Big Data

Bogdan Ichim

University of Bucharest
Faculty of Mathematics and Computer Science
Department of Computer Science

Bucharest, SS 2019

Organization Issues

1. Grading policy
2. Strongly recommended programming languages
3. Datasets
4. Motivation
5. What we do and what we do not do in this course

Grading policy

- Project using ML (choice to be discussed) with technical report (about 10 – 20 pages): 50%
- Final exam: 50%

Remark

Please, note that the projects will be discussed separately, in the Big Data laboratory.

Strongly recommended programming languages

The following programming languages are strongly recommended for the projects:

- R
- python

Remark

Please, note that other programming languages are allowed.

Datasets

You can get some inspiration (especially concerning the projects) by playing with the datasets listed here:

- UCI datasets
- CMU datasets
- MNIST database in Yann LeCun's page
- Tom Mitchell's ML course
- IMDb Datasets
- Protein Classification Benchmark Collection
- Wiki list of datasets for ML research
- Kaggle Datasets
- OpenML
- PMLB (supervised, accessible through Python API)
- R Datasets (accessible in R)

Motivation

- Exciting field providing practitioners with extremely successful industrial applications and researchers with challenging problems
- Big Data methods, especially the Machine Learning algorithms are using mathematical techniques from different areas and is intensively used both in hard and soft sciences
- We are interested in understanding foundations and discussing practical implications

The course is under development so any suggestions as for including or excluding specific topics or chapters are very welcome.

What we do in this course:

- This course is about the **applied** learning from data.
- We will look at ML and learning theory in general in a **mathematical and statistical** framework. We make an introduction into the common mathematical and statistical principles underlying ML techniques.
- Most of this course is devoted to **supervised** learning (better understood from a statistical learning theory point of view); the **unsupervised** problem is less developed in the literature may be addressed in some particular examples.
- We admit a large gap between theory and practice; theory relies on assumptions that may be too strong or too weak.
- In the laboratory sessions we implement a selected range of ML techniques.

What we do NOT do in this course:

- We do NOT discuss a particular Big Data platform. The companies are using different Big Data platforms, so any particular discussion is not really useful.
- We do NOT focus on the technical details of statistical learning. This is NOT another "programming language" course. There are **several** programming languages and platforms for solving Big Data problems in the real world.
- We do NOT address in detail questions from computational learning theory: for example efficiency of learning algorithms.

Structure of the Chapter I

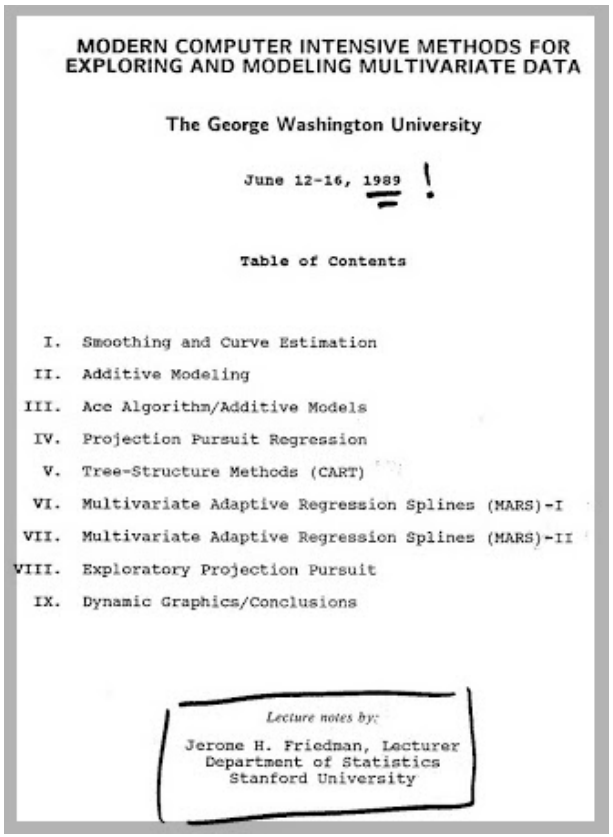
- What is Big Data?
- What is Machine Learning?
  - Related fields
  - Definition
  - Data E (Experience), associated types of learning
  - Tasks T and applications
  - Performance P measures
- Timeline of ML, recent advances and applications

What is NOT Big Data

- Big Data is NOT a so called "Big Data platform".
- Big Data is NOT a "programming language" or "library".
- Big Data is NOT a "cooking book" for algorithms.

High-dimensional challenges in mathematics and statistics

Below is a title page from 1989(!) of the course of Jerome Friedman on – "Big Data" and "machine learning".



MODERN COMPUTER INTENSIVE METHODS FOR EXPLORING AND MODELING MULTIVARIATE DATA

The George Washington University

June 12-16, 1989 !

Table of Contents

Lecture notes by:
Jerone H. Friedman, Lecturer
Department of Statistics
Stanford University

(Idea taken from the "No Hesitations" blog of Francis Diebold.)

High-dimensional challenges in mathematics and statistics

Please take a look at the following quotations of the Jerome Friedman's talk "Modern Statistics and Computer Revolution" in 1989(!).
Some of his amazing statements:

"What do we want data to tell us? I think this is far less focused now than in the past because of automated data collection.... When data are marginally cheap, you tend to collect massive amounts of data on the off chance that it might be useful someday."

"It is a fact of life that nearly all statistical methods that are in common use today were developed before about 1950."

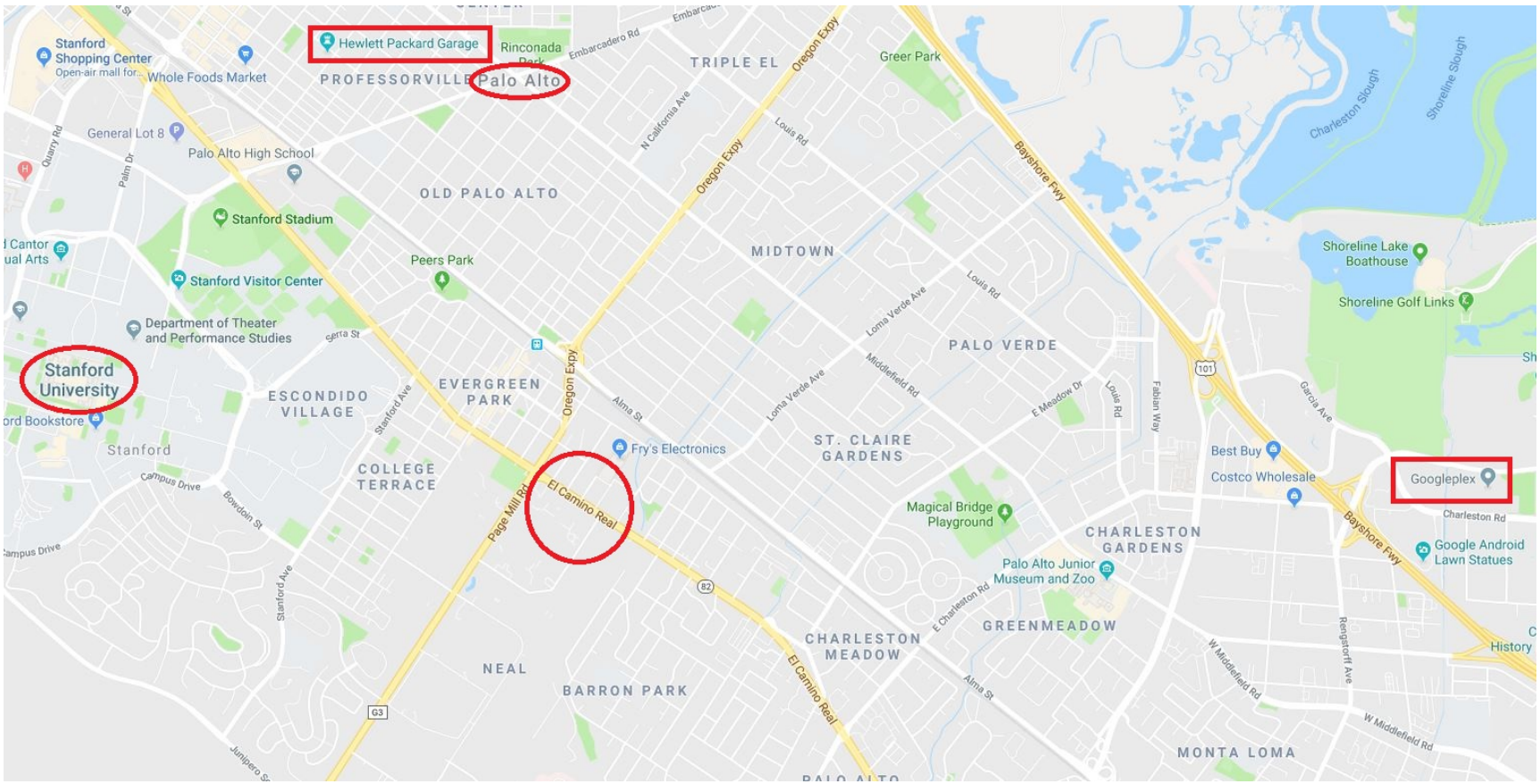
"Because of the tremendous computational power that computers provide, we can now inexpensively calculate very complex estimators. We don't require solutions in closed form."

What is Stanford University?

Leland Stanford Junior University (often referred to as Stanford University or simply Stanford) is an American private research university in California. Stanford University was founded in 1885 by Leland and Jane Stanford, dedicated to Leland Stanford Jr, their only child. The institution opened in 1891 on Stanford's previous Palo Alto farm.

The academic central campus is adjacent to Palo Alto, bounded by El Camino Real, Stanford Avenue, Junipero Serra Boulevard, and Sand Hill Road.

What is Stanford University?



What is Palo Alto?

Palo Alto is the heart of the Silicon Valley. The HP Garage is here and Palo Alto has become synonymous with innovation. A large number of well known companies were founded in Palo Alto.

The following list includes some of the largest, most profitable companies that were founded in Palo Alto.

- Google
- Facebook
- Hewlett-Packard
- Tesla

What is Big Data?

Clive Humby, UK Mathematician and architect of Tesco's Clubcard, 2006 (widely credited as the first to coin the phrase):

> *"Data is the new oil. It's valuable, but if unrefined it cannot really be used. It has to be changed into gas, plastic, chemicals, etc to create a valuable entity that drives profitable activity; so must data be broken down, analyzed for it to have value."*

Gary Wolf, Quantified Self Co-Founder, 2012 :

> *"People are saying, 'Big Data is the new oil.'"*

Virginia Rometty, IBM CEO, 2013:

> *"I want you to think about data as the next natural resource."*

What is Big Data?

Abhishek Mehta, CEO Tresata, 2013:

> *"Just like oil was a natural resource powering the last industrial revolution, data is going to be the natural resource for this industrial revolution. Data is the core asset, and the core lubricant, for not just the entire economic models built around every single industry vertical but also the socioeconomic models."*

Kevin Plank, founder and CEO of Under Armour, 2016:

> *"Data is the new oil. The companies that will win are using math."*

Qi Lu, the chief of Microsoft's Applications and Services, 2016:

> *"Data is the new oil."*

What is Big Data?

A 2016 definition states that "Big data represents the information assets characterized by such a high volume, velocity and variety to require specific technology and analytical methods for its transformation into value".

Down to earth, one can see Big Data as a set of methods and techniques for extracting useful information from high-dimensional sets of data.

The list bellow includes some of the most useful topics associated with Big Data.

- Machine Learning
- Optimization and Statistics
- Combinatorics and Discrete Mathematics
- Decision and Voting Theory

What is Machine Learning?

Machine learning is not a single approach but rather a diverse array of techniques.

ML is a blend of probability theory and statistics, linear algebra, optimization, and control theory, all worth studying in their own.

ML tools embrace classification, regression, clustering techniques, density estimation, feature (or representation) learning, matrix factorization, Bayesian networks, Markov random fields, and many others.

Related Fields and Terminology

- Artificial Intelligence
- Probability Theory and Statistical Inference
- Computational Statistics (high-dimensional statistics)
- Combinatorics
- Discrete Geometry
- Optimization
- Functional Analysis
- Data Mining
- Decision and Voting Theory

Remark

Terminology differs across different fields!!!

Definition of Machine Learning (ML)

According to [M], a machine learning algorithm is an algorithm that is able to learn from data:

> *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T, as measured by P, improves with experience E ."*

Three ingredients:

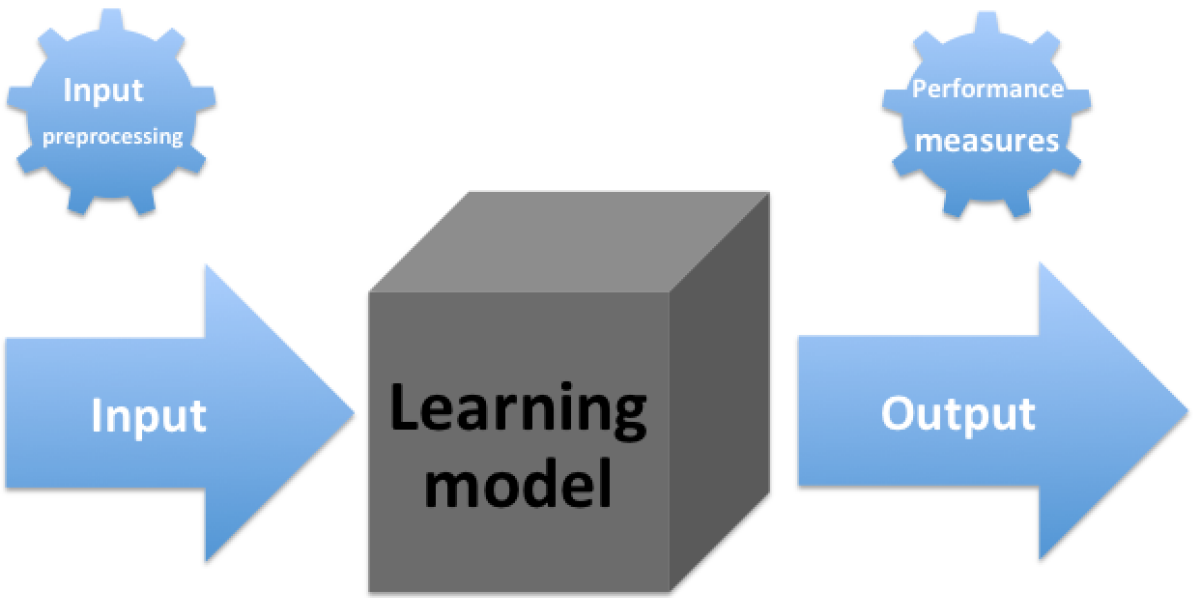
- **Experience E**
- **Task T**
- **Performance P**

Goal of Machine Learning

Goal of ML/SL: learning from data.

One wants to: execute task T based on experience E with optimal performance P.

Machine Learning Model



Main Idea

In the realm of ML these three ingredients interact in the following way:

- Select a ML algorithm (model) to solve the task T.
- The data in E is used to **train** (**estimate**) the algorithm (model) by maximizing the performance P on the training set E.
- By definition, in a ML algorithm, P should increase (error should decrease) with E.

1. Data E (Experience)
   - $\mathcal{X}$ - input space (measurement space, feature space, signal domain)
   - $\mathcal{Y}$ - output space (label space, response space, signal range)
2. Task T
   - to determine a function $f : \mathcal{X} \to \mathcal{Y}$.
3. Performance P
   - reward or utility function (its negative is a loss function)

ML is the solution of choice when dealing with tasks that are too complex to be carried out by completely solving a problem.

(1) We approach to machine learning as an input/output problem.

- Input $x \in \mathcal{X}$: contains available information for the solution of the problem, the so called **predictors** i.e.:
  - historical data
  - explanatory factors
  - features of the individuals
  - qualitative features
- Output $y \in \mathcal{Y}$: contains the solution of the problem, for instance:
  - explained (dependent) variables
  - forecasted data
  - qualitative response or classification results

(2) We distinguish between discrete-time and continuous-time setups and between deterministic and stochastic situations since they lead to very different levels of mathematical complexity.

Examples

- **Deterministic setup**: an explicit functional relation is assumed between input and output.
  - Discrete-time: observable or diagnostics variables in complex physical or noiseless systems.
  - Continuous time: integration or path continuation of differential equations.
- **Stochastic setup**: the input and the output are random variables or processes and only probabilistic dependence is assumed between them.
  - Discrete-time: image classification, time series forecasting, volatility filtering, factor analysis.
  - Continuous time: physiological time series classification, financial bubble detection.

Data E (Experience)

- $\mathcal{X}$ - input space (measurement space, feature space, signal domain)
- $\mathcal{Y}$ - output space (label space, response space, signal range)

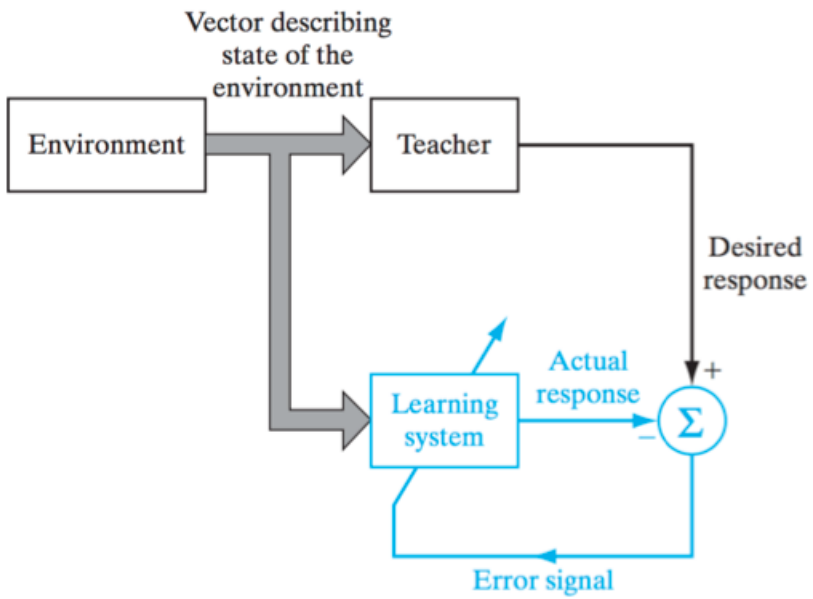
The nature of the data available determines the kind of learning algorithms that can be implemented.

The main groups are:

- **Supervised learning**
- **Unsupervised learning**
- **Reinforcement learning**

Supervised learning

- **Supervised learning**: the dataset contains features, but each example is also associated with a label or target.

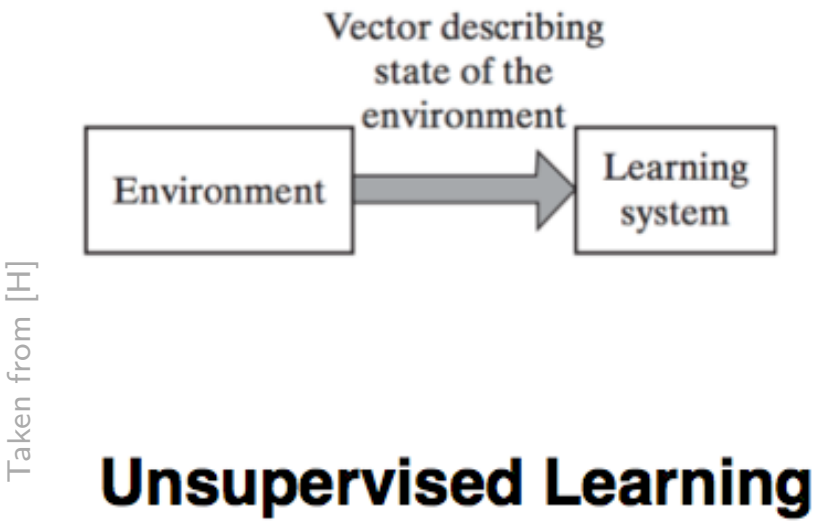


Taken from [H]

Supervised Learning

Unsupervised learning

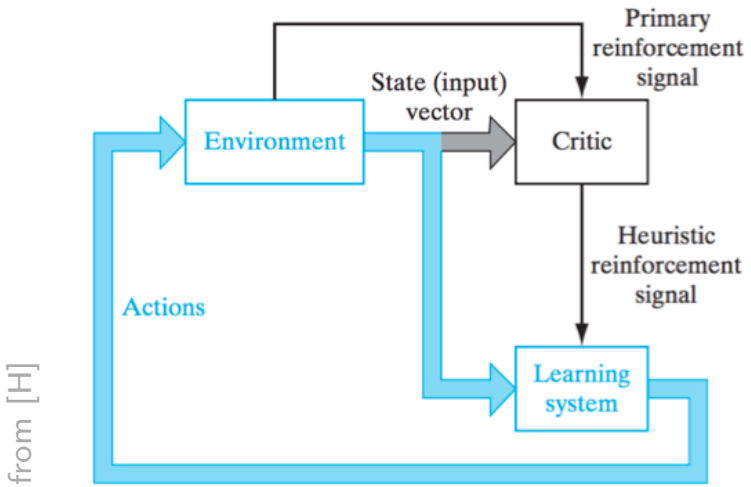
- **Unsupervised learning**: the dataset contains features, but no label or target is given. The structural properties are learnt and data is grouped based on some measure of similarity.



Taken from [H]

Reinforcement learning

- **Reinforcement learning**: algorithms that do not use a fixed dataset and interact with the environment, so there is a feedback loop between the learning system and its experiences.



Reinforcement Learning

- Example from [M, Chapter 13]: A learning robot. The robot, has a set of sensors to observe the state of its environment, and a set of actions it can perform to alter this state.

Task T: Classification

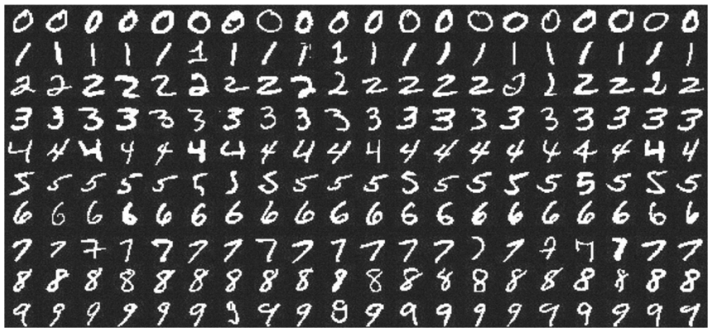
- **Classification**: assign input to one of the k categories, one is interested in producing for example $f : \mathcal{X} \longrightarrow \{1, \ldots, k\}$, with $\mathcal{X}$ containing features (think of $\mathbb{R}^n$). Function f can also be a probability distribution over classes.

Examples:

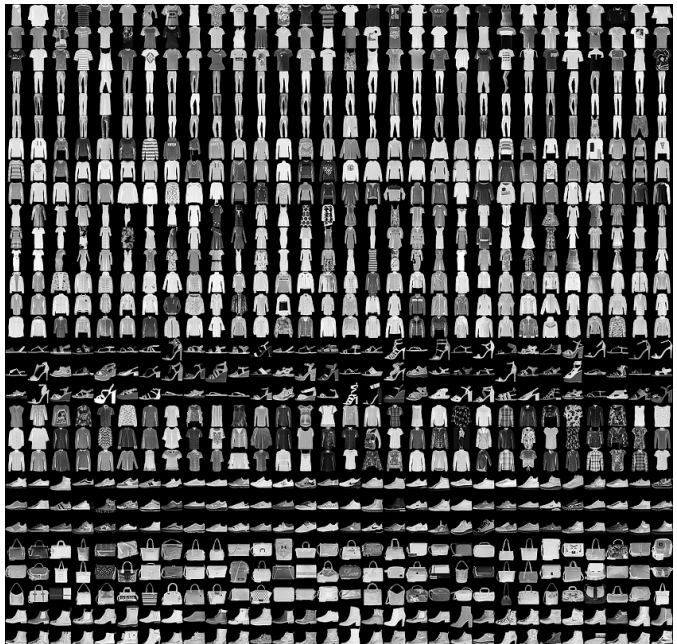
- Document classification
- Speech recognition (pronounced digits, male/female)
- Pattern recognition
- Biological (medical) classification
- Credit scoring
- Object recognition
- Handwriting recognition
- Recommender systems

Classification: Standard Examples and Datasets

- Classification of handwritten digits (MNIST)



- Classification of clothing (Fashion-MNIST)



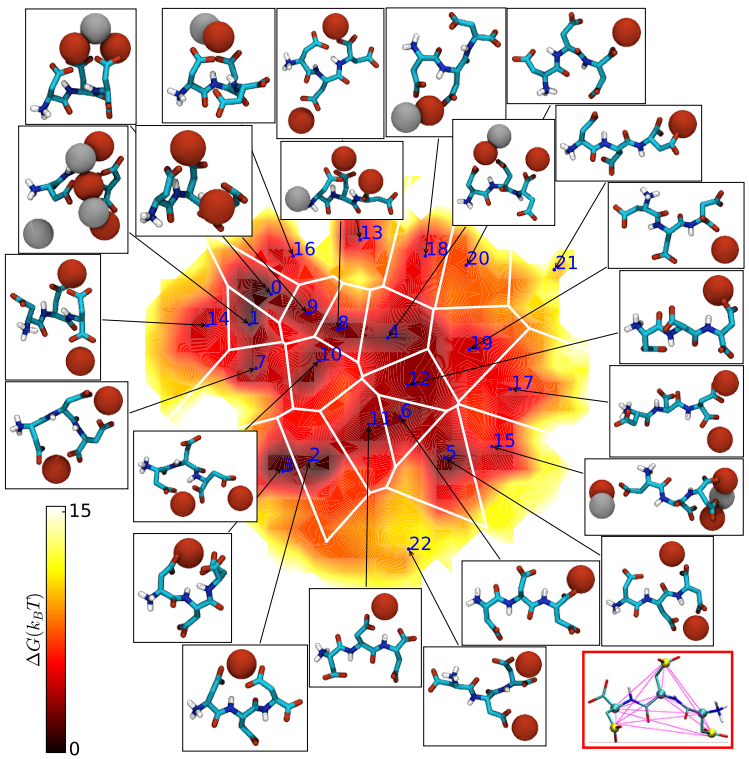
Task T: Clustering

- **Clustering**: grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters), that is learn $f : \mathcal{X} \longrightarrow \{1, \ldots, k\}$ where k need not to be specified.
  Examples:
  - Distribution-based clustering
  - Density-based clustering
  - Connectivity-based clustering (hierarchical clustering)

  Applications
  - Medical imaging
  - Business and marketing (market research, recommender systems)
  - World wide web (social network analysis)
  - Computational chemistry

Computational Chemistry Example

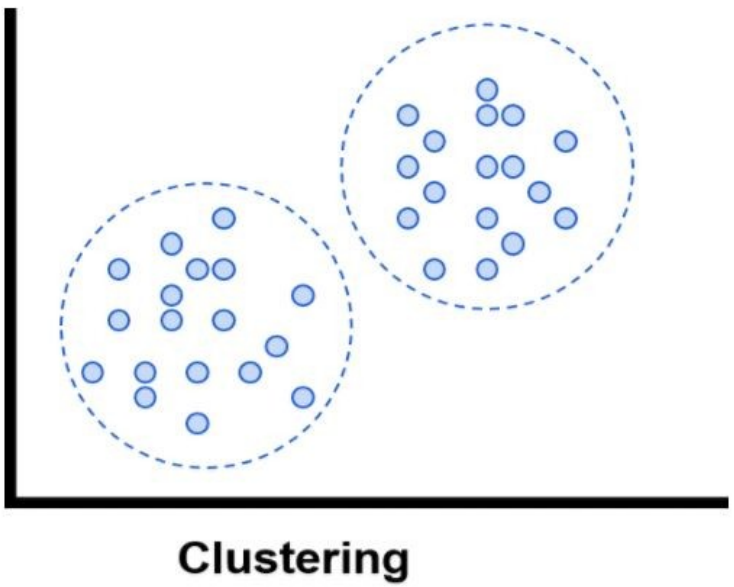
Clustering groups different states of a small peptide in the presence of ions



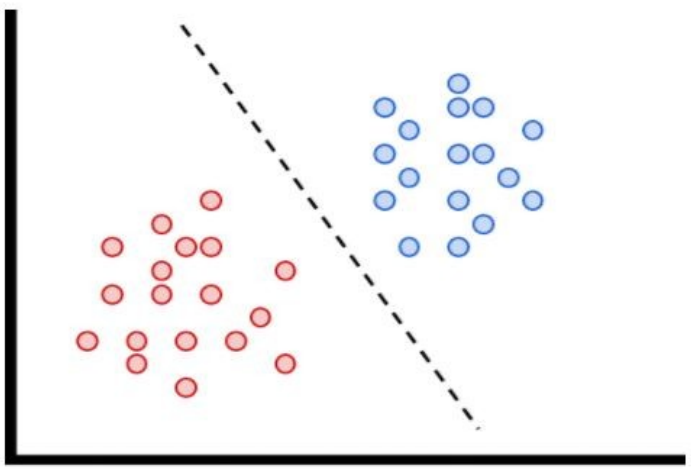
Taken from Oleksandra Kukharenko

Classification vs Clustering

Taken from Hector Klie



Clustering



Classification

Task T: Regression

- **Regression**: learn a mapping from the input (covariates) space to the output space $f : \mathcal{X} \to \mathcal{Y}$ (learn an estimator) (think of multivariate case $f : \mathbb{R}^n \to \mathbb{R}$). Examples:
  - Prediction of the expected claim amount that an insured person will make (used to set insurance premiums).
  - Prediction of future prices of securities. Used for algorithmic trading.
  - Prediction of number of passengers in a given flight.
  - Prediction of energy consumption.
  - Logistics and infrastructure management applications.

Task T: Transcription

- **Transcription**:
  unstructured representation of some kind of data $\longrightarrow$ transcribe into discrete, textual form.

  Examples:
  - Optical character recognition: a photograph containing an image of text $\longrightarrow$ this text in the form of a sequence of characters (e.g., in ASCII or Unicode format). Google Street View uses deep learning to process address numbers in this way.
  - Speech recognition: an audio waveform $\longrightarrow$ a sequence of characters or word ID codes describing the words that were spoken in the audio recording. Deep learning is a crucial component of modern speech recognition systems used at major companies including Microsoft, IBM, and Google.

Task T: Machine translation

- **Machine translation**: the input of a sequence of symbols in some language $\longrightarrow$ a sequence of symbols in another language. This is commonly applied to natural languages, such as translating from English to French. Have a look at this article in the NY Times Magazine.

Task T: Anomaly detection

- **Anomaly detection**: a set of events or objects $\longrightarrow$ some of them marked as unusual or atypical. Example: credit card fraud detection. By modeling your purchasing habits, a credit card company can detect misuse of your cards. If a thief steals your credit card or credit card information, the thief's purchases will often come from a different probability distribution over purchase types than your own. The credit card company can prevent fraud by placing a hold on an account as soon as that card has been used for an uncharacteristic purchase.

  Recent example: Detecting ICS attacks using recurrent neural networks, see here.

Task T: Synthesis and sampling

- **Synthesis and sampling**: the ML algorithm is asked to generate new examples that are similar to those in the training data. Synthesis and sampling via machine learning can be useful for media applications where it can be expensive or boring for an artist to generate large volumes of content by hand. Examples:
  - Video games: automatically generate textures for large objects or landscapes, rather than requiring an artist to manually label each pixel.
  - Speech synthesis: a written sentence $\longrightarrow$ an audio waveform containing a spoken version of that sentence. This is a kind of structured output task, but with the added qualification that there is no single correct output for each input, and we explicitly desire a large amount of variation in the output, in order for the output to seem more natural and realistic.

Two related illustrations to play with:

- Hand writing generation by Google Brain. Here
- Answer generation. Visual question answering (VQA). Here

Task T: Signal treatment

This is an inherently time-dependent domain. Some important examples are:

- **System identification (blackboxing)**: most basic learning task in engineering and signal processing. Some physical system is given which transforms some input $\mathbf{z}$ into some output signal y. One tries to recreate ("learn", "adapt") an artificial system (a computer program) which replicates the same input-output behavior. This system model can then be used for numerous goals. Examples:
  - Model the earth atmosphere (for climate change analysis and weather forecast). Input is sunshine energy and atmosphere composition, output is weather dynamics.
  - Model financial markets. Goals are obvious.
  - Model a robot body. Input: voltages to motors, output: limb motion. A prerequisite for robot motion control.

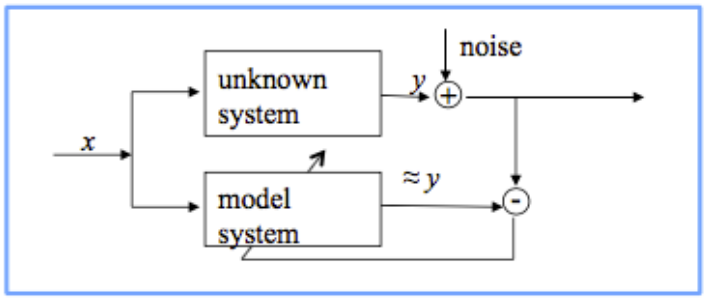
–In signal processing, such temporal input-output systems are also called **filters** or **transducers**.

–Methods for system identification have been developed a long time before the advent of modern, computer-based machine learning.

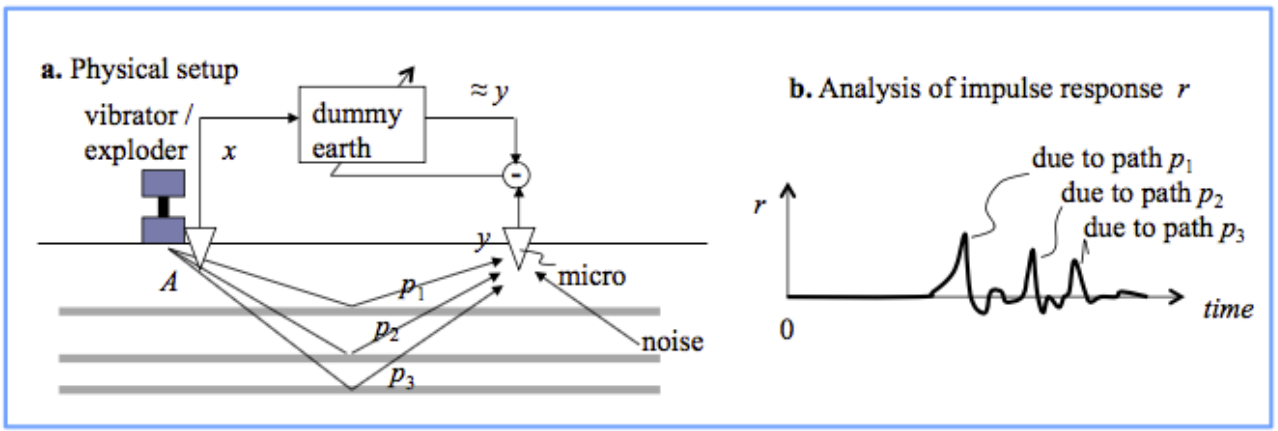
Example

Model geological structures. Input: sound waves, output: sound waves obtained at a distant location. Such sound-in, sound-out models are highly informative about underground structures. A standard tool in geological prospecting.

The basic engineering application: system identification



system identification: principle



application: ground structure exploration for oil prospecting

Taken from [J]

–Methods of ML are being adopted slowly by engineers and signal processing professionals. One reason for their reluctance: classical engineering methods for system identification are linear and can be mathematically fully analyzed. Provable criteria for correctness, accuracy and stability can be given. In contrast, high-performing ML procedures are typically nonlinear and their complexity defies mathematical analysis. While they (might) perform better, no mathematical guarantees can be given that/how they will perform in critical application circumstances. But such guarantees are often desirable (think of safety-critical applications like airplane autopilots).

–System identification is a supervised learning task: the correct system output is available to the learning system at training time.

Applications of the different tasks

- **Time series prediction**
  - financial and macroeconomics time series forecasting
  - local weather development (important for short-term power yield prediction in windmill farms)
  - predicting the consequences of action (robot action planning)
- **System control**
  - steering (or auto-piloting) engines and vehicles
  - controlling chemical production plants
- **Fault monitoring**
  - monitor power grids or power plants
  - monitor any technological device
  - driver sleep detection
- **Temporal pattern generation**
  - generating motions of robots and game characters

Comercial and societally relevant applications

- Customer profiling
- Ad placement optimization
- Financial time series prediction
- Control and monitoring of large technological systems (production plants, energy grids, internet)
- Computer games
- Brain-computer interfaces
- Automatic health diagnostic systems
- Surveillance (communication scanning, face recognition, traffic monitoring)
- Military (autonomous missiles and drones, satellite data interpretation, battlefield robotics)
- Speech and language technology

Performance

The performance evaluation P is needed during:

- Training: maximization of P determines the algorithm hyperparameters.
- Testing: the differences in P obtained during training and testing allow us to assess if we are in an under or overfitting situation.

The choice of P modifies hence how the ML algorithm is going to perform. The pertinence of a given P depends on the task.

Performance of classification

- For classification (or clustering): we often measure the **accuracy** of the model. Accuracy is the proportion of examples for which the model produces the correct output. We can also obtain equivalent information by measuring the **error rate**, the proportion of examples for which the model produces an incorrect output. We often refer to the error rate as the **expected 0-1 loss**. The 0-1 loss on a particular example is 0 if it is correctly classified and 1 if it is not. However, there are other possibilities for which it is important to understand the notion of **confusion matrix** and its derivative concepts.

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**	**False positive** (Type I error)
	Predicted condition negative	**False negative** (Type II error)	**True negative**

condition positive (P)
the number of real positive cases in the data
condition negatives (N)
the number of real negative cases in the data

true positive (TP)
eqv. with hit
true negative (TN)
eqv. with correct rejection
false positive (FP)
eqv. with false alarm, Type I error
false negative (FN)
eqv. with miss, Type II error

sensitivity, recall, hit rate, or true positive rate (TPR)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

specificity or true negative rate (TNR)

$$\mathrm{TNR} = \frac{\mathrm{TN}}{N} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

precision or positive predictive value (PPV)

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

negative predictive value (NPV)

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}}$$

miss rate or false negative rate (FNR)

$$\mathrm{FNR} = \frac{\mathrm{FN}}{P} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} = 1 - \mathrm{TPR}$$

fall-out or false positive rate (FPR)

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}$$

false discovery rate (FDR)

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TP}} = 1 - \mathrm{PPV}$$

false omission rate (FOR)

$$\mathrm{FOR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TN}} = 1 - \mathrm{NPV}$$

accuracy (ACC)

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{P + N} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

Wikipedia

False positives (FP) and false negatives (FN)

Figure: False positive (FP) and false negative (FN)

Performance of regression

- For regression: usually the mean square error (MSE) or the residual sum of squares (RSS) are used. If our dataset (training or testing) contains N samples (epochs) of the form $(y_i, \mathbf{x}_i)$ and $f : \mathbb{R}^n \to \mathbb{R}$ is the regression map then the associated (training or testing) MSE is:

$$\mathrm{MSE}_f := \frac{1}{N} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 ,$$

  while the associated (training or testing) MSE is:

$$\mathrm{RSS}_f := \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 .$$

What would you choose as performance for the following ML systems?

- Pregnancy test
- HIV test
- EEG (electroencephalogram) signal processing system to detect awareness in comatose patients
- Gas bag detection system in drilling platforms
- Ore detection systems

Comment: It is often difficult to choose a performance measure that corresponds well to the desired behavior of the system. Examples:

- Transcription task: should we measure the accuracy of the system at transcribing entire sequences, or should we use a more fine-grained performance measure that gives partial credit for getting some elements of the sequence correct?
- Regression task: should we penalize the system more if it frequently makes medium-sized mistakes or if it rarely makes very large mistakes? These kinds of design choices depend on the application.

Recent Achievements of ML

2006		The Netflix Prize	The Netflix Prize competition is launched by Netflix. The aim of the competition was to use machine learning to beat Netflix's own recommendation software's accuracy in predicting a user's rating for a film given their ratings for previous films by at least 10%.[40] The prize was won in 2009.
2010		Kaggle Competition	Kaggle, a website that serves as a platform for machine learning competitions, is launched.[41]
2011	Achievement	Beating Humans in Jeopardy	Using a combination of machine learning, natural language processing and information retrieval techniques, IBM's Watson beats two human champions in a Jeopardy! competition.[42]
2012	Achievement	Recognizing Cats on YouTube	The Google Brain team, led by Andrew Ng and Jeff Dean, create a neural network that learns to recognise cats by watching unlabeled images taken from frames of YouTube videos.[43][44]
2014		Leap in Face Recognition	Facebook researchers publish their work on DeepFace, a system that uses neural networks that identifies faces with 97.35% accuracy. The results are an improvement of more than 27% over previous systems and rivals human performance.[45]
2014		Sibyl	Researchers from Google detail their work on Sibyl,[46] a proprietary platform for massively parallel machine learning used internally by Google to make predictions about user behavior and provide recommendations.[47]
2016	Achievement	Beating Humans in Go	Google's AlphaGo program becomes the first Computer Go program to beat an unhandicapped professional human player[48] using a combination of machine learning and tree search techniques.[49]

Figure: Taken from Wikipedia page

Examples of Successful Applications

ML algorithms are trained to solve problems that were considered impossible only a few years back:

- face recognition (see Face-Recognition-Papers)
- online text translation
- (almost) inferring a Turing machine from input-output examples (Google DeepMind)
- playing the game of Go at and beyond the level of human grand-masters (Google DeepMind)

Examples of Successful Applications

Other examples:

- Website of one of the "deep learning" pioneers, Geoffrey Hinton (University of Toronto) examples of caption phrases that have been automatically generated by a neural network which was given a photographic image as input (see these examples here)
- Computer Vision: https://cloud.google.com/vision/
- ConvNetJS Demos: http://cs.stanford.edu/people/karpathy/convnetjs/
- Reconstructing physics laws: http://otoro.net/ml/pendulum-esp-mobile/
- Handwriting: https://distill.pub/2016/handwriting/
- Machine Learning with Financial Time Series Data on Google Cloud Platform

Toronto Deep Learning demos

Results

Tags

- riders
- bronc
- steed
- ponies
- rider

Nearest Caption in the Training Dataset

a parade of a horse drawn carriage and horses are going down a street in london .

Generated Captions

- people are riding on a horse drawn carriage in a parade .
- three riders on horseback in front of a parade .
- a group of people riding on the backs of horses in the parade .
- two people on a street with horses and riders .
- two people riding horses in a street with a carriage .

back

Results

Tags

- jeep
- mule
- jeeps
- vehicle
- motorcylce

Nearest Caption in the Training Dataset

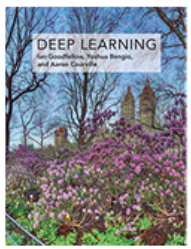
a brown cow and a white car are in a pasture .

Generated Captions

- a cow is standing in the grass by a car .
- a cow is in a vehicle , next to a rural area .
- a cow standing next to a rural road .
- a large black cow standing next to a car .
- a cow and a car are looking at the camera .

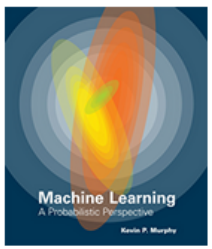
back

Classification applications: Amazon Recommender Systems



Deep Learning (Adaptive Computation and Machine Learning series)
› Ian Goodfellow
★★★★☆ 6
Kindle Edition
EUR 46,89

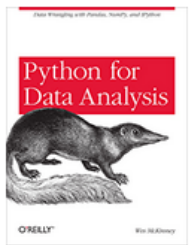
Hands-On Machine Learning with Scikit-Learn and TensorFlow:...
› Aurélien Géron
★★★★★ 7
Kindle Edition
EUR 23,99



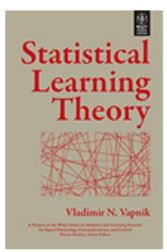
Machine Learning: A Probabilistic Perspective (Adaptive Computation...
› Kevin P. Murphy
★★★★★ 2
Kindle Edition
EUR 57,76

Applied Predictive Modeling
› Max Kuhn
★★★★★ 4
Kindle Edition
EUR 49,97

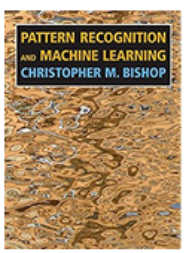
Python Machine Learning
› Sebastian Raschka
★★★★☆ 5
Kindle Edition
EUR 27,48



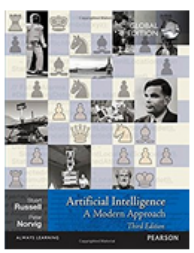
Python for Data Analysis: Data Wrangling with Pandas, NumPy, and...
› Wes McKinney
★★★★☆ 6
Kindle Edition
EUR 24,00



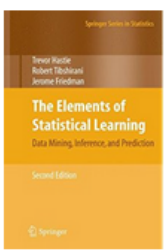
STATISTICAL LEARNING THEORY
★☆☆☆☆ 1
Taschenbuch
7 Angebote ab EUR 21,00



Pattern Recognition and Machine Learning (Information Science...
› Christopher M. Bishop
★★★★☆ 15
Gebundene Ausgabe
EUR 74,18 prime



Artificial Intelligence: A Modern Approach, Global Edition
› Stuart Russell
★☆☆☆☆ 2
Taschenbuch
EUR 61,99 prime



The Elements of Statistical Learning: Data Mining, Inference, and...
Trevor Hastie
★★★½☆ 10
Gebundene Ausgabe
EUR 70,99 prime

Netflix Prize - an open competition

Netflix Prize: the best collaborative algorithm to predict user ratings for films, based on previous ratings without info about users or films

- Training data set: 100,480,507 ratings that 480,189 users gave to 17,770 movies. Each training rating is [user, movie, date of grade, grade]. The user and movie fields are integer IDs, while grades are from 1 to 5 (integer) stars
- Testing data: 2,817,131 of [user, movie, date of grade], grades known only to the jury

On September 21, 2009, the prize of US$1,000,000 was given to the BellKor's Pragmatic Chaos team superior to the Netflix's own algorithm by 10.06% (see Presentation of the Netflix Grand Prize Solution 2009 or their paper). Some remarks:

- sparse system (many users do not rate, only 1.1% observed)
- not a counterfactual but the out-of-sample prediction; not questioning - what would have happened if (Amazon changed their pricing policy)

Classification applications: Recommender Systems

Strands of Recommendation Literature

"Unsupervised": Item Similarity

- Suppose observe item characteristics
- Recommend similar items to what the user likes
- Use unsupervised learning techniques to reduce dimensionality (see earlier lectures)

"Supervised": Collaborative Filtering

- Suppose observe choice data but limited item characteristics
- Find other users with similar tastes
- Recommend items liked by similar users
- E.g. Matrix decomposition

Hierarchical/Graphical Models

Decoupling problem into similarities, grouping into categories (unsupervised) and the rest (for supervised) makes a lot of sense. Same methods can be used. Taken from http://www.nber.org/econometrics_minicourse_2015/.

Example: predicting house prices

Mullainathan, S., and Jann S. 2017. Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives*, 31(2): 87-106 Harvard, (pdf)

- Consider 10,000 randomly selected owner-occupied units from the 2011 metropolitan sample of the American Housing Survey
- Include 150 variables that contain information about the unit and its location, such as the number of rooms, the base area, and the census region within the United States
- evaluate how well each approach predicts (log) unit value on a separate hold-out set of 41,808 units from the same sample

Example: predicting house prices

Performance of Different Algorithms in Predicting House Values

	Prediction performance (R^2)		*Relative improvement over ordinary least squares by quintile of house value*				
Method	*Training sample*	*Hold-out sample*	1st	2nd	3rd	4th	5th
Ordinary least squares	47.3%	41.7% [39.7%, 43.7%]	–	–	–	–	–
Regression tree tuned by depth	39.6%	34.5% [32.6%, 36.5%]	−11.5%	10.8%	6.4%	−14.6%	−31.8%
LASSO	46.0%	43.3% [41.5%, 45.2%]	1.3%	11.9%	13.1%	10.1%	−1.9%
Random forest	85.1%	45.5% [43.6%, 47.5%]	3.5%	23.6%	27.0%	17.8%	−0.5%
Ensemble	80.4%	45.9% [44.0%, 47.9%]	4.5%	16.0%	17.9%	14.2%	7.6%

Note: The dependent variable is the log-dollar house value of owner-occupied units in the 2011 American Housing Survey from 150 covariates including unit characteristics and quality measures. All algorithms are fitted on the same, randomly drawn training sample of 10,000 units and evaluated on the 41,808 remaining held-out units. The numbers in brackets in the hold-out sample column are 95 percent bootstrap confidence intervals for hold-out prediction performance, and represent measurement variation for a fixed prediction function. For this illustration, we do not use sampling weights. Details are provided in the online Appendix at http://e-jep.org.

Lab Schedule

Lab 0	Feb 20, 18:00-21:00, 204
Lab 1	Feb 27, 18:00-21:00, 204
Lab 1	Mar 06, 18:00-21:00, 204
Lab 2	Mar 13, 18:00-21:00, 204
Lab 2	Mar 20, 18:00-21:00, 204
Lab 3	Mar 27, 18:00-21:00, 204
Lab 3	Apr 10, 18:00-21:00, 204
Lab 4	Apr 17, 18:00-21:00, 204
Lab 4	Apr 24, 18:00-21:00, 204
Lab* 5	May 08, 18:00-21:00, 204
Lab* 5	May 15, 18:00-21:00, 204
Lab* 6	May 22, 18:00-21:00, 204
Lab* 6	May 29, 18:00-21:00, 204
Lab? 7	Jun 05, 18:00-21:00, 204

References


T. M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.


Herbert Jaeger. *Machine learning.* Course Notes, Jacobs University Bremen.


S. Haykin. *Neural Networks and Learning Machines.* Pearson, Addison Wesley, 2009.