COGS 118A, Winter 2020

Supervised Machine Learning Algorithms

Zhuowen Tu

Department of Cognitive Science

UC San Diego

Syllabus

Supervised Machine Learning Algorithms: Syllabus

Lecture Time:

12:30PM-1:50PM, Tuesday and Thursday, MANDE B-210

We will be using iClicker during the lectures to increase the classroom engagement.

Study Sections:

A01: Wednesday 9:00AM-9:50AM CENTR 222
A02: Wednesday 10:00AM-10:50AM CENTR 222
A03: Wednesday 11:00AM-11:50AM CENTR 222

TA:

Yifan Xu (yix081@ucsd.edu)
Weijian Xu (wex041@eng.ucsd.edu)

IA:

Yikai Hao (yih307@ucsd.edu)
Yilan Jiang (yij007@ucsd.edu)
Ansuman Somasundaram (ansomasu@ucsd.edu)
Mrinal Verghese (mtverghe@ucsd.edu)
Ziwen Zeng (ziz236@ucsd.edu)
Yuqi Zhang(yuz796@ucsd.edu)
Aaron Wong (aaw016@ucsd.edu)

Web Resources

Piazza
Podcast
Gradescope

Text Books:

1. Christopher M. Bishop, "Pattern Recognition and Machine Learning", 2006.
2. R. Duda, P. Hart, D. Stork, "Pattern Classification", second edition, 2000. here

This course is self-contained; having the textbook is helpful but not absolutely necessary.

Office Hours:

Zhuowen Tu, 2:00PM-3:00PM, Tuesday and Thursday, CSB 132
Yifan Xu, 10:00AM - 11:00AM, Monday CSB132
Weijian Xu, 9:00AM - 10:00AM, Friday CSB132
Mrinal Verghese, 2:00PM - 3:00PM, Thursday CSB 114
Yikai Hao, 4:00PM - 5:00PM, Monday CSB 114
Yuqi Zhang 10:00AM - 11:00AM, Friday CSB132
Yilan Jiang, 3:00PM - 4:00PM, Wednesday CSB 132
Ziwen Zeng, 2:00PM - 3:00PM, Wednesday CSB 132
Ansuman Somasundaram, 4:00PM - 5:00PM, Wednesday CSB 114
Aaron Wong, 3:00PM - 4:00PM, Monday CSB 132

Piazza

Please enroll in this webpage to receive class notification.

Course Description:

Supervised Machine Learning Algorithms: this course will prepare the students in basics of the statistical classification methods which will likely serve the foundation for data analysis and inference in a variety of applications. It will also be helpful in learning more advanced statistical machine learning algorithms, which have been applied in a wide range of scenarios for studying and predicting cognitive models, financial models, social behaviors, brain growth patterns, and visual inference.

You will need to use Python to do your assignments and final project.

Prerequisites:
Mathematics 20F (Linear Algebra) or Mathematics 31AH (Honors Linear Algebra), and Mathematics 180A (Introduction to Probability) or ECE 109 (Engineering Probability & Statistics), and COGS 109 (Modeling and Data Analysis) or CSE 11 (Introduction to Computer Science & Object-Oriented Programming: Java), or consent of instructor.

Grading policy:
Assignments: 38%
Classroom participation: 2%
Midterms: 40%
Final project: 20%
Bonus points: 3% (Piazza activities + final project)

Late policy: 5% reduction for the first day and 10% reduction afterwards for every extra day past due for the homework assignments and the final project.

Policy on Integrity of Scholarship

COGS 118A, Spring 2019

Grading Policy

Assignments: 38%

Classroom participation: 2%

Midterm exams: 40%

Final exam/project: 20%

Bonus points: 3% (classroom participation + Piazza activities + final project)

Course Webpage

https://sites.google.com/site/ucsdcogs118awinter2020/

Piazza

https://piazza.com/class/fall2018/cogs118a

Class Schedule

Calendar and Class Notes

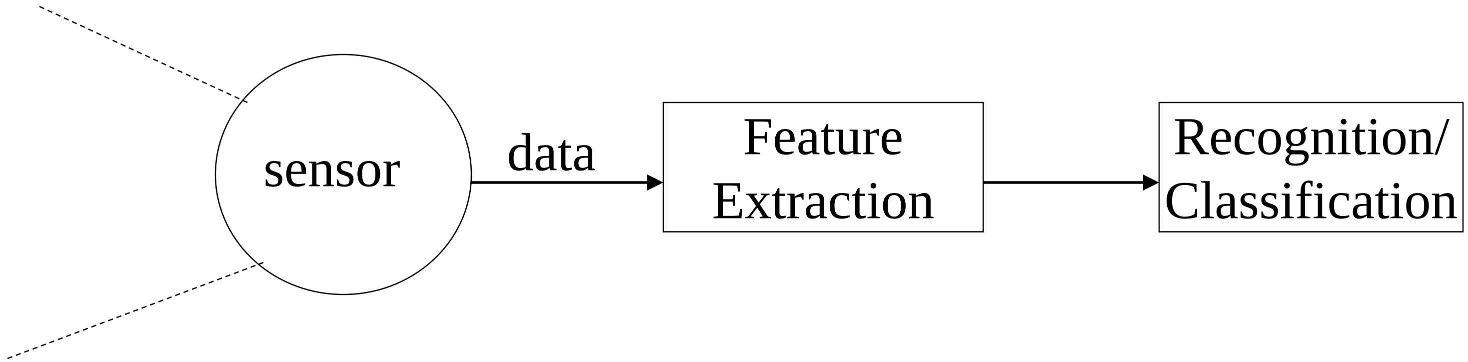
Date and Lecture #	Topic	Readings (most of them will be optional, unless specified as "required")	Video resources
Week 1:	Course overview, introduction to machine learning, real-world applications and impacts, cognitive science applications Slides Ch 1. Introduction (K. Murphy) Ch 1. Introduction (Duda et al.)	Linear algebra review Probability theory (by Matthew Shum) Python tutorial Python 2.7 Documentation Python 3.6 Documentation Jupyter Notebook Documentation Math and Matrix Operations to Python **Other useful things to reads:** Introduction to probability by C.M. Grinstead and J.L. Snell A few useful things to know about machine learning(Pedro Domingos) IBM Watson	Vectors (by 3Blue1Brown) Linear transformations and matrices (by 3Blue1Brown) UC Irvine ML: Introduction (Alexander Ihler)
	Review of linear algebra and vector calculus Part I.2 Linear Algebra (Goodfellow et al.) Data formulation and problem definition	Matrix calculus A Visual Introduction to Machine Learning	UC Irvine ML: Data (Alexander Ihler) UC Irvine ML: Probability (Alexander Ihler)
Week 2:	Decision boundary	Chapter 2 in Murphy's book Review of Probability Theory Conditional Probability	
	Estimation Decision stump classifier Ch 1.1 Example: Polynomial Curve Fitting (C. Bishop) Ch 1.5 Decision Theory (C. Bishop)	Precision and recall Receive operating characteristic Linear regression	UC Irvine ML: Supervised Learning (Alexander Ihler)
Week 3:	Convexity Linear regression Ch 3.1 Linear Basis Function Models (C. Bishop)	Ordinary Least Squares Regression	UC Irvine ML: Complexity and overfitting (Alexander Ihler)
	Robust estimation Gradient descent and Error metrics	Stochastic Gradient Descent An overview of gradient descent Gradient descent	UC Irvine ML: Linear regression (Alexander Ihler)
Week 4:	Perceptron	https://en.wikipedia.org/wiki/Perceptron https://en.wikipedia.org/wiki/Artificial_neural_network	Neural Networks (3Blue1Brown)
	Midterm I	A Visual Introduction to Machine Learning	UC Irvine ML: Gradient Descent (Alexander Ihler)

Class Schedule

Week 5:	Logistic regression classifier	logistic regression	UC Irvine ML: Regression (Alexander Ihler)
	Logistic regression classifier		Logistic Regression(Andrew Ng)
Week 6:	Complexity, VC-dimension Structural Risk Minimization Cross-validation	An overview of statistical learning theory SVM Tutorial	Regularization and Overfitting (Andrew Ng)
	Support Vector Machine		
Week 7:	Support Vector Machine	"Classification and regression trees", Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J., 1984.	UC Irvine ML: Duals (Alexander Ihler)
	Kernels		UC Irvine ML: Kernels (Alexander Ihler)
Week 8:	Nearest neighborhood	Chapter, "Non-parametric Techniques", R. Duda, P. Hart, D. Stork, "Pattern Classification", second edition, 2000	UC Irvine ML: Kernels (Alexander Ihler)
	Midterm 2		
Week 9:	Decision tree	Decision tree (Wiki) "C4.5: Programs for Machine Learning", Quinlan, J. R., 1993. K-D tree (Wiki) "K-D Tree Tutorial", Andrew Moore A Visualization of decision tree (part 1) A Visualization of decision tree (part 2)	
	Ensemble classifier Random Forests		
Week 10:	Boosting	Bagging Predictors", Leo Breiman. "Shape quantization and recognition with randomized trees", Y Amit, D Geman, 1997. "Random Forests", Leo Breiman. "A decision-theoretic generalization of on-line learning and an application to boosting", Yoav Freund and Robert E. Schapire, 1997. "Improved boosting algorithms using confidence-rated predictions", Robert E. Schapire and Yoram Singer, 1999. "Additive Logistic Regression: a Statistical View of Boosting", Jerome Friedman , Trevor Hastie , Robert Tibshirani, 1998 Xgboost Github	
	Boosting		

What is Pattern Recognition and Machine Learning?

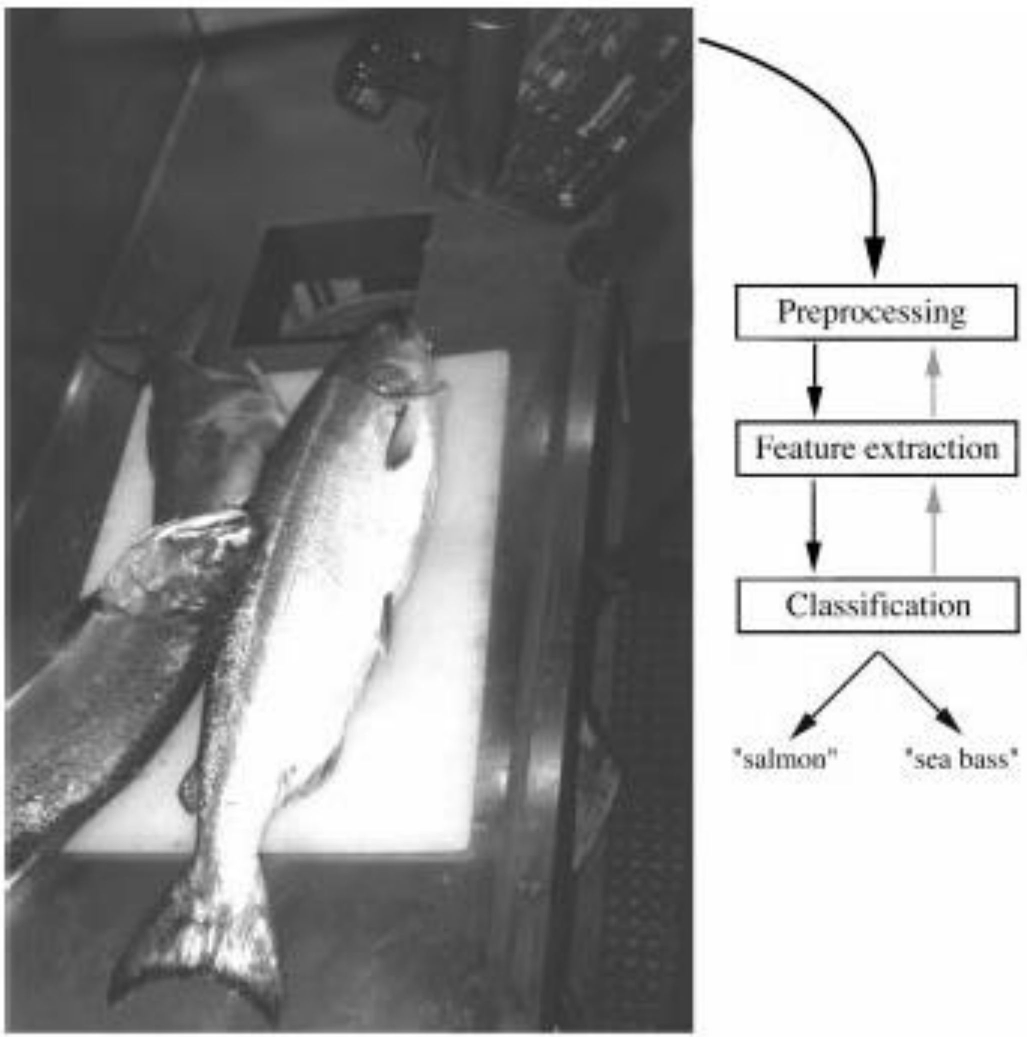
Definition (S. Schmidt): A process of identifying a stimulus. Recognizing a correspondence between a stimulus and information in permanent memory.



This process is often accomplished with incomplete or ambiguous information.

Many variations on a pattern may be recognized as the same class.

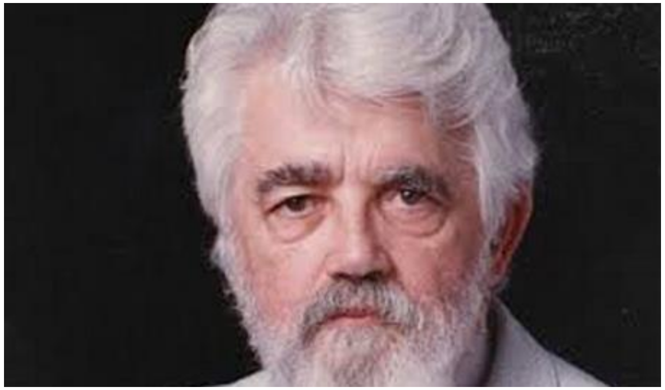
An Example



Preprocessing
Feature extraction
Classification
"salmon"
"sea bass"

Artificial Intelligence

The name of "Artificial Intelligence" was given in 1956 on a meeting held at Dartmouth College by John McCarthy.



What is intelligence?

The term intelligence covers many cognitive skills, including the ability to solve problems, learn, and understand language; AI addresses all of those. But most progress to date in AI has been made in the area of problem solving -- concepts and methods for building programs that reason about problems rather than calculate a solution.

AI's scientific goal is to understand intelligence by building computer programs that exhibit intelligent behavior. It is concerned with the concepts and methods of symbolic inference, or reasoning, by a computer, and how the knowledge used to make those inferences will be represented inside the machine.

AI programs that achieve expert-level competence in solving problems in task areas by bringing to bear a body of knowledge about specific tasks are called knowledge-based or expert systems.

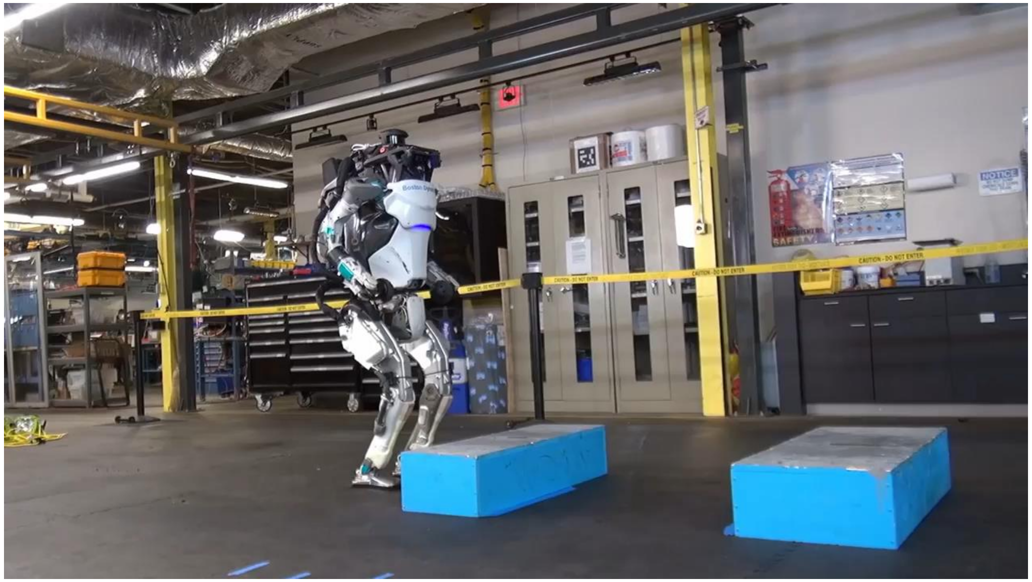
http://en.wikipedia.org/wiki/Edward_Feigenbaum

What is Artificial Intelligence (AI)?

A.I.

Where are we now?

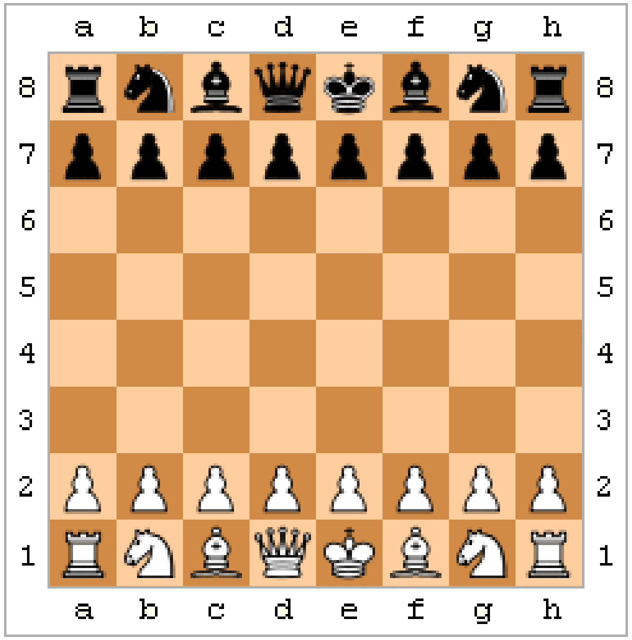
DARPA 2015 Challenge



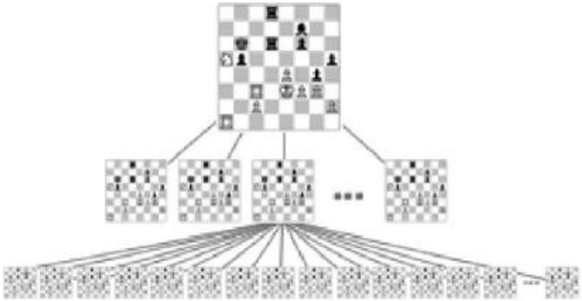
Boston Dynamics Inc.

More recently:
https://www.youtube.com/watch?v=_sBBaNYex3E

Playing Chess



It is about building a effective search strategy that computers are particularly good at, once well-designed. Humans are only able to look beyond 2-4 steps ahead.



https://en.wikipedia.org/wiki/Computer_chess

Difference Aspects of AI

Logic reasoning

Knowledge representation (acquisition and abstraction): expert systems

Machine learning: mostly statistics-driven with measurable performances

Main machine learning conferences and journals

Flagship conferences:

- Neural Information Processing Systems (NeruIPS, https://nips.cc/)
- International Conference on Machine Learning (ICML, http://icml.cc/)

Conferences with focused topics:

- Conference on Learning Theory (COLT)
- Artificial Intelligence and Statistics (AISTATS)
- International Conference on Learning and Representation (ICLR)
- Conference on Uncertainty in Artificial Intelligence (UAI)

Topics to be covered in COGS 118A

- Representation and problem formulation
- Decision boundaries and vector calculus
- Errors and optimization
- Linear regression
- Perceptron
- Logistic regression
- Supervised classification: support vector machines, kernels
- Supervised classification: boosting, random forests
- Decision trees

Potential pitfalls

- It is not just about data.
- Representation is the key.
- Top-down and bottom-up information are equally important.
- Need to understand human cognition to gain insights.
- Neural structures and statistics.

What humans are good at

- Knowledge abstraction
- Adaptation and online learning
- Fine-grained reasoning
- Understanding the context
- We have feelings

What computers are good at

- Storing and retrieving big data
- Fast large-scale numerical computing
- Fault tolerant
- Strict reasoning given the rules

Statistical learning + Representation + Infrastructure + Data

Understand your problem

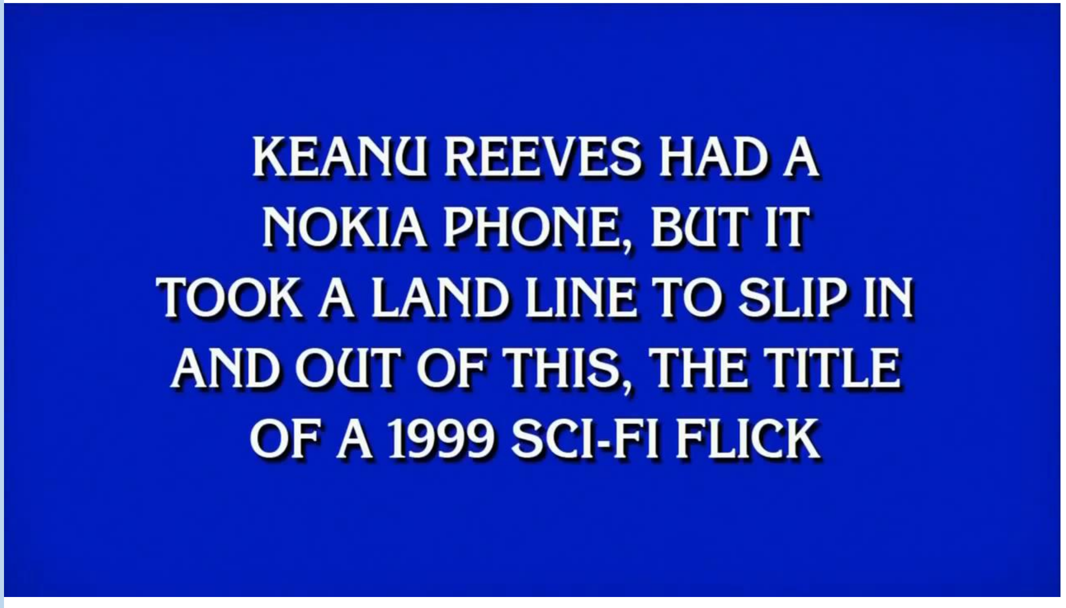
Input (input space)

Output (output space)

Why is machine learning difficult?

https://www.youtube.com/watch?v=sgqSatgoTbk

Ambiguities and uncertainties in machine learning



https://www.youtube.com/watch?v=sgqSatgoTbk

Some lessons learned

- Learning = Representation + Evaluation + optimization
- It’s Generalization that counts
- Data alone is not enough
- Overfitting has many faces
- Intuition Fails in high Dimensions
- Theoretical Guarantees are not What they seem
- Feature engineering is the Key
- More Data Beats a cleverer algorithm
- Learn many models, not Just one
- Simplicity Does not imply Accuracy
- Representable Does not imply Learnable
- Correlation Does not imply Causation

Pedro Domingos

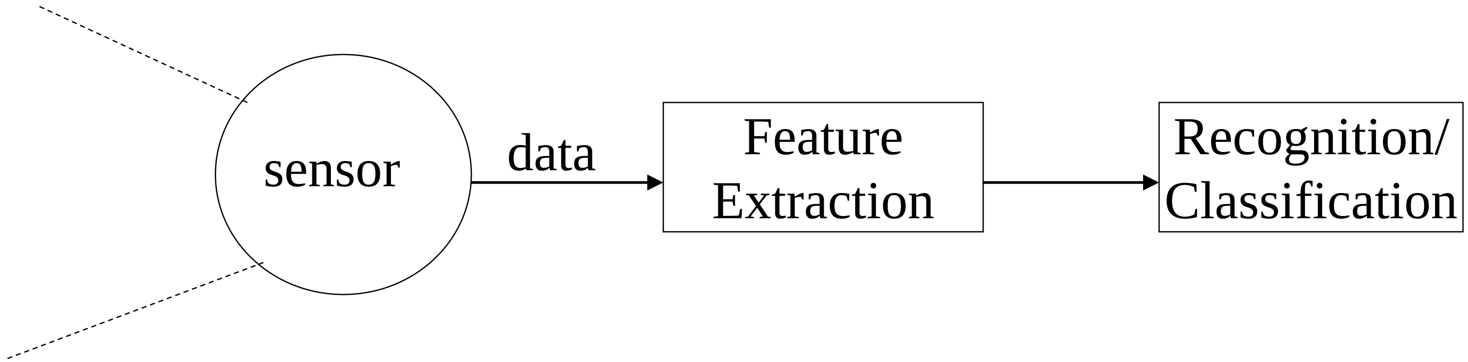
Some Learned Lessons (Pedro Domingos)

Table 1. The three components of learning algorithms.

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

What is Pattern Recognition and Machine Learning?

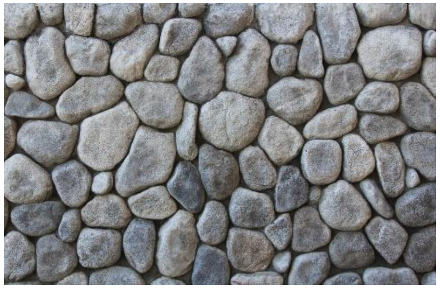
Definition (S. Schmidt): A process of identifying a stimulus. Recognizing a correspondence between a stimulus and information in permanent memory.



This process is often accomplished with incomplete or ambiguous information.

Many variations on a pattern may be recognized as the same class.

What is a pattern?

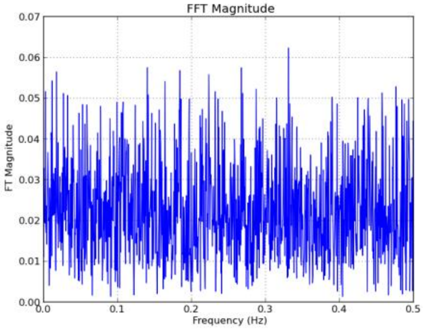


by TextureX-com

Texture?

https://commons.wikimedia.org/wiki

Objects?



https://math.stackexchange.com/

Randomness?

What is a pattern?

A. Being repetitive.

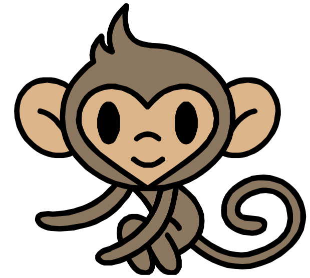
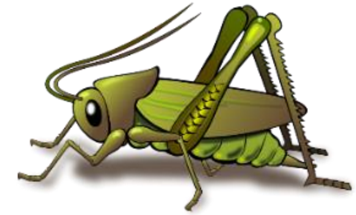
B. Share common features.

C. The definition is subjective.

D. Explicit and implicit descriptions.

E. All of above.

What is not a pattern?

Not being repetitive?

Does not share common features?

No “pattern” is also a **pattern**.

Position ourselves

Roughly speaking:

Before 1997:
Manually defined logics and features with some rules and simplified statistical models on relatively small data.

1997-2006:
Manually designed features with principled statistical learning on relatively small data.

2006-present:
Automatically learned features/representations on big data with/without deep learning.

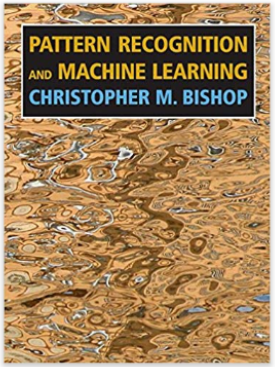
Online shopping

Pattern Recognition and Machine Learning (Information Science and Statistics)

by Christopher M. Bishop (Author)

185 ratings

Look inside



ISBN-13: 978-0387310732
ISBN-10: 0387310738
Why is ISBN important?

Have one to sell? Sell on Amazon

Add to List

Share

Hardcover	Paperback	Other Sellers
$51.47	$61.10	See all 7 versions

Buy new $51.47

Only 8 left in stock - order soon.
Ships from and sold by Sparkle Books.

List Price: ~~$94.95~~
Save: $43.48 (46%)
13 New from $51.47

Get it as soon as Jan. 13 - 16 when you choose **Expedited Shipping** at checkout.

Select delivery location

Qty: 1

$51.47 + Free Shipping

Add to Cart

Buy Now

More Buying Choices

13 New from $51.47 | 13 Used from $51.38

26 used & new from $51.38

See All Buying Options

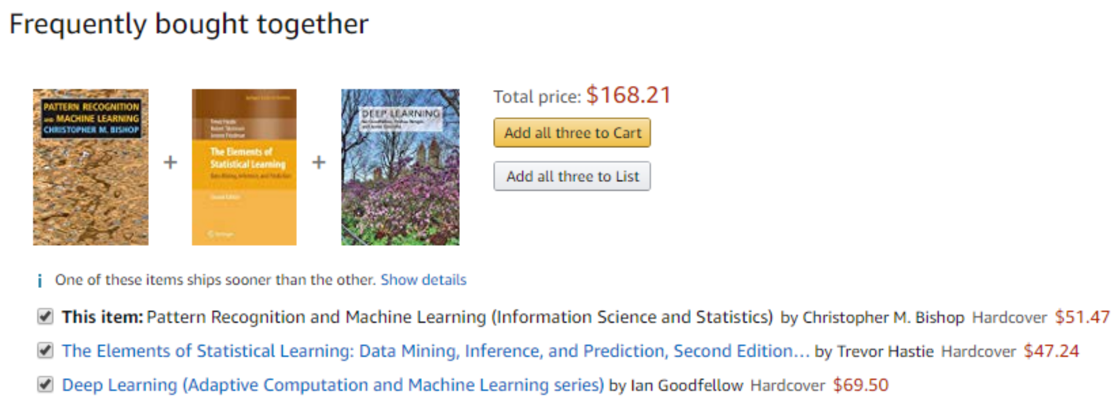
This is the first textbook on pattern recognition to present the Bayesian viewpoint. The book presents approximate inference algorithms that permit fast approximate answers in situations where exact answers are not feasible. It uses graphical models to describe probability distributions when no other books apply graphical models to machine learning. No previous knowledge of pattern recognition or machine learning concepts is assumed. Familiarity with multivariate calculus and basic linear algebra is required, and some experience in the use of probabilities would be helpful though not essential as the book includes a self-contained introduction to basic probability theory.

Are you going to buy this book?

Yes or No

Online shopping

Direct recommendations:



Other recommendations:

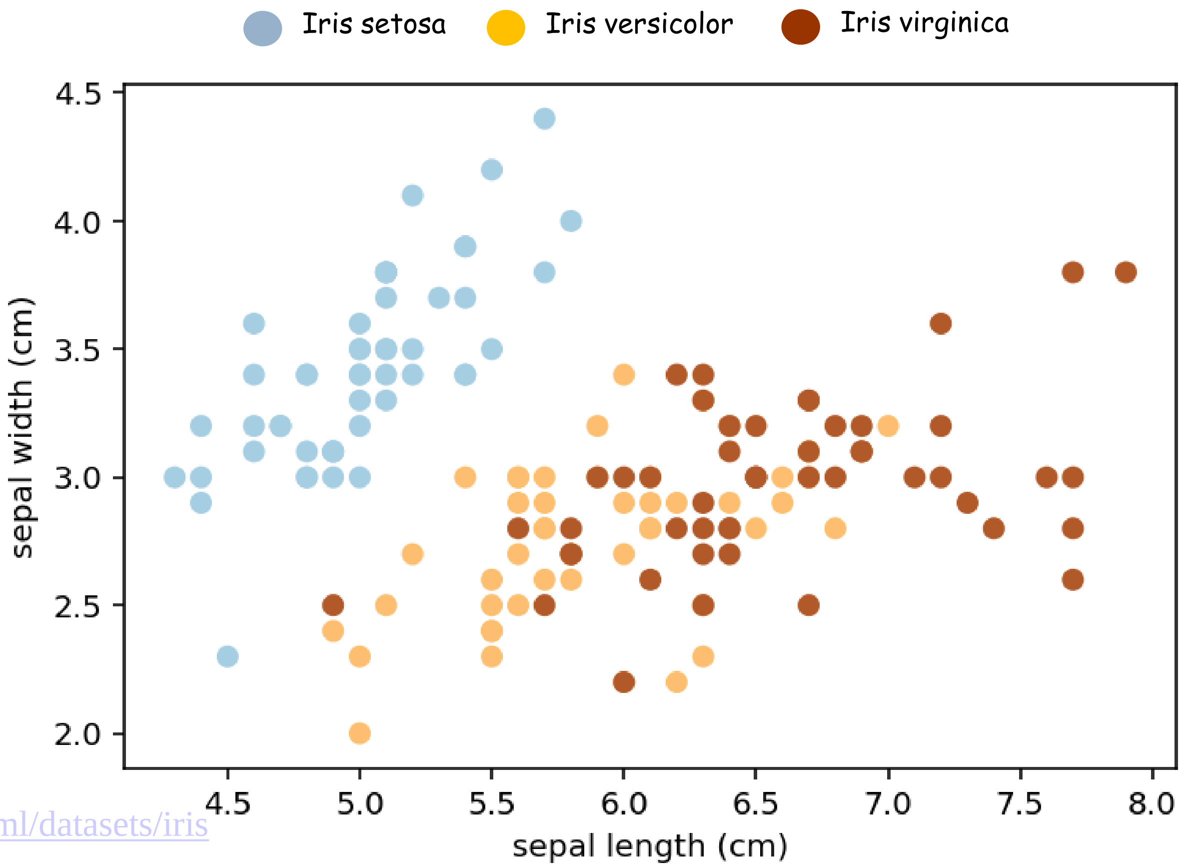
Your clicks help training/improving the underlying machine learning algorithms.

Visualization of First 2 Features

Iris setosa

Iris versicolor

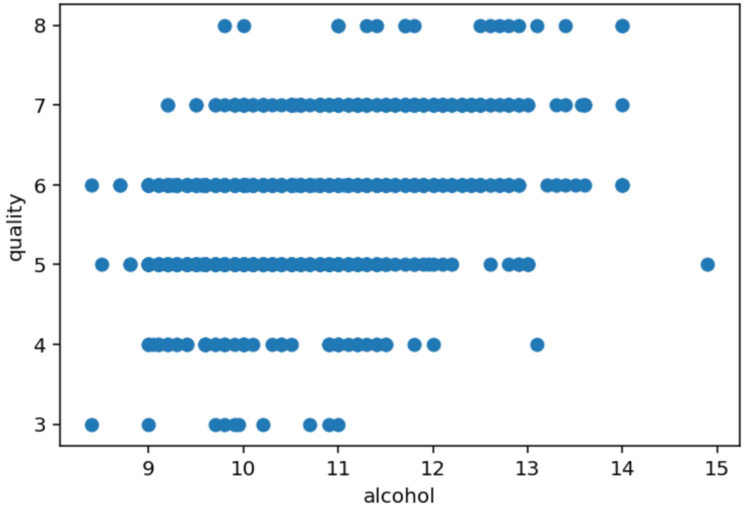
Iris virginica



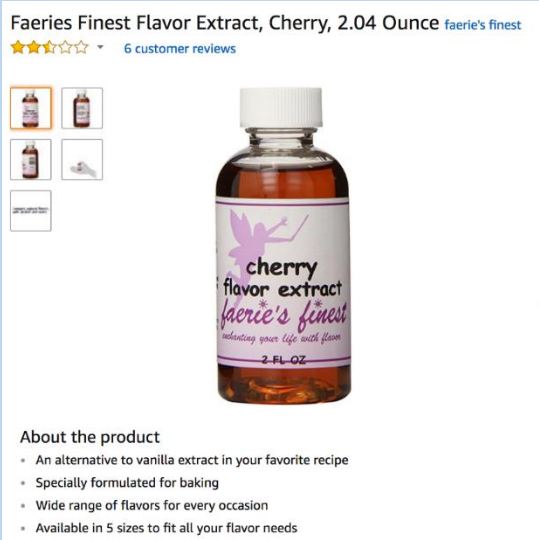
Dataset: https://archive.ics.uci.edu/ml/datasets/iris
Image from Wikipedia

Red Wine Dataset

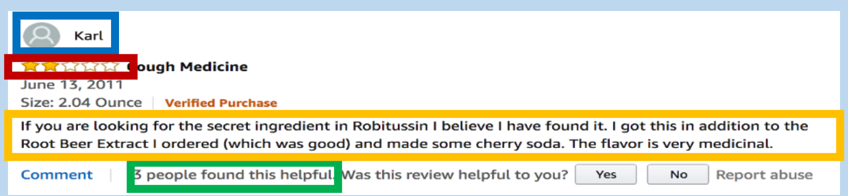
- 1599 data points, each one has:
  - 11 numerical features:
    - Fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol.
  - 1 numerical target:
    - Quality (0 to 10)



Dataset: https://archive.ics.uci.edu/ml/datasets/wine+quality
or https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009
Image from WineMag.com



Pipeline



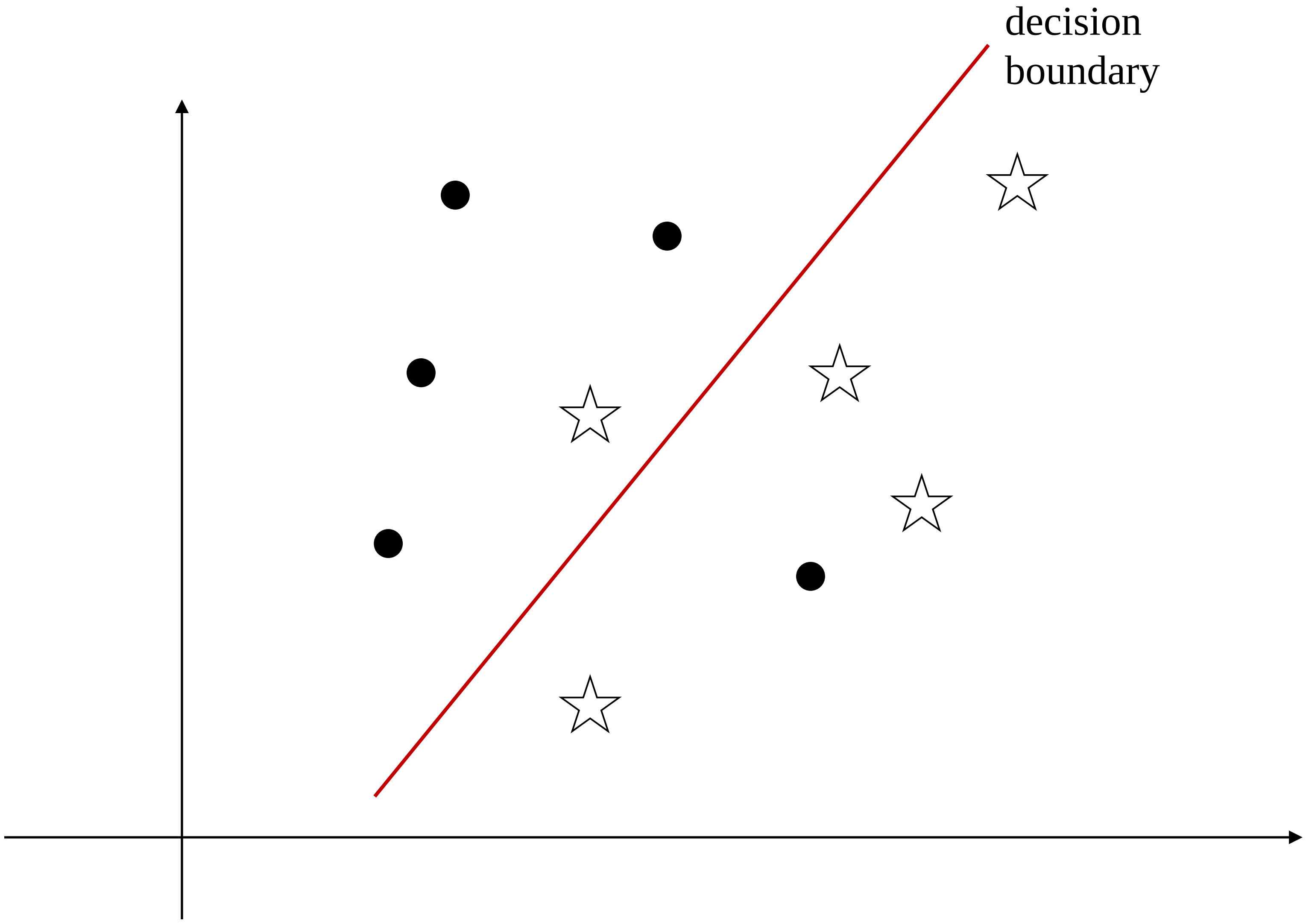
- Find data source.
- Crawl the data.
- Perform data cleansing.
- Data processing and visualization.
- Training your machine learning algorithm.

Data Cleansing

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g. instrument fault, human or computer error, transmission error
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g. *ProductID* = “ ” (missing data)
  - noisy: containing noise, errors, or outliers
    - e.g. *Time* = “−10” (an error)
  - inconsistent: containing discrepancies in codes or names, e.g.,
    - e.g. *HelpfulnessDenominator = 0, HelpfulnessNumerator = 1*
  - intentional (e.g., disguised missing data)
    - e.g. *Text* = “Please write the review here.”

From *Data Mining: Concepts and Techniques*, Jiawei Han et al.

Train your classifier



What you are supposed to know and will be strengthened

- Linear algebra and basics about vector calculus
- Basics about probability theory
- Python/iPython programming
- Calculus and numeric analysis

What you will learn

- Understand data representations
- Know how to formulate your problem using sound mathematical formulations
- Be comfortable with optimization
- Understand the essence of various supervised learning methods
- Implement your own classifier and know how and know to use existing ML packages

What you will be able to do in the end

- Given a standard classification task, know how to collect, store, and convert the data.
- Able to connect your data to right mathematical formulations.
- Train classifiers using a wide variety of machine learning algorithms.
- Build a data processing pipeline by taking input, building internal representation, to training classifiers to produce the output.
- Be hands-on to properly make your choice, tune hyper-parameters, and know to interpret your results.

Reasons for you NOT to take COGS 118A

- COGS 118A is not just a introductory class to machine learning.
- I can learn COGS 118A without knowing the math.
- Writing Python code to connect abstract concepts with the mathematical representations is too hard for me.
- Professor Tu cannot teach the class well and he is boring.
- Getting a job offer in machine learning requires more than COGS 118A.
- The slides are messy and I don’t understand the materials at all.
- …

Reasons for you to consider taking COGS 118A

- COGS 118A gives an overview of the basic supervised machine learning techniques.
- I am willing to take the challenge from building conceptual understanding, deriving sound mathematical formulations, to making effective implementations.
- Taking COGS 118A makes me better prepared for my future in-depth study of machine learning theory and applications.
- It is fun to apply machine learning techniques to solve real-world problems.
- ….

A few things that have driven modern machine learning

Representation: With better and better understanding of the underlining statistics about the data and methods.

Evaluation: The ideal strategy is always to aim at your target directly (take non-stop flight as opposed to having multiple stops).

Optimization: Based on the chosen representation and evaluation, you pick a strategy (mathematical/statistical) to achieve your goal.

Data: Having sufficient amount of data for learning and justification is increasingly important.

Computing power: In terms of both capacity and computation.

Some notations that we will be using

Set: a collection of distinct elements

$$\text{Color} = \{\text{white, red, blue, green}\}$$

$$\text{School} = \{\text{UCSD, UCLA, CMU, Caltech, Stanford}\}$$

Important to note: the order in a set doesn't matter, e.g.

$$\{\text{white, red, blue, green}\} = \{\text{red, white, green, blue}\}$$

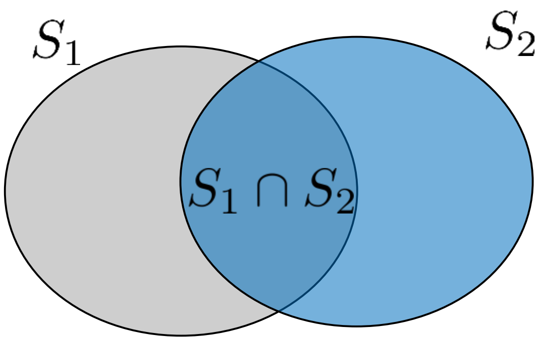
but the absence or presence of different elements does matter,

$$\{\text{white, red, blue, green}\} \neq \{\text{white, red, blue, green, black}\}$$

Mathematical operations for sets

1. Size (the number of total elements in a set): $|S|$

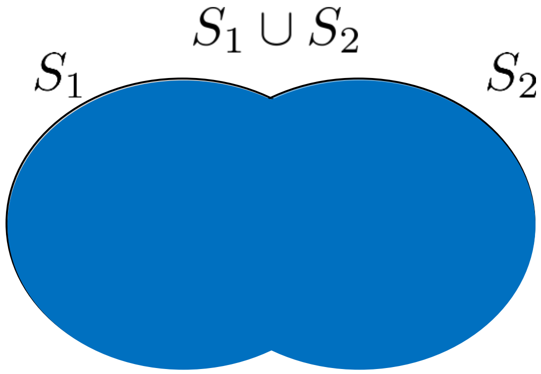
$$|\{\text{white, red, blue, green}\}| = 4$$



2. Intersection: $S_1 \cap S_2$

$$\{\text{white, red, blue, green}\} \cap \{\text{black, red, blue}\} = \{\text{red, blue}\}$$

3. Union: $S_1 \cup S_2$



$$\{\text{white, red, blue, green}\} \cup \{\text{black, red, blue}\} = \{\text{white, black, red, blue, green}\}$$

Mathematical operations for sets

More mathematical operations:

Complements [edit]

Main article: Complement (set theory)

Two sets can also be "subtracted". The *relative complement* of B in A (also called the *set-theoretic difference* of A and B), denoted by $A \setminus B$ (or $A - B$), is the set of all elements that are members of A but not members of B. It is valid to "subtract" members of a set that are not in the set, such as removing the element *green* from the set {1, 2, 3}; doing so has no effect.

In certain settings all sets under discussion are considered to be subsets of a given universal set U. In such cases, $U \setminus A$ is called the *absolute complement* or simply *complement* of A, and is denoted by A'.

- $A' = U \setminus A$

Examples:

- $\{1, 2\} \setminus \{1, 2\} = \emptyset$.
- $\{1, 2, 3, 4\} \setminus \{1, 3\} = \{2, 4\}$.
- If U is the set of integers, E is the set of even integers, and O is the set of odd integers, then $U \setminus E = E' = O$.

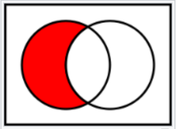
Some basic properties of complements:

- $A \setminus B \neq B \setminus A$ for $A \neq B$.
- $A \cup A' = U$.
- $A \cap A' = \emptyset$.
- $(A')' = A$.
- $\emptyset \setminus A = \emptyset$.
- $A \setminus \emptyset = A$.
- $A \setminus A = \emptyset$.
- $A \setminus U = \emptyset$.
- $A \setminus A' = A$ and $A' \setminus A = A'$.
- $U' = \emptyset$ and $\emptyset' = U$.
- $A \setminus B = A \cap B'$.
- if $A \subseteq B$ then $A \setminus B = \emptyset$.

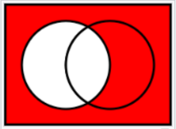
An extension of the complement is the symmetric difference, defined for sets A, B as

$$A \,\Delta\, B = (A \setminus B) \cup (B \setminus A).$$

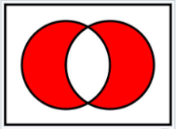
For example, the symmetric difference of {7, 8, 9, 10} and {9, 10, 11, 12} is the set {7, 8, 11, 12}. The power set of any set becomes a Boolean ring with symmetric difference as the addition of the ring (with the empty set as neutral element) and intersection as the multiplication of the ring.



The relative complement of B in A



The complement of A in U



The symmetric difference of A and B

Cartesian product [edit]

Main article: Cartesian product

A new set can be constructed by associating every element of one set with every element of another set. The *Cartesian product* of two sets A and B, denoted by $A \times B$ is the set of all ordered pairs (a, b) such that a is a member of A and b is a member of B.

Examples:

- {1, 2} × {red, white, green} = {(1, red), (1, white), (1, green), (2, red), (2, white), (2, green)}.
- {1, 2} × {1, 2} = {(1, 1), (1, 2), (2, 1), (2, 2)}.
- {a, b, c} × {d, e, f} = {(a, d), (a, e), (a, f), (b, d), (b, e), (b, f), (c, d), (c, e), (c, f)}.

Some basic properties of Cartesian products:

- $A \times \emptyset = \emptyset$.
- $A \times (B \cup C) = (A \times B) \cup (A \times C)$.
- $(A \cup B) \times C = (A \times C) \cup (B \times C)$.

Let A and B be finite sets; then the cardinality of the Cartesian product is the product of the cardinalities:

- $|A \times B| = |B \times A| = |A| \times |B|$.

https://en.wikipedia.org/wiki/Set_(mathematics)

We don’t need to deal with these operations explicitly in the class but it is important to have the basic understanding of them.

Vector

(probably the most important concept in this class)

Vector: a sequence of elements

(white, red, blue, green)

Important to note: the order DOES matter for vectors

(white, red, blue, green) $\neq$ (red, white, green, blue)

Sometimes, we also use: <white, red, blue, green>

In Python: [white, red, blue, green]

It is of critical importance to under the vector representation in machine learning!

Some notations that we will be using

Input data:

We use x (lower case) to denote a feature value (scalar).

The ith input data sample is represented as a vector using bold $\mathbf{x}$:

$$\mathbf{x}_i = (x_{i1}, ..., x_{im}) \in \mathbb{R}^m\text{: A row vector of } m \text{ elements.}$$

$$\mathbf{x}_i = (22, 1, 0, 160, 180)$$

The entire dataset is represented by a set (the sequence in which each data input $\mathbf{x}_i$ usually doesn't matter.

$$S = \{\mathbf{x}_i, i = 1..n\}\text{: A set } S \text{ with } n \text{ samples. } i \text{ goes from 1 to } n.$$