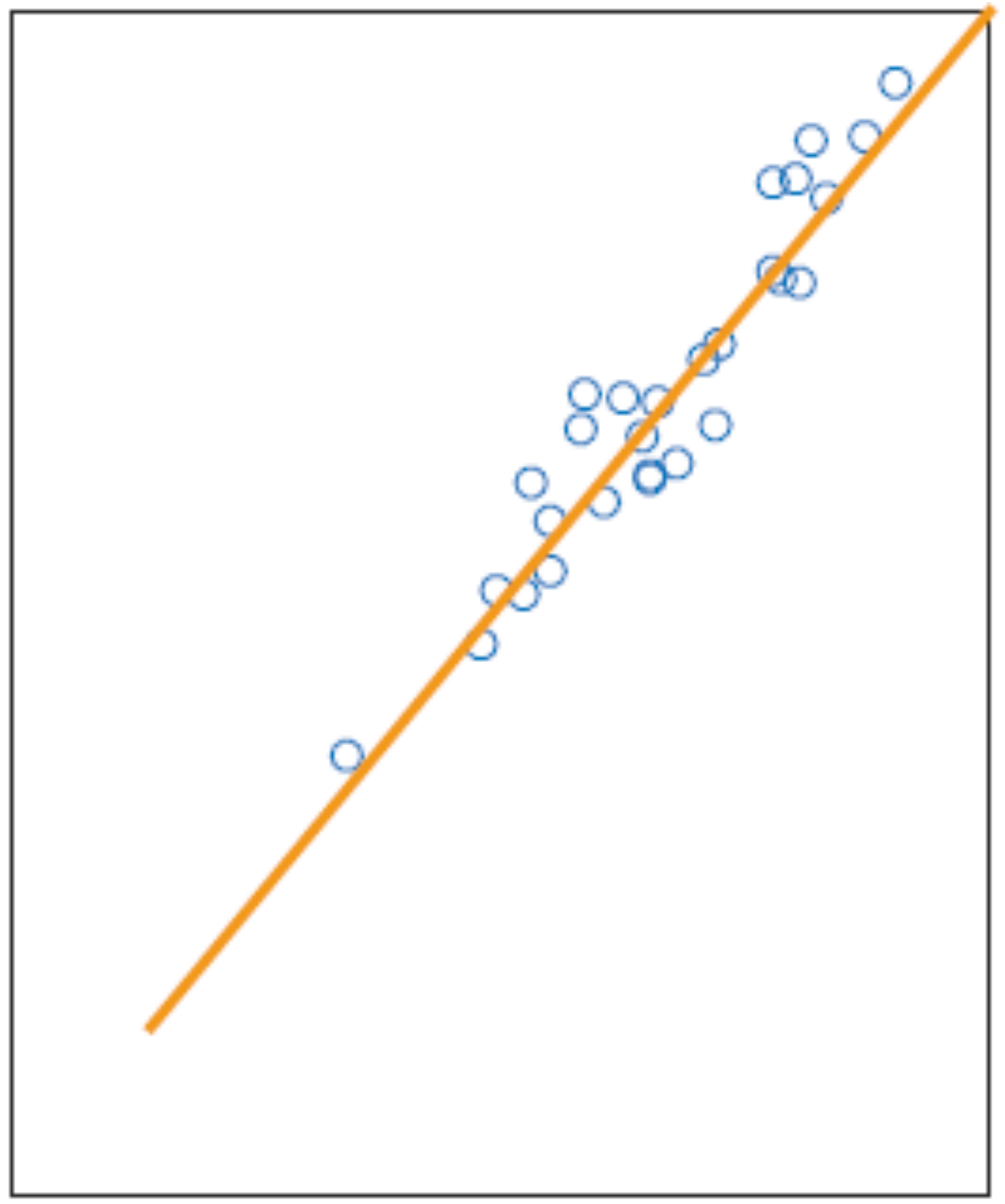
Key Idea

- **Represent** the world as a collection of random variables $X_1, \ldots, X_n$ with joint distribution $p(X_1, \ldots, X_n)$
- **Learn** the distribution from data
- Perform "**inference**" - i.e., compute conditional distributions $p(X_i \mid X_1 = x_1, \ldots, X_m = x_m)$

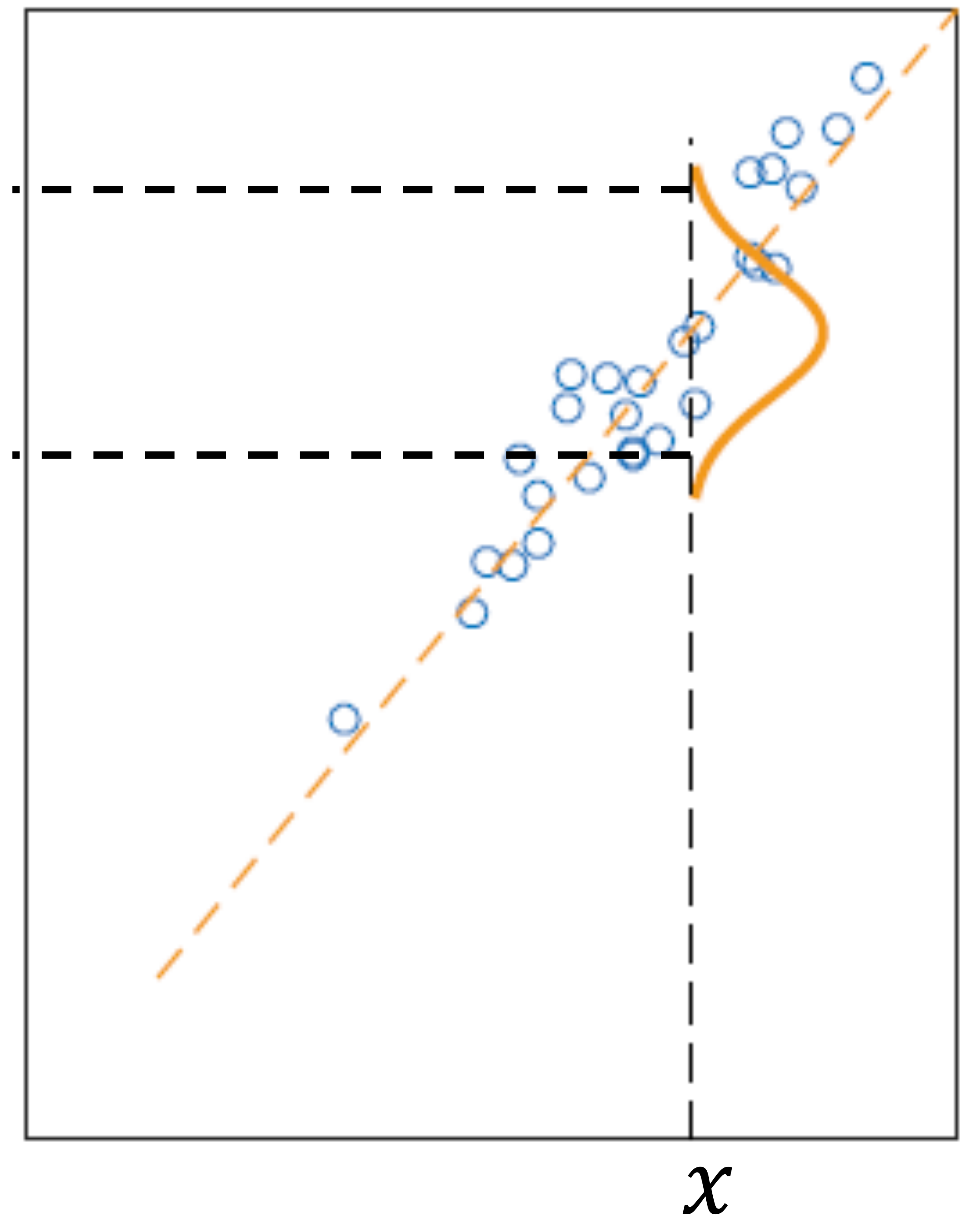
Reasoning under uncertainty

- We are continuously making predictions under uncertainty
- But uncertainty does not play a big role in “classical” AI and many of the machine learning approaches
- Probabilistic approaches enable us to do things that are not possible otherwise
- Different kinds of uncertainty:
  partial knowledge / noise / modelling limitations / inherent randomness

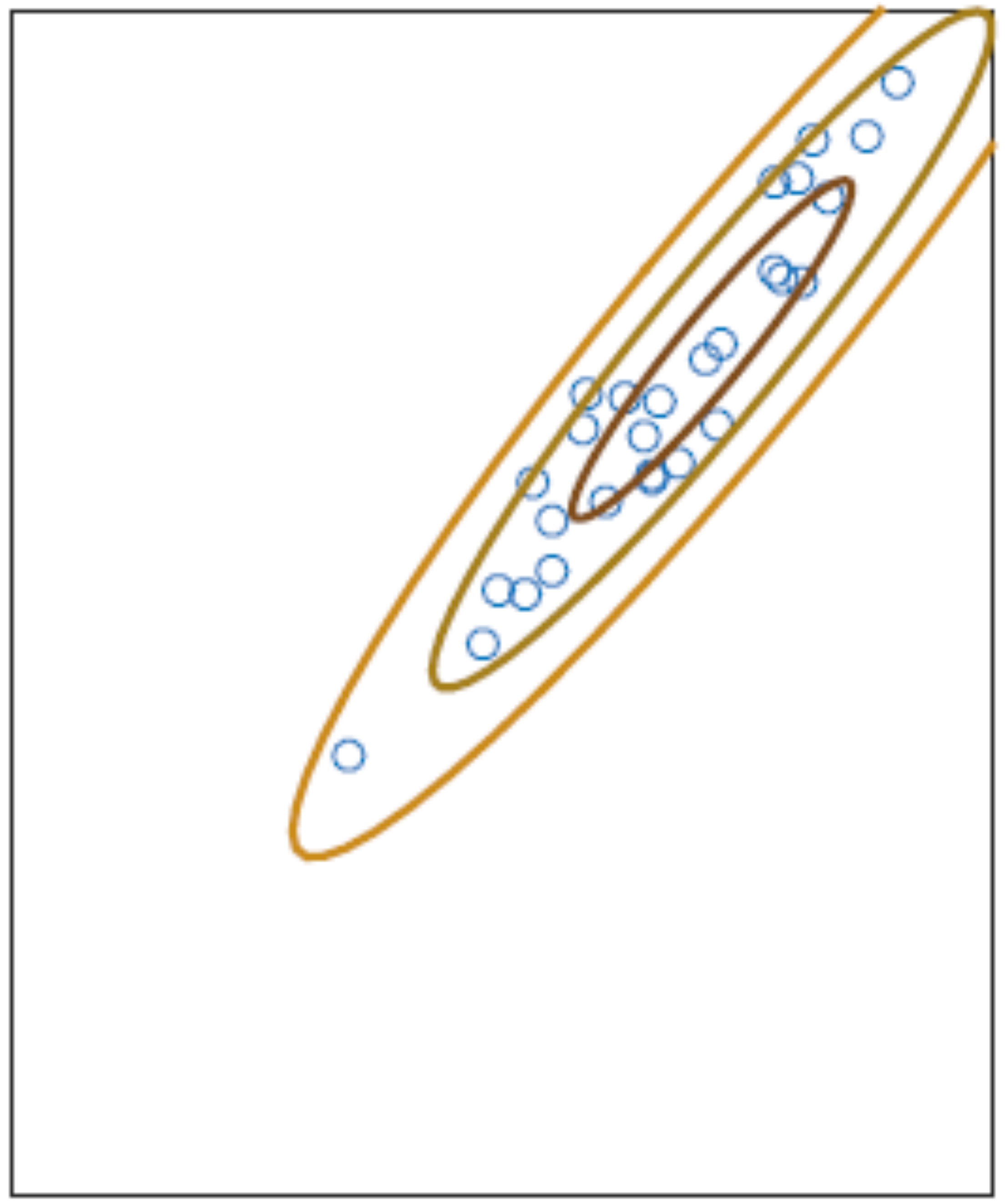
Types of Models



$\hat{y} = f(x)$



$P(y|x)$



$P(x, y)$

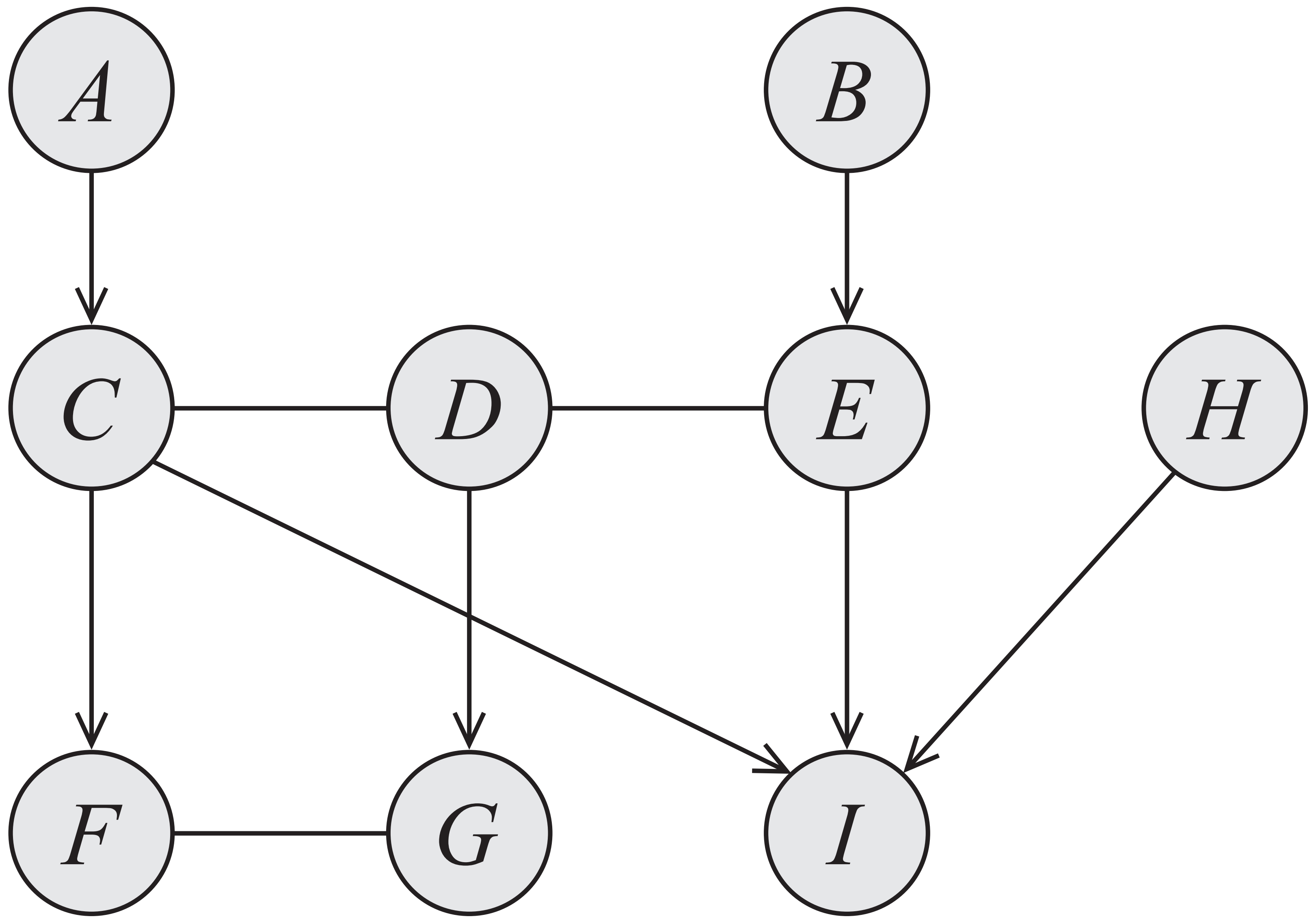
Trevor Cohn, Uni Melbourne

Graphical models

- Graph – as in: a set of nodes connected with edges / vertices
- To organise and represent knowledge and relations
- If we have random variables $X_1, \ldots, X_n$ with joint distribution $p(X_1, \ldots, X_n)$,
  and every random variable could only take 2 values,
  a complete table would have 2^n rows.

Graphs

- A graph is a data structure G, consisting of a set of nodes / vertices, and a set of edges
- Graphs can be directed or undirected



Key challenges

1. **Represent** the world as a collection of random variables $X_1, \ldots, X_n$ with joint distribution $p(X_1, \ldots, X_n)$
   - How can we *compactly describe* this joint distribution?
   - Directed graphical models (Bayesian Networks)
   - Undirected graphical models (Markov random fields, factor graphs)

2. **Learn** the distribution from data
   - Maximum likelihood estimation, other estimation methods
   - How much data do we need?
   - How much computation does it take?

3. Perform “**inference**” - i.e., compute conditional distributions $p(X_i \,|\, X_1 = x_1, \ldots, X_m = x_m)$

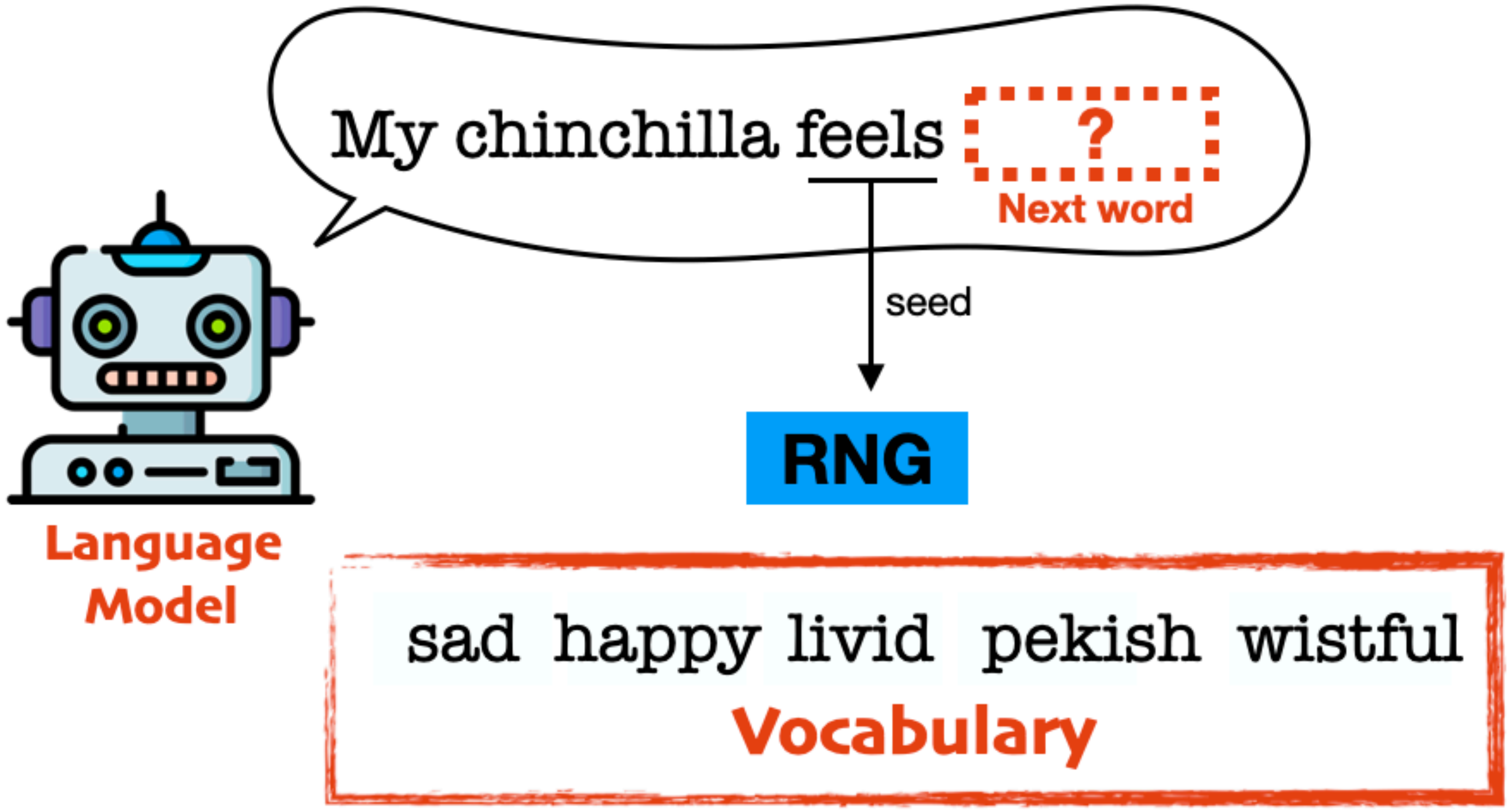
Application of probabilities: Detecting generated texts

A watermark for language models

https://arxiv.org/abs/2301.10226 John Kirchenbauer et al.

It is possible for LLMs to watermark generated text

- without degrading text quality
- without re-training the language model
- with an open-source approach, without publishing the language model



(Image: Tom Goldstein @tomgoldsteincs)

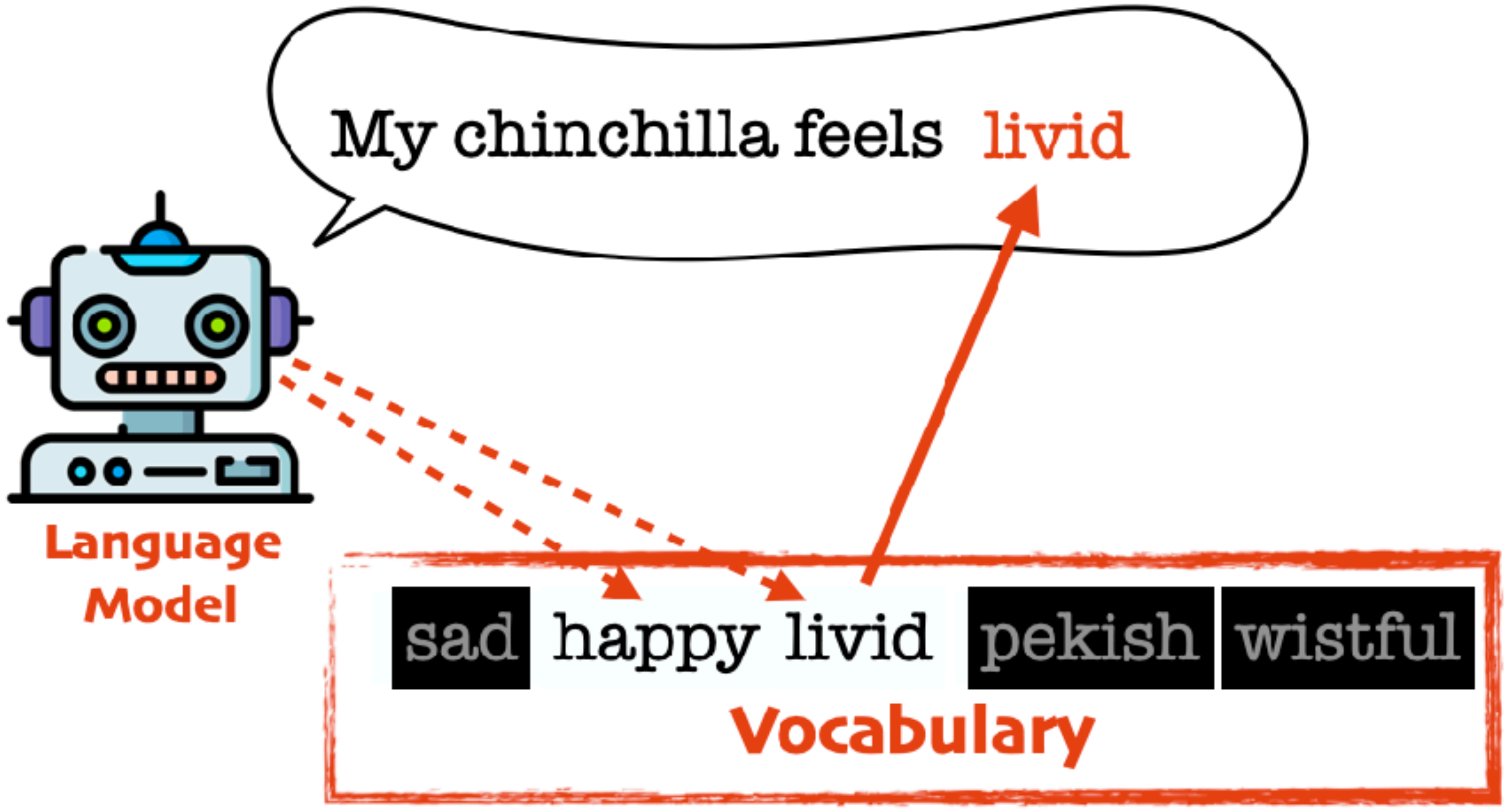
The resulting text can be detected as “generated”, with extremely high probability.

Here’s the idea (simplified):

For every word (token) to be generated:

- seed the RNG with previous word
- create a new whitelist, a random 50% of the entire vocabulary
- we only choose words from whitelist

Generated text will have only* whitelisted words. Probability that you pick a word from whitelist is 50%.



(Image: Tom Goldstein @tomgoldsteincs)

For N words, this probability is 0.5^N.

A tweet of 25 words, with only words from whitelists, is 99.99999997% generated.

A watermark for language models

Actual approach is more sophisticated:

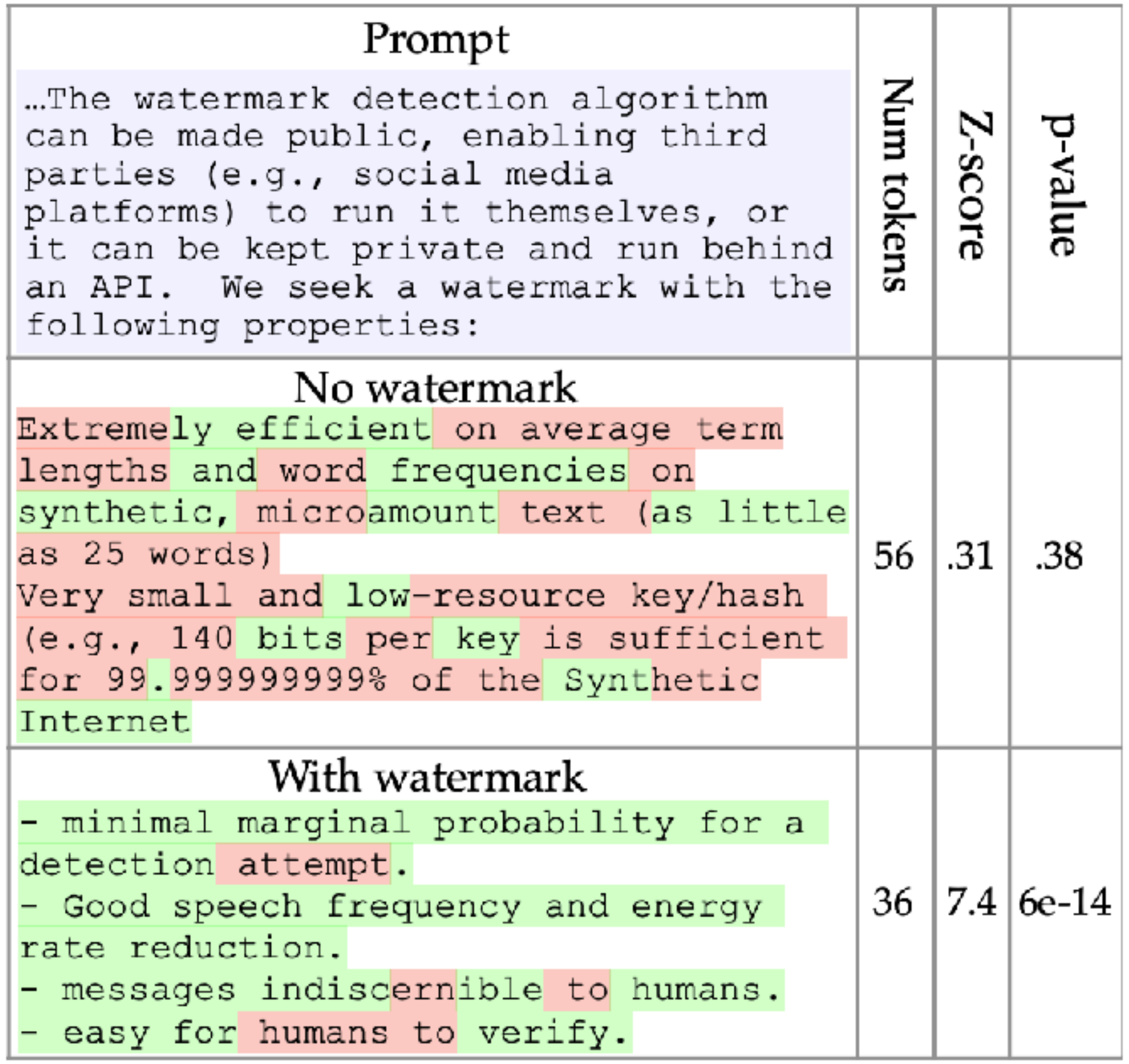
- No "strict" black-/whitelist, but avoid blacklisted words probabilistically.

- Can better deal with "low entropy" parts of the text (e.g., "Barack"⇝"Obama", almost always).

- Can then use smaller whitelist (e.g., 25%)

False positives (human text flagged as fake) are improbable.

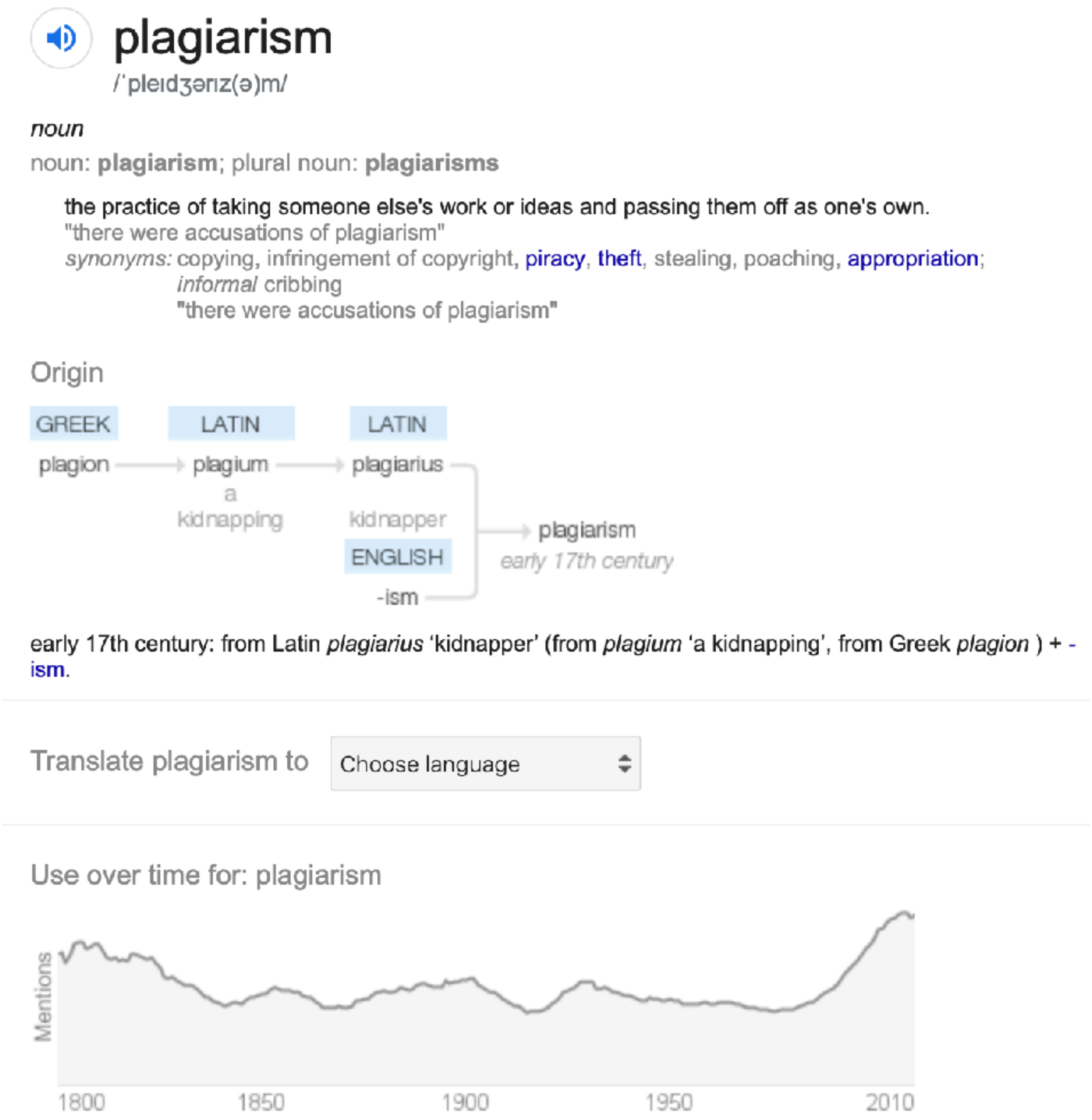
"Synonym attacks" need to replace impractically large parts of the generated text.



Prompt	Num tokens	Z-score	p-value
...The watermark detection algorithm can be made public, enabling third parties (e.g., social media platforms) to run it themselves, or it can be kept private and run behind an API. We seek a watermark with the following properties:			
No watermark			
Extremely efficient on average term lengths and word frequencies on synthetic, microamount text (as little as 25 words) Very small and low-resource key/hash (e.g., 140 bits per key is sufficient for 99.999999999% of the Synthetic Internet	56	.31	.38
With watermark			
- minimal marginal probability for a detection attempt. - Good speech frequency and energy rate reduction. - messages indiscernible to humans. - easy for humans to verify.	36	7.4	6e-14

Figure 1. Outputs of a language model, both with and without the application of a watermark. The watermarked text, if written by a human, is expected to contain 9 "whitelisted" tokens, yet it contains 28. The probability of this happening by random chance is $\approx 6 \times 10^{-14}$, leaving us *extremely* certain that this text is machine generated. Whitelist words are green, blacklist words are red. The model is OPT-6.7B using multinomial sampling. Watermark parameters are $\gamma, \delta = (0.25, 2)$. The prompt is the whole blue paragraph marked in blue below.

https://arxiv.org/abs/2301.10226 John Kirchenbauer et al.



plagiarism

/ˈpleɪdʒərɪz(ə)m/

noun

noun: **plagiarism**; plural noun: **plagiarisms**

the practice of taking someone else's work or ideas and passing them off as one's own.

"there were accusations of plagiarism"

synonyms: copying, infringement of copyright, piracy, theft, stealing, poaching, appropriation; *informal* cribbing

"there were accusations of plagiarism"

Origin

GREEK plagion → LATIN plagium a kidnapping → LATIN plagiarius kidnapper + ENGLISH -ism → plagiarism *early 17th century*

early 17th century: from Latin *plagiarius* 'kidnapper' (from *plagium* 'a kidnapping', from Greek *plagion*) + -ism.

Translate plagiarism to Choose language

Use over time for: plagiarism

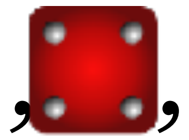
Academic misconduct

Basics (recap)

Probability: outcomes

(PGM chapter 2)

- An outcome space specifies the possible outcomes that we would like to reason about, for example:

  - $\Omega =$ { , } coin toss
  - $\Omega =$ { , , ,  , , } die toss

- We specify a probability $p(\omega)$ for each outcome ω such that

  - $$p(\omega) \geq 0, \qquad \sum_{\omega \in \Omega} p(\omega) = 1$$

For example: p( ) = 0.6, p( ) = 0.4

Probability: events

- An event is a subset of the outcome space, for example:
  - E = {⚁, ⚃, ⚅} even die tosses
  - O = {⚀, ⚂, ⚄} odd die tosses

- The probability of an event is given by the sum of the probabilities of the (elementary) outcomes it contains,

$$P(E) = \sum_{\omega \in E} p(\omega)$$

For example, $P(E) = p($⚁$) + p($⚃$) + p($⚅$) = \frac{1}{2}$, for fair dice.

Discrete random variables

Often, each outcome corresponds to a setting of various *attributes* (e.g., “age”, “gender”, “hasPneumonia”, “hasDiabetes”)

A random variable X is a mapping $X : \Omega \to D$

- D is some set (for example: the integers)
- It induces a partition of all outcomes Ω

For some $x \in D$, we say $p(X = x) = p(\{\omega \in \Omega : X(\omega) = x\})$
“probability that variable X assumes state x”

- Val(X): set D of all values assumed by X (“values” or “states” of X)

Probability measures and spaces

- A probability measure maps from a set of events to a real number (probability)

  $P : E \mapsto [0,1]$, for example $P(\{1\}) = 1/6,\ P(\{1,2\}) = 1/3, \ldots$

- A probability space consists of a set of outcomes Ω, a set of events E, and a probability measure P that maps from events to probabilities, $p : E \mapsto [0,1]$.

- Distributions over a discrete space can be described by the probabilities of the *atomic* events in that space.

- Example:{⚁} is an atomic event, for a single die, {⚁,⚃,⚅} is not.

σ - algebra

(sigma-algebra)

Let E be a set of elementary events. Consider a power set of E, written as 2^E.

A subset $\mathscr{F} \subseteq 2^E$ is called a σ-algebra if it satisfies the following properties:

1. $E \in \mathscr{F}$, i.e., the algebra contains all elementary events.
2. $\mathscr{F}$ is closed under complementation: If $A \in \mathscr{F}$, then so is its complement,$A^c \in \mathscr{F}$.
3. $\mathscr{F}$ is closed under countable unions: If $A_1, A_2, \ldots, \in \mathscr{F}$, then $A_1 \cup A_2 \cup \ldots \in \mathscr{F}$.

Probability measure

For the set E with the σ-algebra $\mathscr{F}$, a non-negative real function $P : \mathscr{F} \mapsto \mathbb{R}_0^+$ is called a **measure** if it satisfies the following properties:

1. $P(\emptyset) = 0$
2. If $A_1, A_2, \ldots \in \mathscr{F}$ are pairwise disjoint, then $P(\cup_{i=1}^{\infty} A_i) = \displaystyle\sum_{i=1}^{\infty} P(A_i)$.

A measure P is called a **probability measure** if

3. $P(E) = 1$.

We call $(E, \mathscr{F}, P)$ a probability space.

Sum rule

Foundations of the theory of probability, A. N. Kolmogorov, 1933

From $A + \neg A = E$ we can see that

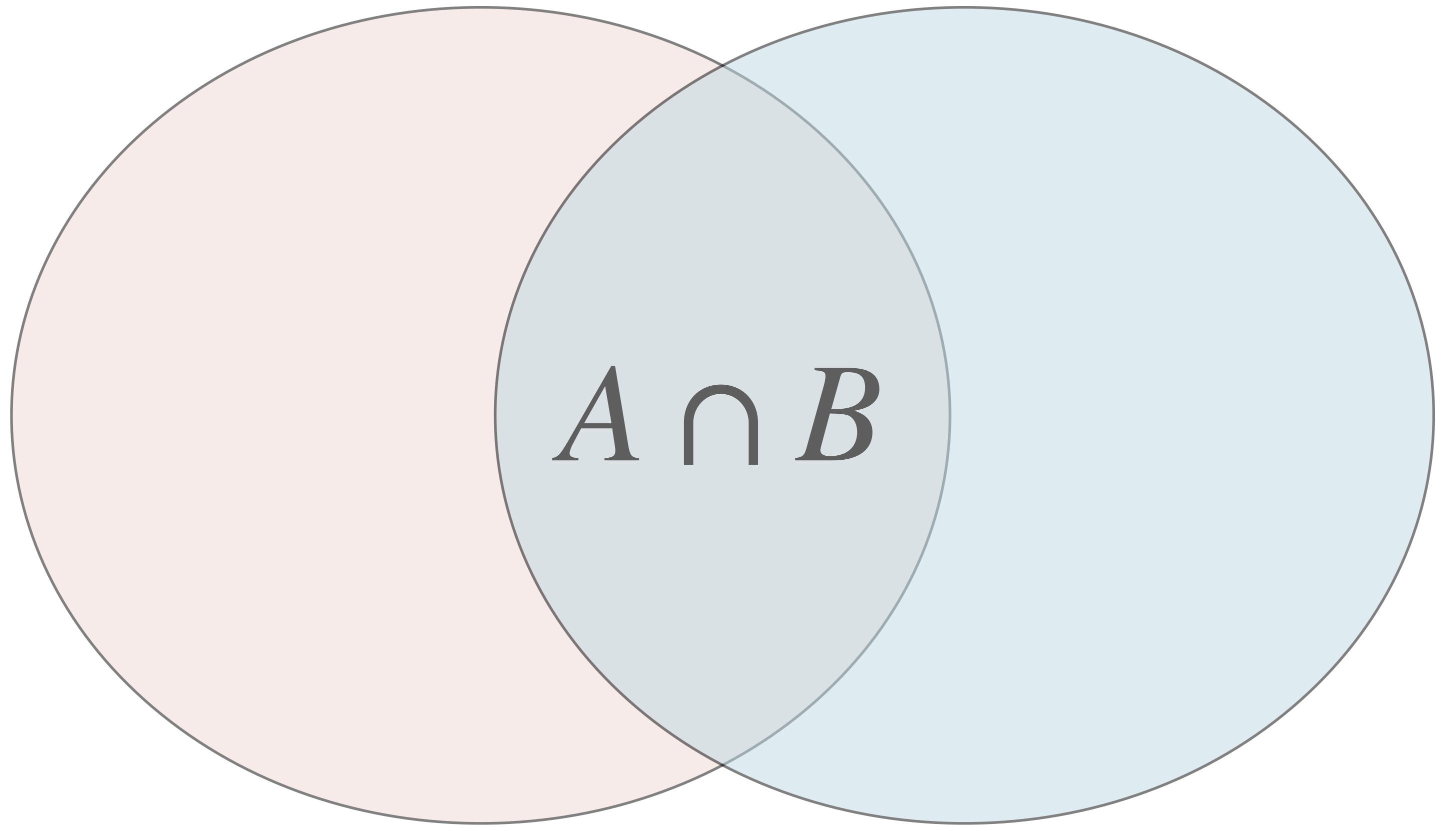
$$P(A) + P(\neg A) = P(E) = 1, \text{ so } \; P(A) = 1 - P(\neg A).$$

From $A = A \cap (B + \neg B)$ and using the notation $P(A, B) = P(A \cap B)$ for the joint probability of A and B, we get the **sum rule**:

$$P(A) = P(A, B) + P(A, \neg B).$$

Conditional Probability

Foundations of the theory of probability, A. N. Kolmogorov, 1933



If $P(A) > 0$, the quotient

$$P(B\,|\,A) = \frac{P(A,B)}{P(A)}$$

is called the **conditional probability** of B given A.

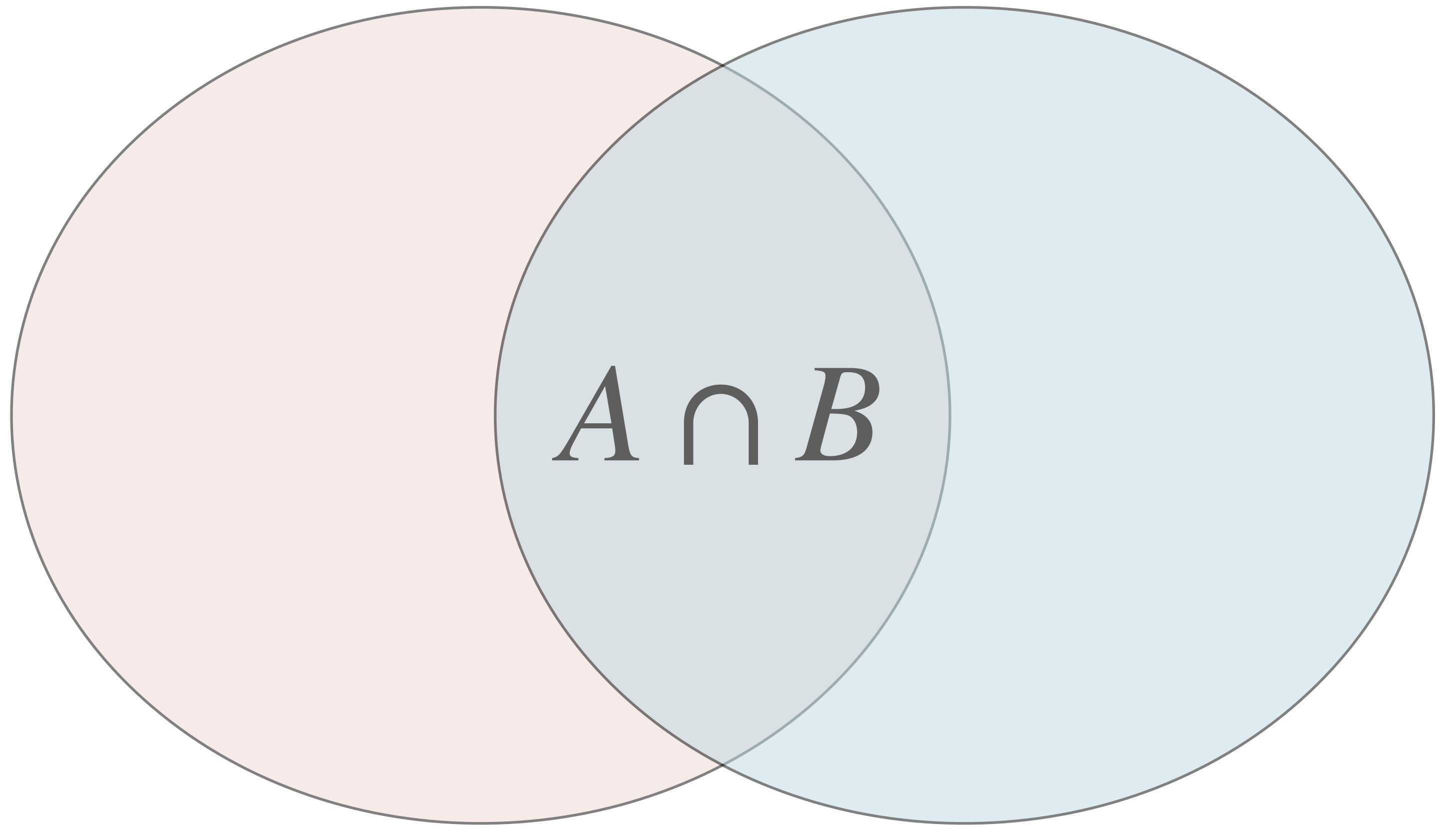
$$P(A,B) = P(B\,|\,A)\,P(A) = P(A\,|\,B)\,P(B).$$

1) $\sum_{\omega \in S} p(\omega\,|\,S) = 1$

2) If A and B are independent, then $P(B\,|\,A) = P(B)$.

Conditional Probability

Foundations of the theory of probability, A. N. Kolmogorov, 1933



$$P(B\,|\,A) = \frac{P(A, B)}{P(A)}$$

1) $\sum_{\omega \in S} p(\omega\,|\,S) = 1.$

- $P(A\,|\,A) = \dfrac{P(A \cap A)}{P(A)} = 1$
- $P(E\,|\,A) = 1$

2) If A and B are independent, then $P(B\,|\,A) = P(B)$.

- for $B \cap C = \varnothing$: $P(B \cup C\,|\,A) = P(B\,|\,A) + P(C\,|\,A)$

3) For a given (fixed) A, $(E, \mathscr{F}, P(\,\cdot\,|\,A))$ is a probability space.

Law of total probability

Let $A_1 \cup A_2 + \ldots + A_n = E$, and $A_i \cap A_j = \varnothing$ for all $i \neq j$.

For any $X \in \mathscr{F}$,

$$P(X) = \sum_{i=1}^{n} P(X \,|\, A_i)\, P(A_i).$$

Bayes' Theorem

Let $A_1 \cup A_2 + \ldots + A_n = E$, and $A_i \cap A_j = \varnothing$ for all $i \neq j$.

For any $X \in \mathscr{F}$,

$$P(A_i \,|\, X) = \frac{P(A_i)\, P(X \,|\, A_i)}{\sum_{j=1}^{n} P(A_j)\, P(X \,|\, A_j)}.$$

Working with random variables

(Notation)

When we write

$$p(X_1 \mid x_2) = \frac{p(X_1, x_2)}{p(x_2)},$$

then this notation means

$$p(X_1 = x_1 \mid X_2 = x_2) = \frac{p(X_1 = x_1, X_2 = x_2)}{X_2 = x_2} \qquad \forall x_1 \in \mathrm{Val}(X_1)$$

We write $X_1 \perp X_2$ if the two random variables are independent:

$p(X_1 = x_1, X_2 = x_2) = p(X_1 = x_1)\, p(X_2 = x_2),$

for all values $x_1 \in \mathrm{Val}(X_1)$ and $x_2 \in \mathrm{Val}(X_2)$.

- Sum rule:

$$P(A) = P(A, B) + P(A, \neg B)$$

- Product rule:

$$P(A, B) = P(A \,|\, B)\, P(B) = P(B \,|\, A)\, P(A)$$

- Bayes' theorem:

$$P(A \,|\, B) = \frac{P(B \,|\, A)\, P(A)}{P(B)} = \frac{P(B \,|\, A)\, P(A)}{P(B, A) + P(B, \neg A)}$$

Bayes' theorem

Terminology

$$\underbrace{P(X\,|\,D)}_{\textbf{posterior for X given D}} = \frac{\overbrace{P(X)}^{\textbf{prior for X}}\ \overbrace{P(D\,|\,X)}^{\textbf{likelihood for X}}}{\underbrace{P(D)}_{\textbf{evidence for the model}}} = \frac{P(X)\,P(D\,|\,X)}{\sum_{x\in\mathscr{X}} P(D\,|\,x)\,P(x)}$$

- $P(D\,|\,X)$ is the likelihood of X, but the conditional probability for D, given X.
- The prior is the marginal distribution $P(X) = \sum\limits_{d\in\mathscr{D}} P(X,d)$ under all possible data.

Expectation, Variance

- Expectation $\mathbb{E}[X]$ is the random variable X's "average" value
  - $\mathbb{E}[X] = \sum_{x \in \mathrm{Val}(X)} x\, P(X = x)$ or $\mathbb{E}[X] = \int_x x\, P(x)\, dx$

- Variance: $\mathbb{V}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \mathbb{E}[(X - \mathbb{E}[X])^2]$

standard deviation: $\sqrt{\mathbb{V}(\cdot)}$

- Chebyshev's inequality:

$$P[\,|X - \mathbb{E}[X]| \geq c\sqrt{\mathbb{V}(X)}] \leq \frac{1}{c^2}$$

e.g., for $c = 10$: probability that a value is more than 10 standard deviations away from expectation is less than 0.01.

Entropy

(PMLI, Chapter 6.1)

For discrete random variables:

$$\mathbb{H}(X) = -\sum_{k=1}^{K} p(X=k)\,\log_2 p(X=k) = -\,\mathbb{E}_X[\log_2 p(X)]$$

The entropy of a distribution: amount of uncertainty, “disorder”.

For a discrete distribution, entropy is maximised for the uniform distribution.

How would a discrete minimum entropy distribution look like (and what is it’s value)?

Multivariate distributions

Instead of one random variable, multivariate distributions have a random *vector*:

$\boldsymbol{X}(\omega) = [X_1(\omega), \ldots, X_n(\omega)]$

- $X_i = x_i$ is an event. The joint distribution

$p(X_1 = x_1, \ldots, X_n = x_n)$

is simply defined as $p(X_1 = x_1 \cap \ldots \cap X_n = x_n)$

- We will often write $p(x_1, \ldots, x_n)$ instead of $p(X_1 = x_1, \ldots, X_n = x_n)$
- Conditioning, chain rule, Bayes' rule, etc. all apply

Examples

Consider three binary-valued random variables

$$X_1, X_2, X_3 \qquad \mathrm{Val}(X_i) = \{0,1\}.$$

Let outcome space Ω be the cross-product of their states:

$$\Omega = \mathrm{Val}(X_1) \times \mathrm{Val}(X_2) \times \mathrm{Val}(X_3)$$

- $X_i(\omega)$ is the value for X_i in the assignment $\omega \in \Omega$
- Specify $p(\omega)$ for each outcome $\omega \in \Omega$ using a big table.
- How many parameters do we need to specify?

$2^3 - 1$

x_1	x_2	x_3	$p(x_1x_2x_3)$
0	0	0	0.11
0	0	1	0.02
	…		
1	1	1	0.05

Marginalisation

joint distribution	Very high	High
p	0.7	0.15
f	0.1	0.05

- Marginals: probabilities of individual variables

- Suppose X and Y are random variables with distribution $p(X, Y)$

X : Intelligence, $\text{Val}(X) = \{''\text{Very high}'', ''\text{High}''\}$

Y : Grade, $\text{Val}(Y) = \{''\text{p}'', ''\text{f}''\}$

- $p(Y = p) = \,? = 0.85$

Marginalisation

Suppose we have a joint distribution $p(X_1, \ldots, X_n)$.
Then,

$$p(X_i = x_i) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_{i-1}} \sum_{x_{i+1}} \cdots \sum_{x_n} p(x_1, \ldots, x_n)$$

Marginalisation means: summing away all but the random variable(s) of interest

Conditioning

Suppose X and Y are random variables with distribution $p(X, Y)$

X : Intelligence, $\mathrm{Val}(X) = \{\text{"Very high"}, \text{"High"}\}$

Y : Grade, $\mathrm{Val}(Y) = \{\text{"p"}, \text{"f"}\}$

	Very high	High
p	0.7	0.15
f	0.1	0.05

Compute the conditional probability

$$
\begin{aligned}
p(Y = p \mid X = vh) &= \frac{p(Y = p, X = vh)}{p(X = vh)} \\
&= \frac{p(Y = p, X = vh)}{p(Y = p, X = vh) + p(Y = f, X = vh)} \\
&= \frac{0.7}{0.7 + 0.1} = 0.875
\end{aligned}
$$

Example: Medical diagnosis

A variable for each symptom (e.g., “fever”, “cough”, “fast breathing”, “shaking”, “nausea”, “vomiting”)

A variable for each disease (e.g., “pneumonia”, “flu”, “common cold”, “covid”, “tuberculosis”)

Diagnosis is performed by inference on the model:

$$p(\text{pneumonia} = 1 \,|\, \text{cough} = 1, \text{fever} = 1, \text{vomiting} = 0)$$

The Quick Medical Reference (QMR-DT) model has 600 diseases and 4000 symptoms.

Representing the distribution

We could represent multivariate distributions with a table of probabilities for each outcome

- How many outcomes in QMR-DT? 2^{4600}

Estimation of joint distribution would require a huge amount of data, and inference of conditional probabilities, e.g., for

$$p(\text{pneumonia} = 1 \mid \text{cough} = 1, \text{fever} = 1, \text{vomiting} = 0)$$

would require summing over exponentially many variables' values.

This would also defeat the purpose of probabilistic modelling, which is to make predictions with *previously unseen observations*.

Structure through independence

If $X_1, \ldots, X_n$ are independent, then $p(x_1, \ldots, x_n) = p(x_1)\,p(x_2) \ldots p(x_n)$.

- 2^n entries can be described by just n numbers (if $|\mathrm{Val}(X_i)| = 2$)

This is unfortunately not a very *useful* model. Observing a variable X_i cannot influence predictions of X_j.

If $X_1, \ldots, X_n$ however are conditionally independent given Y, (written as $X_i \perp \boldsymbol{X} \mid Y$) then

$$\begin{aligned} p(y, x_1, \ldots, x_n) &= p(y)\,p(x_1 \mid y) \prod_{i=2}^{n} p(x_i \mid x_1, \ldots, x_{i-1}, y) \\ &= p(y)\,p(x_1 \mid y) \prod_{i=2}^{n} p(x_i \mid y)\,. \end{aligned}$$

This is a simple, yet powerful model.

Example: naîve Bayes for classification

Classify emails as spam ($Y = 1$) or not spam ($Y = 0$).

- $i : 1...n$ index the words in our vocabulary (e.g., all English words)
- $X_i = 1$ if word i appears in an email, and 0 otherwise
- Emails are drawn according to some distribution $p(Y, X_1, \ldots, X_n)$

Suppose that the words are conditionally independent given Y. Then,

$$p(y, x_1, \ldots, x_n) = p(y) \prod_{i=1}^{n} p(x_i \mid y)$$

Estimate the model with maximum likelihood. Predict with:

$$p(Y = 1 \mid x_1, \ldots, x_n) = \frac{p(Y = 1) \prod_{i=1}^{n} p(x_i \mid Y = 1)}{\sum_{y=\{0,1\}} p(Y = y) \prod_{i=1}^{n} p(x_i \mid Y = y)}$$

- Are the independence assumptions made here reasonable?

Philosophy: Nearly all probabilistic models are “wrong”, but many are nonetheless useful.

Graphical models

Graphs, formal definitions

A graph is a data structure $K = (V, E)$, where V is the set of vertices (nodes), and E is the set of edges.

- $V = \{X_1, \ldots, X_n\}$.
  A pair of nodes, X_i, X_j can be connected by a directed edge $X_i \rightarrow X_j$ or an undirected edge $X_i - X_j$.

- The set of edges E is a set of pairs; each pair is one of $X_i \rightarrow X_j$, $X_j \rightarrow X_i$, or $X_i - X_j$, for $X_i, X_j \in V$, $i < j$.
  To denote any connection (directed or undirected): $X_i \rightleftharpoons X_j$

Graphs that have only directed edges are called *directed graphs*.
In the PGM book, directed graphs are usually written using the letter G.

Graphs with only undirected edges are *undirected graphs*, denoted with H.

- We can get the undirected version of a graph $K = (V, E)$ by replacing all edges with undirected edges:
  $H = (V, E')$, where $E' = \{X - Y : X \leftrightharpoons Y \in E\}$.

Graphs, formal definitions

$X_i \rightarrow X_j \in E$: X_j is a *child* of X_i in K.

Notation: P_{a_X}, Ch_X : parents, children of X

$X_i - X_j \in E$: X_i, X_j are *neighbours*

Notation: Nb_X : neighbours of X

The set of all parents and neighbours of X combined, $\mathrm{Pa}(X) \cup \mathrm{Nb}(X)$, is called the *boundary* of X.

Notation: $\mathrm{Boundary}_X$

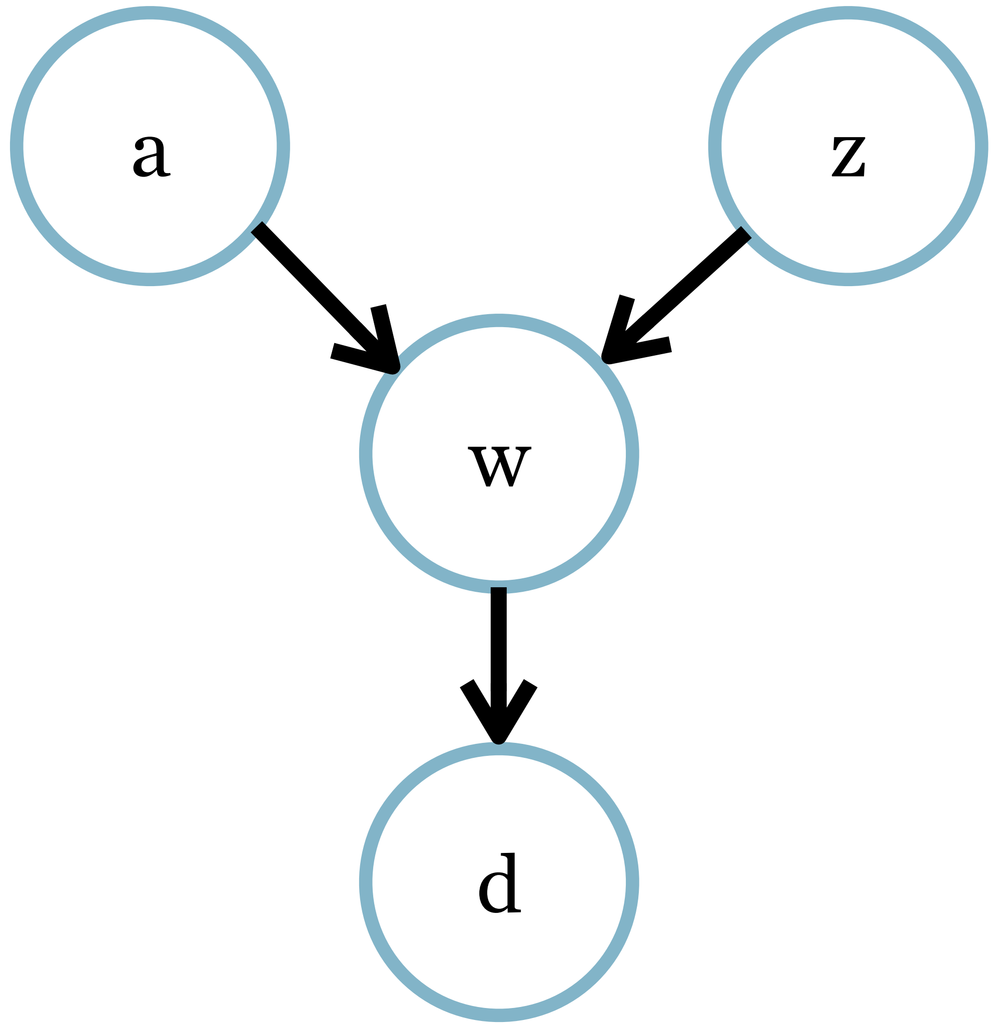
Topological ordering

For directed graphs

- A linear ordering of the nodes so that
  for every edge from *u* to *v* the node *u* comes before *v* in the ordering.

- This requires the graph to have no cycles, i.e., requires directed acyclic graphs (DAG).

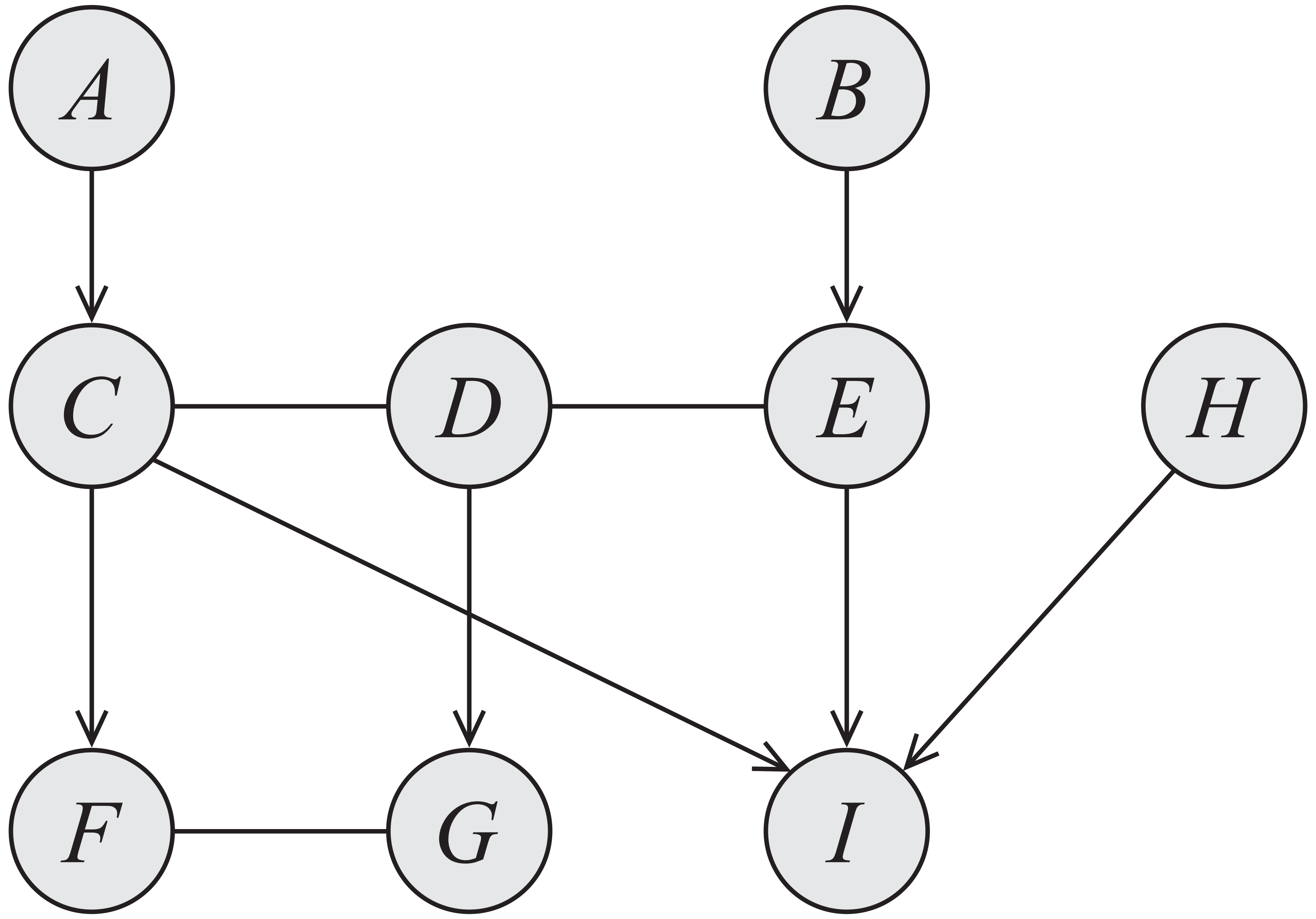
- In general, multiple topological orderings are possible for the same graph.

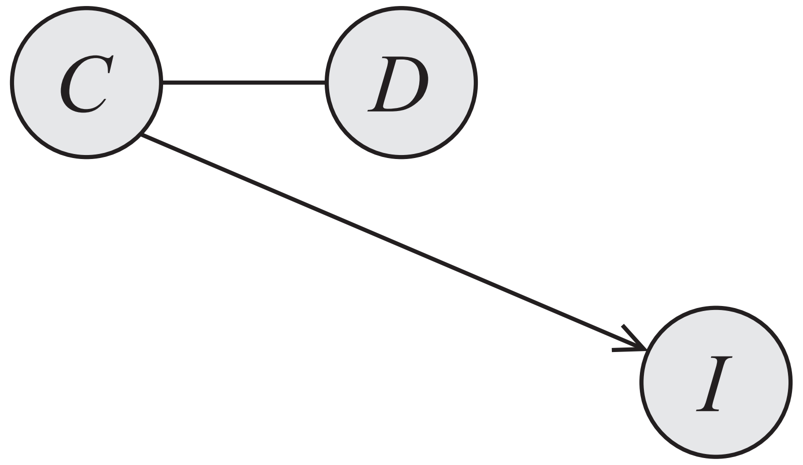


Solutions:
(a, z, w, d) (z, a, w, d)

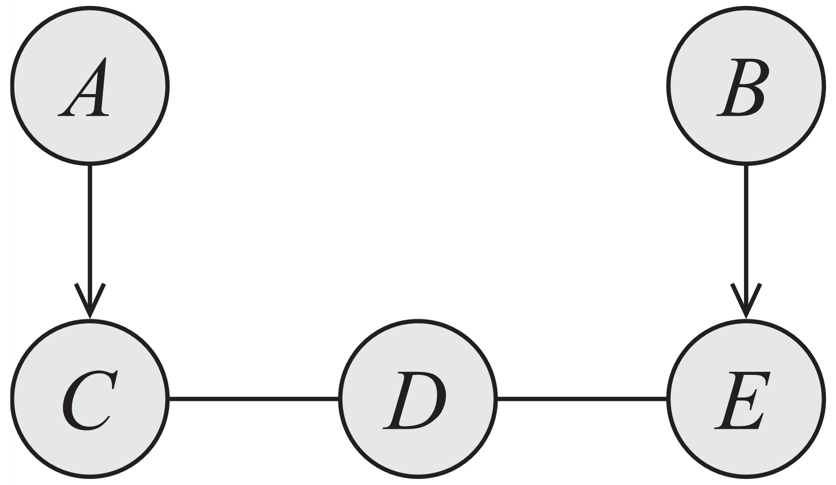
Not a solution:
(a, w, d, z)

(or anything else, really)

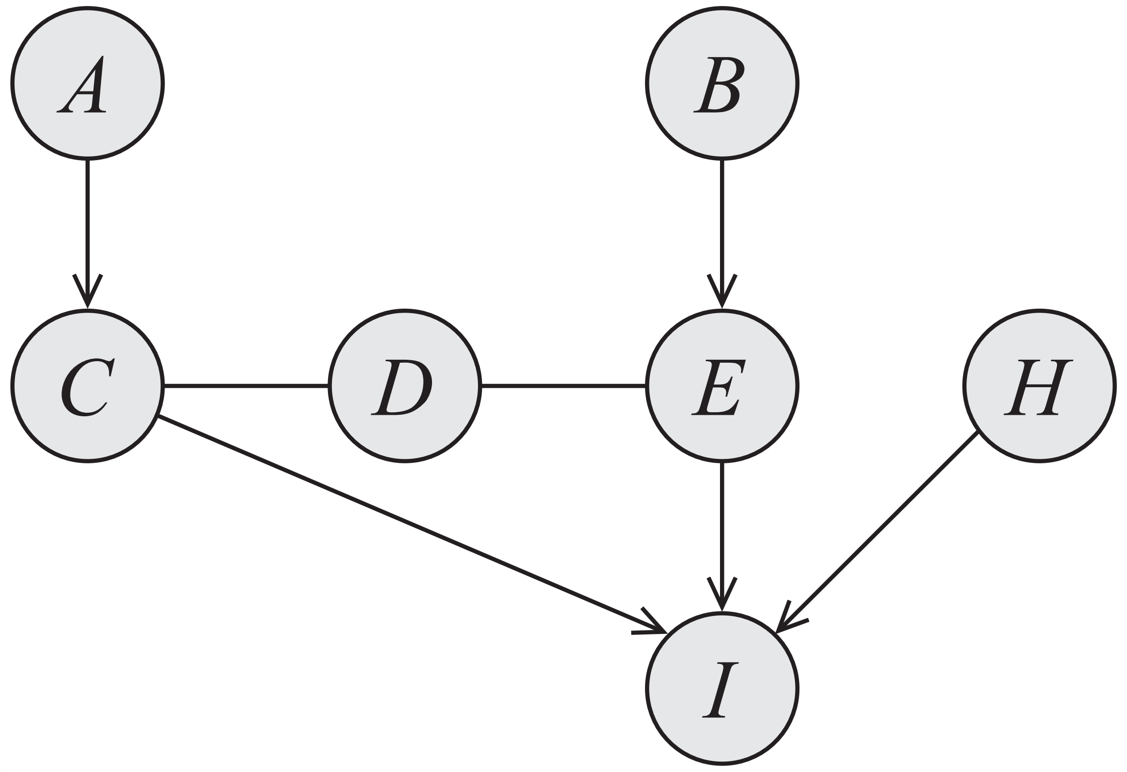




(a)



(b)



(c)

(a) $K = (V, E)$; let $\boldsymbol{X} \in V$. The induced subgraph $K[\boldsymbol{X}]$ is the graph $(\boldsymbol{X}, E')$, where E' are all the edges $X \rightleftharpoons Y \in E$ such that $X, Y \in \boldsymbol{X}$.
(example: $K[C, D, I]$)

- A subgraph over $\boldsymbol{X}$ is complete if every two nodes in $\boldsymbol{X}$ are connected by some edge.The set $\boldsymbol{X}$ is often called a clique; we say a clique $\boldsymbol{X}$ is maximal if for any superset of nodes $\boldsymbol{Y} \supset \boldsymbol{X}$, $\boldsymbol{Y}$ is not a clique.

(b) and (c) A subset of nodes $\boldsymbol{X} \in V$ is upwardly closed in K if, for any $X \in \boldsymbol{X}$, we have that $\text{Boundary}_X \subset \boldsymbol{X}$. The upward closure of $\boldsymbol{X}$ is the minimal upwardly closed subset $\boldsymbol{Y}$ that contains $\boldsymbol{X}$. The upwardly closed subgraph of $\boldsymbol{X}$, denoted $K^+[\boldsymbol{X}]$, is the induced subgraph over $\boldsymbol{Y}$, $K[\boldsymbol{Y}]$.

Bayesian networks

PGM chapter 3

A Bayesian network is specified by a directed acyclic graph $G = (V, E)$ with:

1. One node $i \in V$ for each random variable X_i
2. One conditional probability distribution (CDP) per node, $p(x_i \,|\, \boldsymbol{x}_{\mathrm{Pa}(i)})$, specifying the variable's probability conditioned on its parents' values.

Corresponds 1-1 with a particular factorisation of the joint distribution:

$$p(x_1, \ldots, x_n) = \prod_{i \in V} p(x_i \,|\, \boldsymbol{x}_{\mathrm{Pa}(i)})$$

Powerful framework for designing algorithms to perform probability computations.

That’s it for today

Please also go to your tutorials!