Selected Topics in Algorithms and Machine Learning

Lecture 1 (Part I): Introduction

Thomas Sauerwald

University of Cambridge, Department of Computer Science and Technology
email: thomas.sauerwald@cl.cam.ac.uk

July 2020

UNIVERSITY OF CAMBRIDGE


Welcome to this Program!!!
Cambridge
Wuhan
Sweden
Finland
Russia
United Kingdom
Denmark
Ireland
Poland
Belarus
Ukraine
Kazakhstan
Mongolia
Austria
France
Romania
Italy
Spain
Portugal
Greece
Turkey
Uzbekistan
Kyrgyzstan
Turkmenistan
China
Syria
Iraq
Iran
Afghanistan
Pakistan
Tunisia
Morocco
Algeria
Libya
Egypt
Nepal
Saudi Arabia
India
Myanmar (Burma)
Oman
Mali
Niger
Chad
Sudan
Yemen
Thailand
Gulf of Aden
Arabian Sea
Bay of Bengal
Vietnam
Philippines
Burkina Faso
Nigeria
Ethiopia
Gulf of Thailand

Outline

Short Bio

Recent Developments in AI/ML

Introduction to this Course

Warm-Up: Two Very Simple Machine Learning Problems

Brief Bio

- Diploma in Mathematics 2005, University of Paderborn, Germany
- PhD in Computer Science 2008, University of Paderborn, Germany
- Postdoc-Positions at Berkeley, Vancouver and Max Planck Institute for Informatics
- Reader at the Department of Computer Science and Technology, Cambridge (joined 2013)
- 2015 Visiting Researcher at Microsoft Research Cambridge
- ERC Starting Grant 2015 on Concurrent Markov Chains and Randomised Algorithms
- since 2017 Director of Studies in Cambridge at Emmanuel College

Brief Bio

- Diploma in Mathematics 2005, University of Paderborn, Germany
- PhD in Computer Science 2008, University of Paderborn, Germany
- Postdoc-Positions at Berkeley, Vancouver and Max Planck Institute for Informatics
- Reader at the Department of Computer Science and Technology, Cambridge (joined 2013)
- 2015 Visiting Researcher at Microsoft Research Cambridge
- ERC Starting Grant 2015 on Concurrent Markov Chains and Randomised Algorithms
- since 2017 Director of Studies in Cambridge at Emmanuel College

Teaching:

- Advanced Algorithms (3rd year undergraduate course)
- Probability & Computation (3rd year undergraduate course)
- Machine Learning (graduate course, till 2018/19)
- Algorithms (1st year undergraduate course, till 2016/17)
- Introduction to Probability (1st year undergraduate course)

Outline

Short Bio

Recent Developments in AI/ML

Introduction to this Course

Warm-Up: Two Very Simple Machine Learning Problems

Garry Kasparov faced off against Deep Blue, IBM's chess-playing computer in 1997. Deep Blue was able to imagine an average of 200,000,000 positions per second. Kasparov ended up losing the match. (AP Photo/Adam Nadel)

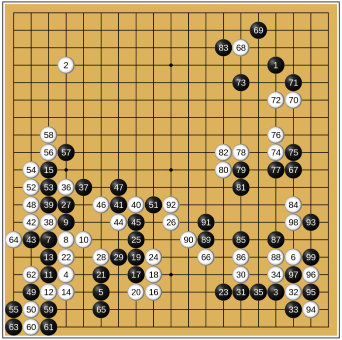
Adam Nadel/AP

1996: Kasparov-Deep Blue 4:2
1997: Kasparov-Deep Blue 2,5:3,5

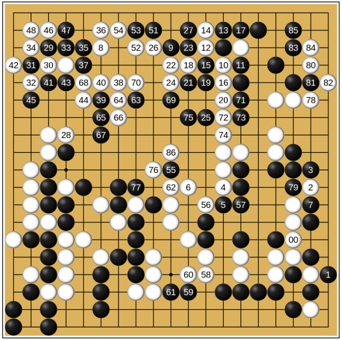
AlphaGo

Example game [edit]

AlphaGo Master (white) v. Tang Weixing (31 December 2016), AlphaGo won by resignation. White 36 was widely praised.



First 99 moves



Moves 100–186 (149 at 131, 150 at 130)

Source: Wikipedia

Source: Google

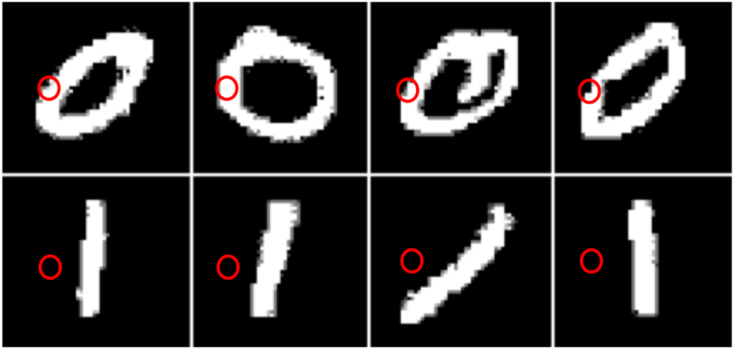


Figure 5: Example of frontal upright face images used for training.

Source: Japan Times

UNITED NATIONS
DEPARTMENT OF ECONOMIC AND SOCIAL AFFAIRS

Home | About UN DESA | Key Issues | News | UN DESA Voice | Multimedia Library | Calendar | Contact Us

Home » News » Will robots and AI cause mass unemployment? Not necessarily, but they do bring other threats

News

Will robots and AI cause mass unemployment? Not necessarily, but they do bring other threats

13 September 2017, New York

The Economist explains

Why Uber's self-driving car killed a pedestrian

It was the first fatal accident of its kind

The Economist explains ›
May 29th 2018 | by T.S.

Source: The Economist

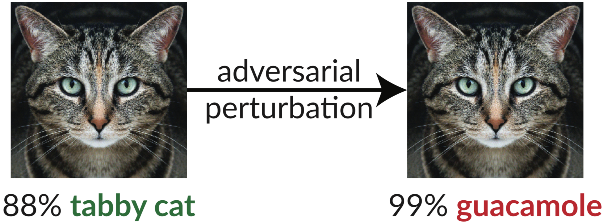


Fig. 1. A small change imperceptible to humans misleads the InceptionV3 network into classifying an image of a tabby cat as guacamole. Image taken from https://github.com/anishathalye/obfuscated-gradients.

A Simple Explanation for the Existence of Adversarial Examples with Small Hamming Distance

Adi Shamir[1], Itay Safran[1], Eyal Ronen[2], and Orr Dunkelman[3]

[1] Computer Science Department, The Weizmann Institute, Rehovot, Israel
[2] Computer Science Department, Tel Aviv University, Tel Aviv, Israel
[3] Computer Science Department, University of Haifa, Israel

Abstract. The existence of adversarial examples in which an imperceptible change in the input can fool well trained neural networks was experimentally discovered by Szegedy et al in 2013, who called them "Intriguing properties of neural networks". Since then, this topic had become one of the hottest research areas within machine learning, but the ease with which we can switch between any two decisions in targeted attacks is still far from being understood, and in particular it is not clear which parameters determine the number of input coordinates we have to change in order to mislead the network. In this paper we develop a simple mathematical framework which enables us to think about this baffling phenomenon from a fresh perspective, turning it into a natural consequence of the geometry of $\mathbb{R}^n$ with the L_0 (Hamming) metric, which can be quantitatively analyzed. In particular, we explain why we should expect to find targeted adversarial examples with Hamming distance of roughly m in arbitrarily deep neural networks which are designed to distinguish between m input classes.

BBC | Sign in | News | Sport | Weather | Shop | Earth | Travel

NEWS

Home | Video | World | UK | Business | Tech | Science | Stories | Entertainment & Arts

Technology

Stephen Hawking warns artificial intelligence could end mankind

By Rory Cellan-Jones
Technology correspondent

2 December 2014 | Share

Stephen Hawking: "Humans, who are limited by slow biological evolution, couldn't compete and would be superseded"

Prof Stephen Hawking, one of Britain's pre-eminent scientists, has said that efforts to create thinking machines pose a threat to our very existence.

Source: BBC News

Outline

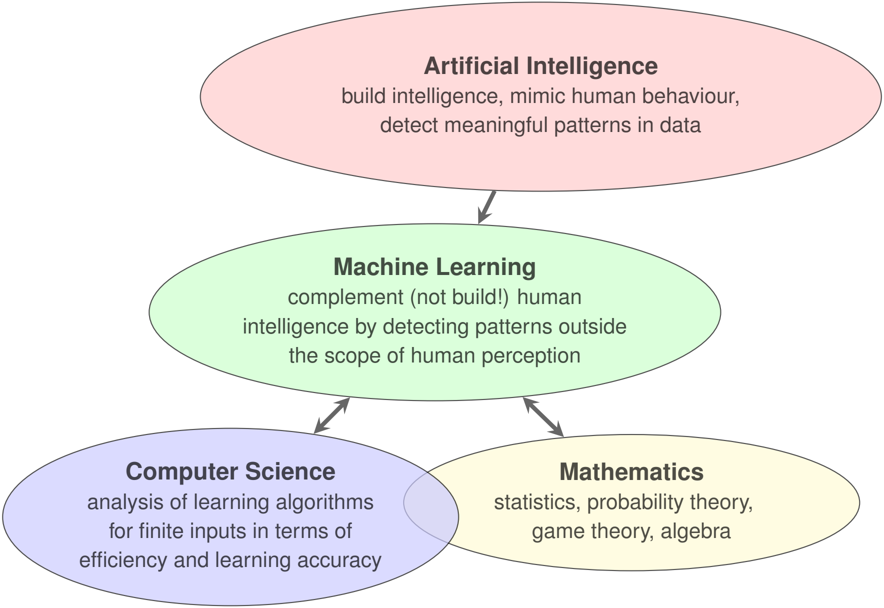
Short Bio

Recent Developments in AI/ML

Introduction to this Course

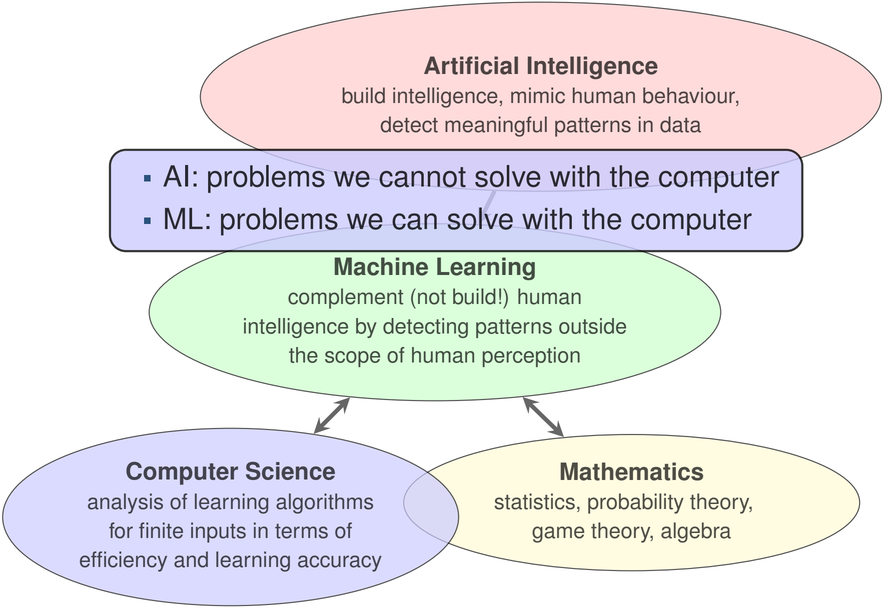
Warm-Up: Two Very Simple Machine Learning Problems

Machine Learning and Artificial Intelligence



Machine Learning and Artificial Intelligence

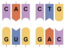
- AI: problems we cannot solve with the computer
- ML: problems we can solve with the computer



Motivation

Data is everywhere:

- we generate it continuously
- big data is freely available
- storage is cheap



Machine Learning is everywhere:

- text recognition
- medicine and healthcare
- spam filtering
- self-driving cars
- recommender system
- photo search

Motivation

Data is everywhere:

- we generate it continuously
- big data is freely available
- storage is cheap

Machine Learning is everywhere:

- text recognition
- medicine and healthcare
- spam filtering
- self-driving cars
- recommender system
- photo search

We have access to huge computational power.
We have new sophisticated (and parallel!) machine learning algorithms.

Motivation

Data is everywhere:

- we generate it continuously
- big data is freely available
- storage is cheap

Machine Learning is everywhere:

- text recognition
- medicine and healthcare
- spam filtering
- self-driving cars
- recommender system
- photo search

We have access to huge computational power.
We have new sophisticated (and parallel!) machine learning algorithms.

Represent large amounts of data so that learning algorithms can:

- spot patterns and understand better behaviour of entity (**descriptive** learning);

Motivation

Data is everywhere:

- we generate it continuously
- big data is freely available
- storage is cheap

Machine Learning is everywhere:

- text recognition
- medicine and healthcare
- spam filtering
- self-driving cars
- recommender system
- photo search

Discover hidden trends in data, Extract clusters in a network, Segment image into pieces

ver.

We ning algorithms.

Represent large amounts of data so that learning algorithms can:

- spot patterns and understand better behaviour of entity (**descriptive** learning);

Motivation

Data is everywhere:

- we generate it continuously
- big data is freely available
- storage is cheap

Machine Learning is everywhere:

- text recognition
- medicine and healthcare
- spam filtering
- self-driving cars
- recommender system
- photo search

Discover hidden trends in data, Extract clusters in a network, Segment image into pieces

ver.

We hing algorithms.

Represent large amounts of data so that learning algorithms can:

- spot patterns and understand better behaviour of entity (**descriptive** learning);
- learn behaviour of entity to predict future behaviour (or unknown feature) (**predictive** learning).

Motivation

Data is everywhere:

- we generate it continuously
- big data is freely available
- storage is cheap

Machine Learning is everywhere:

- text recognition
- medicine and healthcare
- spam filtering
- self-driving cars
- recommender system
- photo search

Discover hidden trends in data, Extract clusters in a network, Segment image into pieces

ver.

We ning algorithms.

Represent large amounts of data so that learning algorithms can:

- spot patterns and understand better behaviour of entity (**descriptive** learning);
- learn behaviour of entity to predict future behaviour (or unknown feature) (**predictive** learning).

Patient X does not have diabetes, this image is a face, the weather tomorrow is sunny, the expected salary of person Y is 40K, ...

Landscape of Machine Learning Algorithms

Training Set
provided initially

Supervised Learning
Classification, regression: logistic regr., SVM, decision tree, neural network, naive Bayes, Perceptron, kNN, Boosting

Predict
unseen data

No Training Set

Unsupervised Learning
Clustering: spectral, hierarchical, k-means. Principal Component Analysis, Singular value Decomposition, Dimensionality Reduction, Density Estimation

Extract
Knowledge

Landscape of Machine Learning Algorithms

Training Set provided initially

Feedback after Decisions

No Training Set

Supervised Learning

Classification, regression: logistic regr., SVM, decision tree, neural network, naive Bayes, Perceptron, kNN, Boosting

Online/Reinforcement Learning

Weighted-Majority, Multiplicative-Update. control learning: Markov Decision Processes, temporal difference

Unsupervised Learning

Clustering: spectral, hierarchical, k-means. Principal Component Analysis, Singular value Decomposition, Dimensionality Reduction, Density Estimation

Predict unseen data

Maximise Reward

Extract Knowledge

Workshop Overview (Lectures and Presentation)

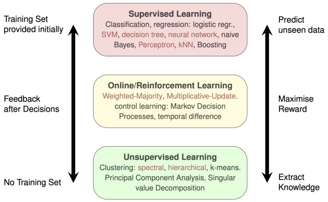
Intro / Supervised Learning Nearest Neighbour	Lecture 1	Dr Thomas Sauerwald
Reinforcement Learning Online Learning	Lecture 2	Dr Thomas Sauerwald
Supervised Learning Decision Trees & Random Forests	Lecture 3	Dr Thomas Sauerwald
Supervised Learning Perceptron and SVMs	Lecture 4	Dr Thomas Sauerwald
Unsupervised Learning Graph Clustering	Lecture 5	Dr Thomas Sauerwald
Supervised Learning Recommendation Systems	Lecture 6	Dr Thomas Sauerwald
Supervised Learning Neural Networks	Lecture 7	Dr Thomas Sauerwald
Linear Programming Travelling Salesman Problem	Lecture 8	Dr Thomas Sauerwald
Final Project Presentation and Q & A	Lecture 9	Dr Sven-Ake Wegner Dr Thomas Sauerwald

Lecture 1

1a. **Introduction**

Course Introduction and Overview

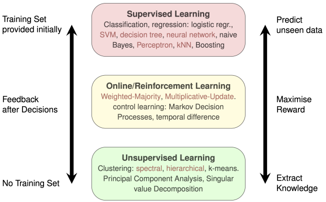
- Recent Developments in Machine Learning
- Landscape of ML vs. AI
- Course Overview



1a. **Introduction**

Course Introduction and Overview

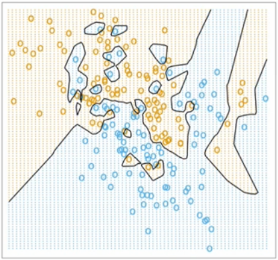
- Recent Developments in Machine Learning
- Landscape of ML vs. AI
- Course Overview



1b. **Nearest Neighbour Algorithm**

Our First Simple ML Tool

- Nearest Neighbour Algorithm
- Distance Functions and Number of Neighbours
- Curse of Dimensionality
- Dimensionality Reduction using Random Projection



Lectures 2 and 3

2. **Online Learning**
   Making Predictions based on Experts Advice

   - Online Learning using Experts
   - Weighted Majority Algorithm
   - Randomised Weighted Majority Algorithm
   - Why Randomisation?

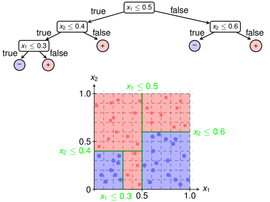
Lectures 2 and 3

2. **Online Learning**
   Making Predictions based on Experts Advice

   - Online Learning using Experts
   - Weighted Majority Algorithm
   - Randomised Weighted Majority Algorithm
   - Why Randomisation?

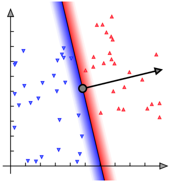
3. **Decision Trees and Random Forests**
   Predictions using Training Sets

   - Supervised Machine Learning Model
   - Decision Stumps (Most Basic Type)
   - ID3 algorithm
   - Regression Trees
   - Test on Data from Financial Markets



4. **Perceptron and Support Vector Machines**
   Classifying Geometric Data

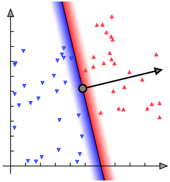
   - Perceptron Algorithm
   - Analysis and Application to Spam Classification
   - Introduction to Support Vector Machines
   - Hard-SVM, Soft-SVM, Kernel-Trick
   - Comparison between Decision Trees and SVMs



Lectures 4 and 5

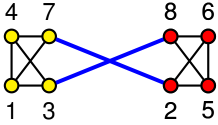
4. **Perceptron and Support Vector Machines**
   Classifying Geometric Data

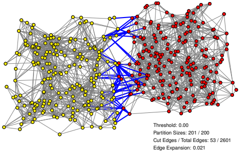
   - Perceptron Algorithm
   - Analysis and Application to Spam Classification
   - Introduction to Support Vector Machines
   - Hard-SVM, Soft-SVM, Kernel-Trick
   - Comparison between Decision Trees and SVMs



5. **Spectral Graph Theory and Clustering**
   Spectral Analysis of Big Data

   - Introduction to Graph Theory
   - Adjacency Matrix, Eigenvalues and Eigenvectors
   - Visualising Networks
   - Spectral Clustering Algorithm
   - Examples





6. **Recommendation Systems**
   Recommending Products or Movie Ratings
   - Intro to Recommender Systems, Netflix challenge
   - Content-Based Recommendation
   - Collaborative Filtering (User-to-User, Item-to-Item)
   - Outlook: Singular Value Decomposition

users

movies	1	2	3	4	5	6
1	1		3		??	5
2			5	4		
3	2	4		1	2	
4		2	4		5	
5			4	3	4	2
6	1		3		3	

$$x_{5,1} = \frac{0.41\cdot 2+0.59\cdot 3}{0.41+0.59}$$

Lectures 6 and 7

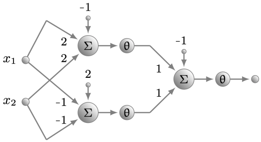
6. **Recommendation Systems**
   Recommending Products or Movie Ratings
   - Intro to Recommender Systems, Netflix challenge
   - Content-Based Recommendation
   - Collaborative Filtering (User-to-User, Item-to-Item)
   - Outlook: Singular Value Decomposition

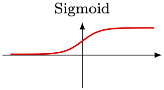
users

movies	1	2	3	4	5	6
1	1		3		??	5
2			5	4		
3	2	4		1	2	
4		2	4		5	
5			4	3	4	2
6	1		3		3	

$$x_{5,1} = \frac{0.41\cdot 2 + 0.59\cdot 3}{0.41 + 0.59}$$

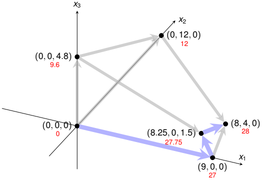
7. **Neural Networks**
   Deep Learning
   - Introduction and Connection with Human Thinking
   - Components of Neural Networks
   - Computation with Neural Networks
   - Implementation Aspects and Test





8 **Linear Programming**
Solving Combinatorial Optimisation Problems

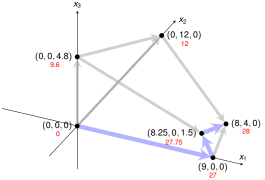
- Definition and Formulation of Linear Programming

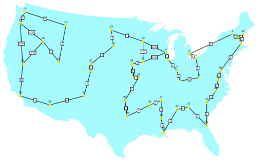


8 **Linear Programming**

Solving Combinatorial Optimisation Problems

- Definition and Formulation of Linear Programming
- Introduction to the Travelling Salesman Problem
- Solving a classical TSP instance from 1950





Final Project (15 Sep 2020)

9 Project Presentation

- pick your favourite ML topic ☺
- conduct a mini-project focusing on fundamental ideas (e.g., literature survey comparing different ML tools) or practical aspects (implement some algorithms and evaluate them)
- prepare 10-min presentation for the Final Session
- there will be Q&A session and you will receive detailed feedback later
- optional (and after the program): write a short technical report summarising your findings and submit to lecturer

Tutorials

Brief Bio of Sven-Ake Wegner:

- Diploma in Mathematics 2007, University of Paderborn, Germany
- PhD in Mathematics 2010, Polytechnic University of Valencia, Spain
- Postdoc-Positions at Sobolev Institute (Russia), Adam Mickiewicz University (Poland), University of Wuppertal (Germany)
- Senior Lecturer at Teesside University (UK) since 2018

Tutorials:

- 4 tutorials given by Dr Sven Ake Wegner, 4 tutorials given by myself
- tutorials will be based on practical (and sometimes also theoretical) aspects of the material lectured
- we will be using Google Colab `colab.research.google.com` and Python
- First tutorial on Saturday 25 July
  (topic: Nearest Neighbour Classification and Dimensionality Reduction)

Learning Goals

- appreciate the power of algebraic and randomised techniques to solve a variety of problems arising in the context of supervised and reinforcement learning

Learning Goals

- appreciate the power of algebraic and randomised techniques to solve a variety of problems arising in the context of supervised and reinforcement learning
- understand the basic ideas and intuitions behind spectral graph theory and its applications in unsupervised learning (visualising and clustering)

Learning Goals

- appreciate the power of algebraic and randomised techniques to solve a variety of problems arising in the context of supervised and reinforcement learning
- understand the basic ideas and intuitions behind spectral graph theory and its applications in unsupervised learning (visualising and clustering)
- understand how to formulate and solve linear and integer programs using SIMPLEX and Branch & Bound

Learning Goals

- appreciate the power of algebraic and randomised techniques to solve a variety of problems arising in the context of supervised and reinforcement learning
- understand the basic ideas and intuitions behind spectral graph theory and its applications in unsupervised learning (visualising and clustering)
- understand how to formulate and solve linear and integer programs using SIMPLEX and Branch & Bound
- get an idea on how these concepts are taught at Cambridge

Learning Goals

- appreciate the power of algebraic and randomised techniques to solve a variety of problems arising in the context of supervised and reinforcement learning
- understand the basic ideas and intuitions behind spectral graph theory and its applications in unsupervised learning (visualising and clustering)
- understand how to formulate and solve linear and integer programs using SIMPLEX and Branch & Bound
- get an idea on how these concepts are taught at Cambridge
- implement and test machine learning tools using Python

Learning Goals

- appreciate the power of algebraic and randomised techniques to solve a variety of problems arising in the context of supervised and reinforcement learning
- understand the basic ideas and intuitions behind spectral graph theory and its applications in unsupervised learning (visualising and clustering)
- understand how to formulate and solve linear and integer programs using SIMPLEX and Branch & Bound
- get an idea on how these concepts are taught at Cambridge
- implement and test machine learning tools using Python
- develop your own project on your favourite Machine Learning Topic, Project Presentation 15 September 2020

Learning Goals

- appreciate the power of algebraic and randomised techniques to solve a variety of problems arising in the context of supervised and reinforcement learning
- understand the basic ideas and intuitions behind spectral graph theory and its applications in unsupervised learning (visualising and clustering)
- understand how to formulate and solve linear and integer programs using SIMPLEX and Branch & Bound
- get an idea on how these concepts are taught at Cambridge
- implement and test machine learning tools using Python
- develop your own project on your favourite Machine Learning Topic, Project Presentation 15 September 2020
- . . . and have fun (☺)

Mathematical Tools

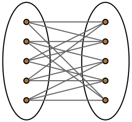
- **Matrices and Geometry**
  - Data points (predictions, observations, classifications) encoded in matrices/vectors
  - This allows geometric representation that is the basis of many ML methods
  - Networks and graphs $\Leftrightarrow$ adjacency matrices

Inner product, Hyperplanes, Eigenvectors

Mathematical Tools

- **Matrices and Geometry**
  - Data points (predictions, observations, classifications) encoded in matrices/vectors
  - This allows geometric representation that is the basis of many ML methods
  - Networks and graphs ⇔ adjacency matrices

Inner product, Hyperplanes, Eigenvectors

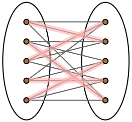


$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Mathematical Tools

- **Matrices and Geometry**
  - Data points (predictions, observations, classifications) encoded in matrices/vectors
  - This allows geometric representation that is the basis of many ML methods
  - Networks and graphs $\Leftrightarrow$ adjacency matrices

Inner product, Hyperplanes, Eigenvectors

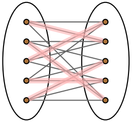


$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Mathematical Tools

- **Matrices and Geometry**
  - Data points (predictions, observations, classifications) encoded in matrices/vectors
  - This allows geometric representation that is the basis of many ML methods
  - Networks and graphs $\Leftrightarrow$ adjacency matrices

  Inner product, Hyperplanes, Eigenvectors



$$A = \begin{pmatrix} 0&0&0&0&0&1&1&0&0&1\\ 0&0&0&0&0&0&1&1&0&1\\ 0&0&0&0&0&1&1&0&1&0\\ 0&0&0&0&0&1&0&1&1&0\\ 0&0&0&0&0&0&0&1&1&1\\ 1&0&1&1&0&0&0&0&0&0\\ 1&1&1&0&0&0&0&0&0&0\\ 0&1&0&1&1&0&0&0&0&0\\ 0&0&1&1&1&0&0&0&0&0\\ 1&1&0&0&1&0&0&0&0&0 \end{pmatrix}$$

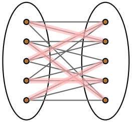
- **Probability Theory**
  - Sampling-based algorithms allow us to process massive data sets and ensure robustness against tailored worst-case instances
  - Many ML methods exploit concentration of measure
  - Learning Model: training data set represented by a probability distribution

  Random Variables, Chernoff Bounds, Hashing

Mathematical Tools

- **Matrices and Geometry**
  - Data points (predictions, observations, classifications) encoded in matrices/vectors
  - This allows geometric representation that is the basis of many ML methods
  - Networks and graphs ⇔ adjacency matrices

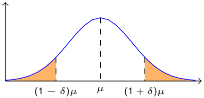
  Inner product, Hyperplanes, Eigenvectors



$$A = \begin{pmatrix} 0&0&0&0&0&1&1&0&0&1 \\ 0&0&0&0&0&0&1&1&0&1 \\ 0&0&0&0&0&1&1&0&1&0 \\ 0&0&0&0&0&1&0&1&1&0 \\ 0&0&0&0&0&0&0&1&1&1 \\ 1&0&1&1&0&0&0&0&0&0 \\ 1&1&1&0&0&0&0&0&0&0 \\ 0&1&0&1&1&0&0&0&0&0 \\ 0&0&1&1&1&0&0&0&0&0 \\ 1&1&0&0&1&0&0&0&0&0 \end{pmatrix}$$

- **Probability Theory**
  - Sampling-based algorithms allow us to process massive data sets and ensure robustness against tailored worst-case instances
  - Many ML methods exploit concentration of measure
  - Learning Model: training data set represented by a probability distribution

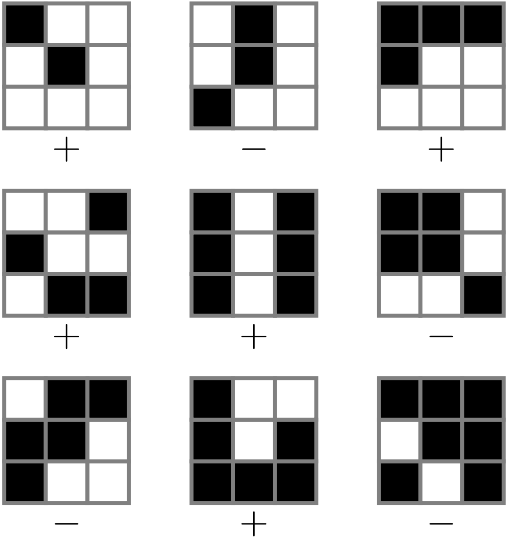
  Random Variables, Chernoff Bounds, Hashing



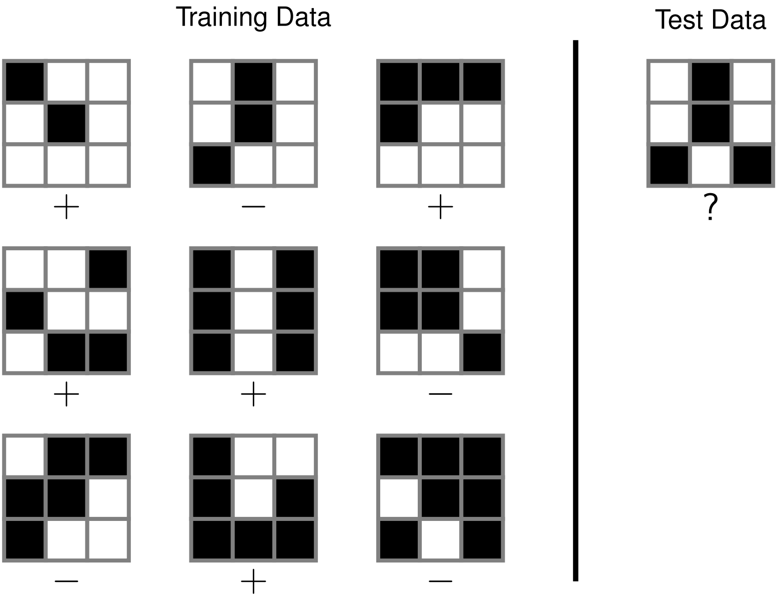
Outline

Warm-up: A First Learning Problem

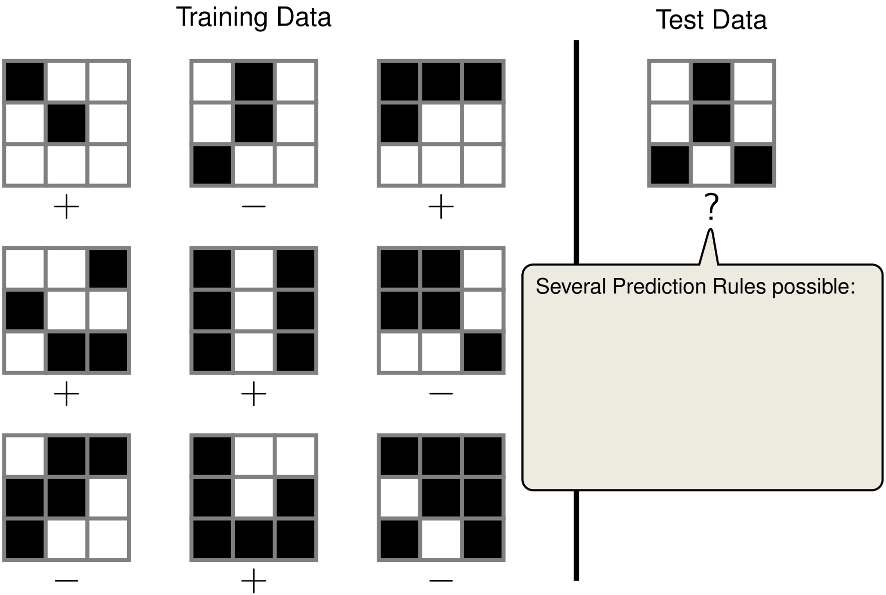
Training Data



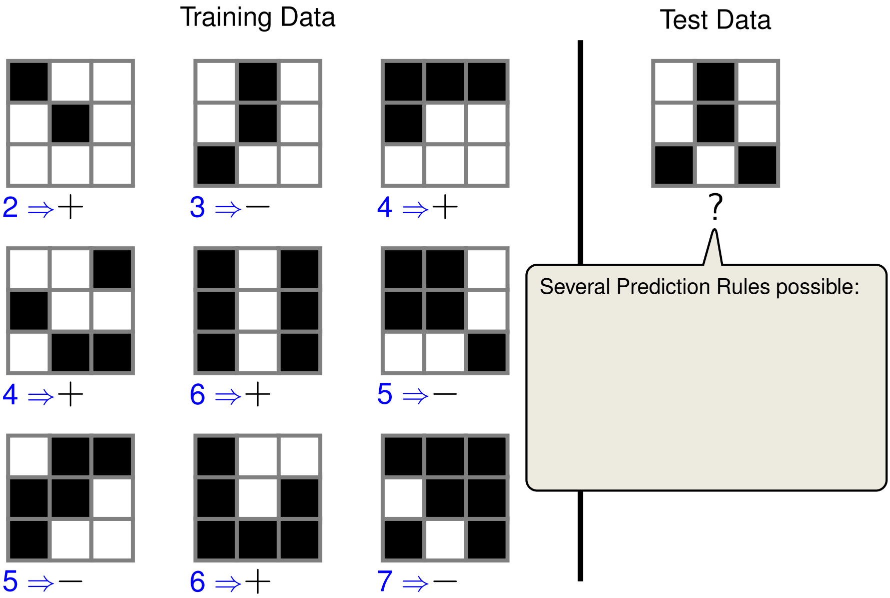
Warm-up: A First Learning Problem



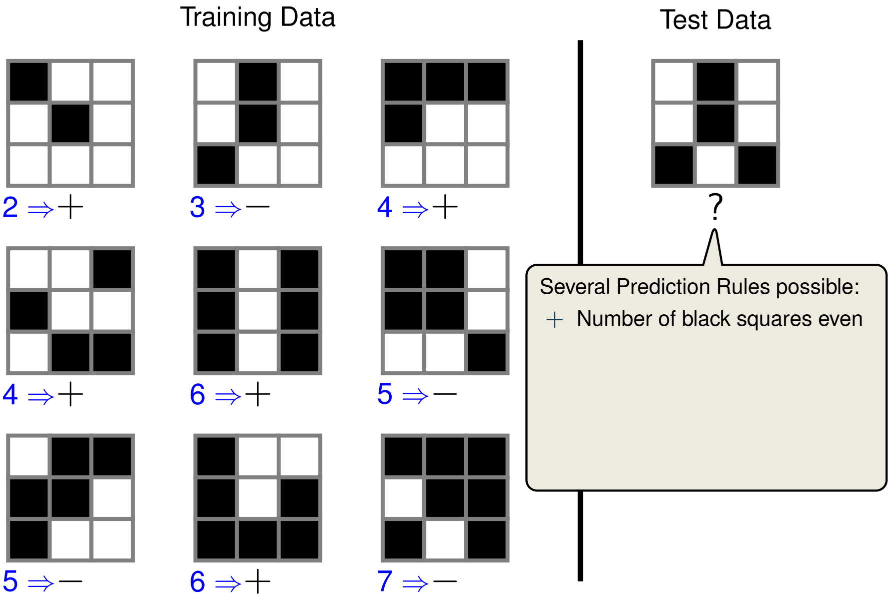
Warm-up: A First Learning Problem



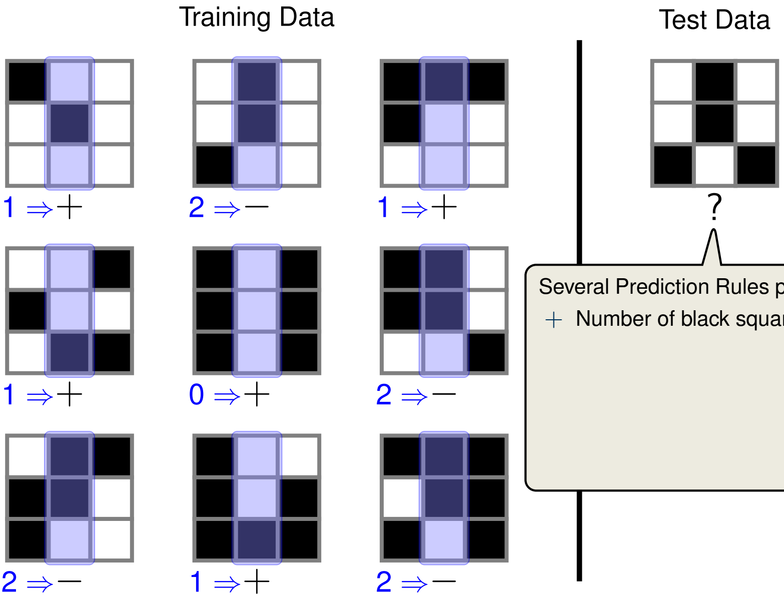
Warm-up: A First Learning Problem



Warm-up: A First Learning Problem



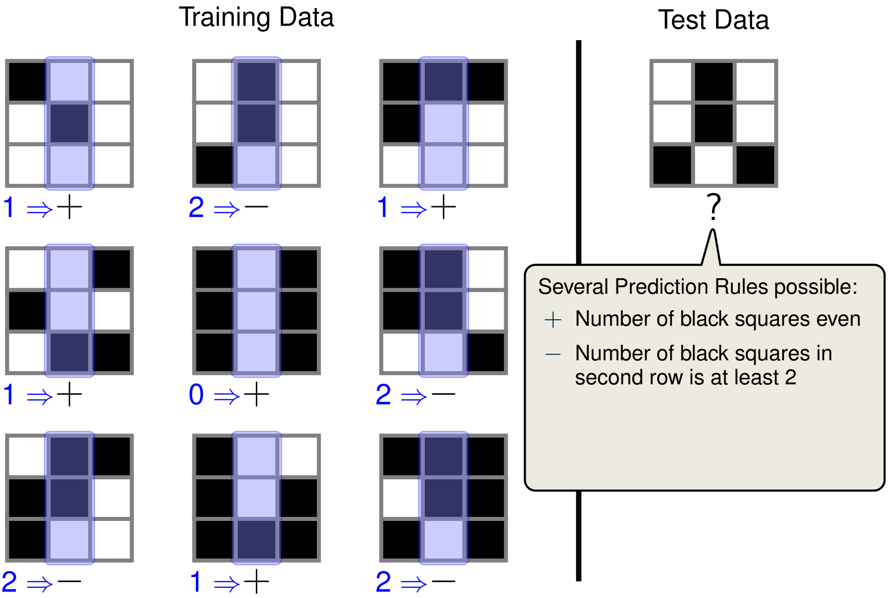
Warm-up: A First Learning Problem



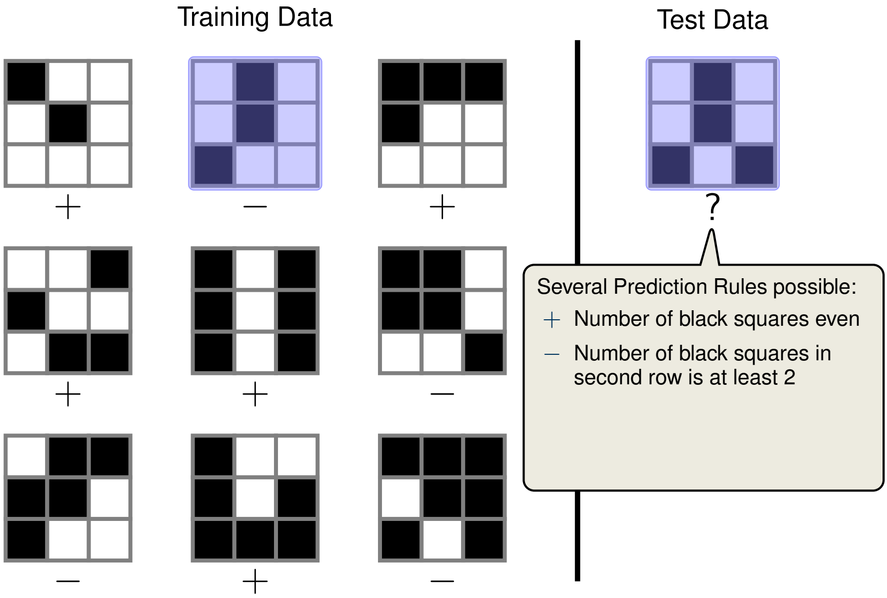
Several Prediction Rules possible:

+ Number of black squares even

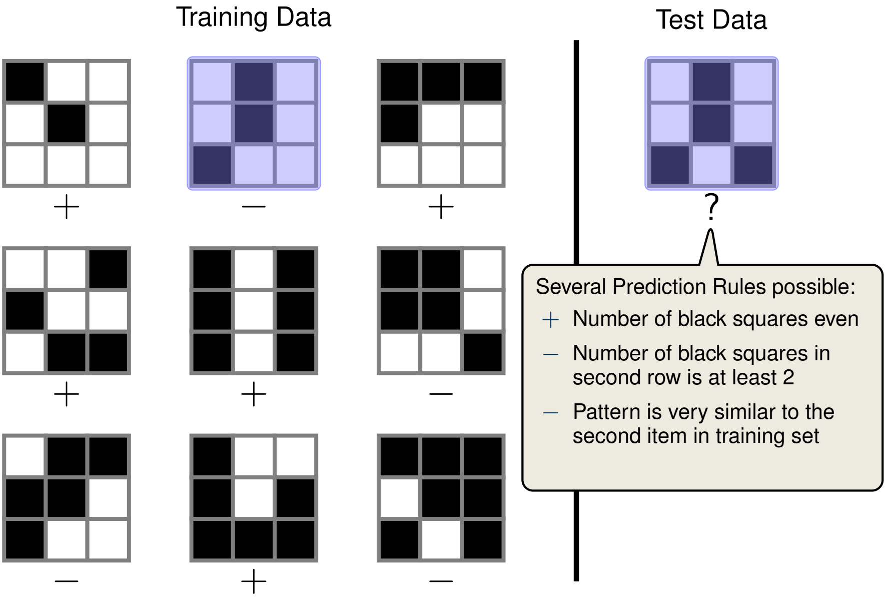
Warm-up: A First Learning Problem



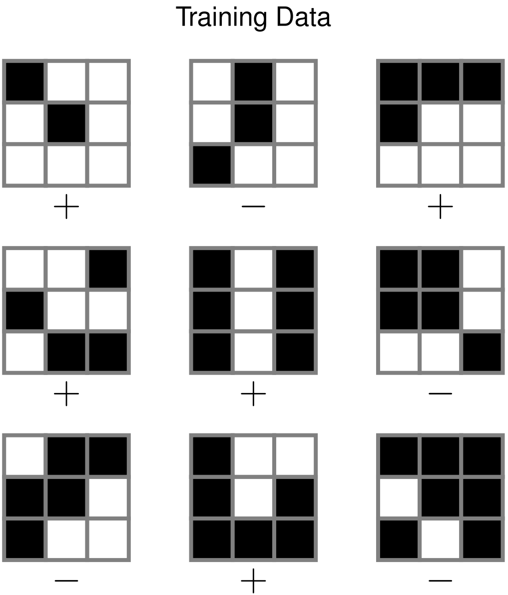
Warm-up: A First Learning Problem

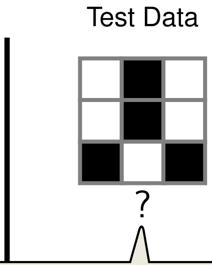


Warm-up: A First Learning Problem



Warm-up: A First Learning Problem





Several Prediction Rules possible:

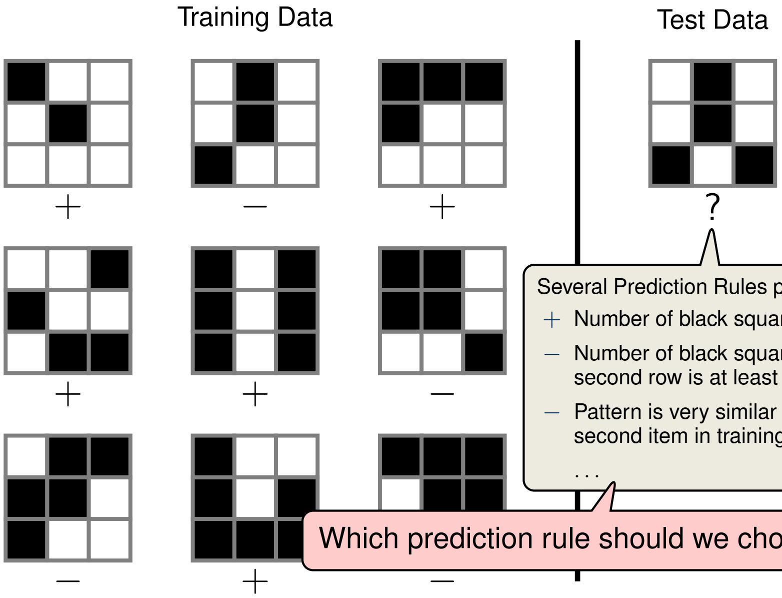
+ Number of black squares even

− Number of black squares in second row is at least 2

− Pattern is very similar to the second item in training set

. . .

Warm-up: A First Learning Problem



Several Prediction Rules possible:

- $+$ Number of black squares even
- $-$ Number of black squares in second row is at least 2
- $-$ Pattern is very similar to the second item in training set
- ...

Which prediction rule should we choose?

Warm-up: A Second Learning Problem (1/2)

example	**label**
train	
197	$+$
128	$-$
30	$-$
72	$-$
133	$-$
109	$+$
213	$+$
84	$+$
3	$-$

(example taken from Robert Schapire)

Warm-up: A Second Learning Problem (1/2)

example	label
train	
197	$+$
128	$-$
30	$-$
72	$-$
133	$-$
109	$+$
213	$+$
84	$+$
3	$-$
test	
200	?
68	?

(example taken from Robert Schapire)

Warm-up: A Second Learning Problem (1/2)

example	label
train	
197	$+$
128	$-$
30	$-$
72	$-$
133	$-$
109	$+$
213	$+$
84	$+$
3	$-$
test	
200	?
68	?
111	???

(example taken from Robert Schapire)

Warm-up: A Second Learning Problem (2/2)

example										**label**
				train						
197	=	1	1	0	0	0	1	0	1	$+$
128	=	1	0	0	0	0	0	0	0	$-$
30	=	0	0	0	1	1	1	1	0	$-$
72	=	0	1	0	0	1	0	0	0	$-$
133	=	1	0	0	0	0	1	0	1	$-$
109	=	0	1	1	0	1	1	0	1	$+$
213	=	1	1	0	1	0	1	0	1	$+$
84	=	0	1	0	1	0	1	0	0	$+$
3	=	0	0	0	0	0	1	1	1	$-$
				test						
200	=	1	1	0	0	1	0	0	0	$+$(?)
68	=	0	1	0	0	0	1	0	0	$-$(?)
111	=	0	1	1	0	1	1	1	1	??

(example taken from Robert Schapire)

Warm-up: A Second Learning Problem (2/2)

example										label
				train						
197	$=$	1	1	0	0	0	1	0	1	$+$
128	$=$	1	0	0	0	0	0	0	0	$-$
30	$=$	0	0	0	1	1	1	1	0	$-$
72	$=$	0	1	0	0	1	0	0	0	$-$
133	$=$	1	0	0	0	0	1	0	1	$-$
109	$=$	0	1	1	0	1	1	0	1	$+$
213	$=$	1	1	0	1	0	1	0	1	$+$
84	$=$	0	1	0	1	0	1	0	0	$+$
3	$=$	0	0	0	0	0	1	1	1	$-$
				test						
200	$=$	1	1	0	0	1	0	0	0	$+$(?)
68	$=$	0	1	0	0	0	1	0	0	$-$(?)
111	$=$	0	1	1	0	1	1	1	1	??

(example taken from Robert Schapire)

Warm-up: A Second Learning Problem (2/2)

example										label
train										
197	=	1	1	0	0	0	1	0	1	+
128	=	1	0	0	0	0	0	0	0	−
30	=	0	0	0	1	1	1	1	0	−
72	=	0	1	0	0	1	0	0	0	−
133	=	1	0	0	0	0	1	0	1	−
109	=	0	1	1	0	1	1	0	1	+
213	=	1	1	0	1	0	1	0	1	+
84	=	0	1	0	1	0	1	0	0	+
3	=	0	0	0	0	0	1	1	1	−
test										
200	=	1	1	0	0	1	0	0	0	+(?)
68	=	0	1	0	0	0	1	0	0	−(?)
111	=	0	1	1	0	1	1	1	1	??

(example taken from Robert Schapire)

Warm-up: A Second Learning Problem (2/2)

example										**label**
				train						
197	$=$	1	1	0	0	0	1	0	1	$+$
128	$=$	1	0	0	0	0	0	0	0	$-$
30	$=$	0	0	0	1	1	1	1	0	$-$
72	$=$	0	1	0	0	1	0	0	0	$-$
133	$=$	1	0	0	0	0	1	0	1	$-$
109	$=$	0	1	1	0	1	1	0	1	$+$
213	$=$	1	1	0	1	0	1	0	1	$+$
84	$=$	0	1	0	1	0	1	0	0	$+$
3	$=$	0	0	0	0	0	1	1	1	$-$
				test						
200	$=$	1	1	0	0	1	0	0	0	$-$
68	$=$	0	1	0	0	0	1	0	0	$+$
111	$=$	0	1	1	0	1	1	1	1	$+$

(example taken from Robert Schapire)

Conclusions from these Examples

- More Data Points is usually good: training set should be sufficiently large (but leave part of as test set for evaluation!)
- Feature selection: use information about data set
  $\rightsquigarrow$ changing dimensions is a common ML technique
- Simplicity of prediction rule: among competing hypotheses chose the one which makes the smallest number of assumptions (Occam's razor)
  $\rightsquigarrow$ prevents overfitting