RegML 2013

Regularization Methods for High Dimensional Learning

Genova, **June, 3-7 2013**

Francesca Odone and Lorenzo Rosasco

Course organized within the PhD Program in Computer Science *for the*
PhD School in Sciences and Technologies for Information and Knowledge (STIC)
PhD School in Life and Humanoid Technologies

Who are we?

The course is co-organized by

- **SLIPGURU, University of Genova**
- **Laboratory for Computational and Statistical Learning, Istituto Italiano di Tecnologia and Massachusetts Institute of Technology**

Schedule

Syllabus

- each class is 90 min. no breaks -

class 1 (C1) Welcome. Introduction to Learning
class 2 (C2) Kernels, Dictionaries, and Regularization
class 3 (C3) Regularization Networks and Support Vector Machines
class 4 (C4) Error Analysis and Parameter Choice
class 5 (C5) Lab 1
class 6 (C6) Spectral methods for supervised learning
class 7 (C7) Multi-Output Learning
class 8 (C8) Lab 2
class 9 (C9) Sparsity-based Regularization
class 10 (C10) Multiple Kernel Learning
class 11 (C11) Lab 3
class 12 (C12) Manifold regularization
class 13 (C13) Applications to high dimensional problems

Schedule and rooms

	MON 3	TUE 4	WED 5	THU 6	FRI 7
9:30-11:00	-	C3	C6	C9	C12
11:30-13:00	C1	C4	C7	C10	C13
14:30-16:00	C2	C5 (lab)	C8 (lab)	C11 (lab)	-

- room 322 (*sala conferenze*) - 3rd floor

Course Schedule and Material

disi.unige.it/dottorato/corsi/RegML2013/

Other Sources

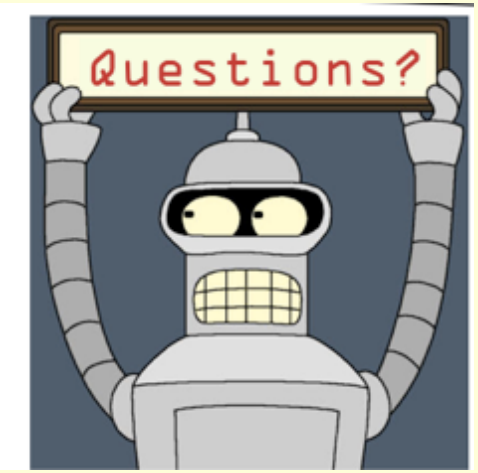
- Slipguru: **slipguru.disi.unige.it**
- LCSL: **lcsl.mit.edu**

Instructors e-mails

odone@disi.unige.it, lrosasco@mit.edu

Teaching Assistants

Nicoletta Noceti
Silvia Villa
Alessandra Stagliano'
Sean Fanello
Gabriele Chiusano
Alessandro Rudi
Luca Zini



Exams?
Credits?
Certificates?
Attendance?
Housing?

What We Talk About When We Talk About (Machine) Learning

Francesca Odone and Lorenzo Rosasco

RegML 2013

Menu'

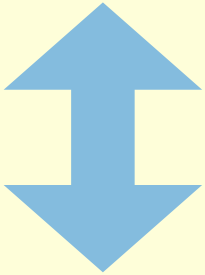
Appetizer: AI some context and some history

Entree: Machine Learning at a Glance

Main Course: Intro to Statistical Learning Theory

(Artificial) Intelligence

Build intelligent machines

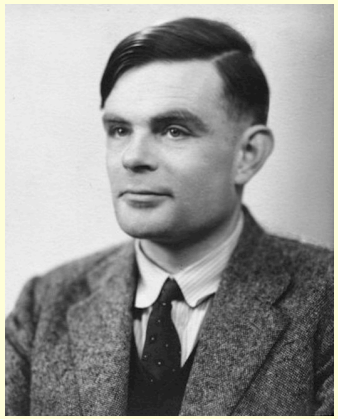


Understand Intelligence

Science and Engineering of Intelligence

(Artificial) Intelligence: A Working Definition

Turing test



Alan Turing 1912-1954

Ingredients for AI

- natural language processing
- knowledge representation
- automated reasoning
- machine learning

- computer vision
- robotics to manipulate

(Artificial) Intelligence & its Neighbors

Neuroscience

Phylosophy

Mathematics

Psychology

AI

Engineering

Cognitive Science

- What are the formal rules to draw valid conclusions?
- What can be computed?
- How do we reason with uncertain information?

Computer Science

Birth of a Dream

1943
Arturo Rosenblueth, Norbert Wiener and Julian Bigelow coin the term "cybernetics". Wiener's popular book by that name published in 1948.
1945
Game theory which would prove invaluable in the progress of AI was introduced with the 1944 paper, Theory of Games and Economic Behavior by mathematician John von Neumann and economist Oskar Morgenstern.
1945
Vannevar Bush published *As We May Think* (The Atlantic Monthly, July 1945) a prescient vision of the future in which computers assist humans in many activities.
1948
John von Neumann (quoted by E.T. Jaynes) in response to a comment at a lecture that it was impossible for a machine to think: "You insist that there is something a machine cannot do. If you will tell me *precisely* what it is that a machine cannot do, then I can always make a machine which will do just that!". Von Neumann was presumably alluding to the Church-Turing thesis which states that any effective procedure can be simulated by a (generalized) computer.
...
1950
Alan Turing proposes the Turing Test as a measure of machine intelligence.
1950
Claude Shannon published a detailed analysis of chess playing as search.
1955
The first Dartmouth College summer AI conference is organized by John McCarthy, Marvin Minsky, Nathan Rochester of IBM andClaude Shannon.
1956
The name *artificial intelligence* is used for the first time as the topic of the second Dartmouth Conference, organized by John McCarthy[30]

.....

How did it go?

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for **a summer.**

Dartmouth Summer Research Conference on Artificial Intelligence organised by John McCarthy and proposed by McCarthy, Marvin Minsky, Nathaniel Rochester and Claude Shannon.

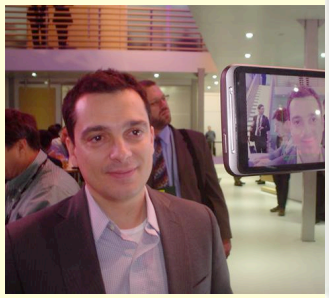
Late 1990s
Web crawlers and other AI-based information extraction programs become essential in widespread use of the World Wide Web.
1997
The Deep Blue chess machine (IBM) beats the world chess champion, Garry Kasparov.
2004
DARPA introduces the DARPA Grand Challenge requiring competitors to produce autonomous vehicles for prize money.

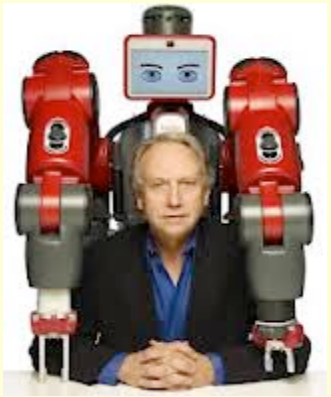
10/15 years ago

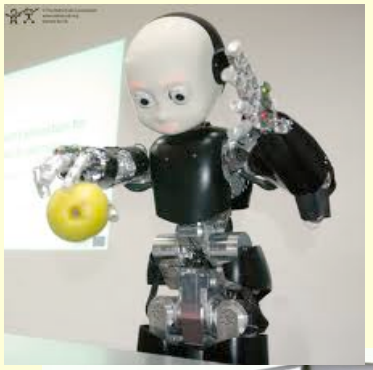
How are we doing now?

Pedestrians Detection at Human Level Performance

ML and AI

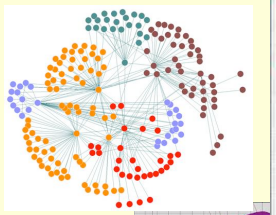


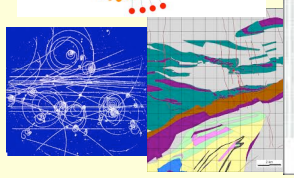


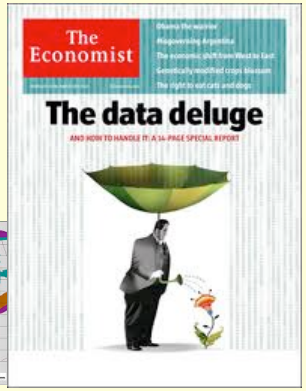


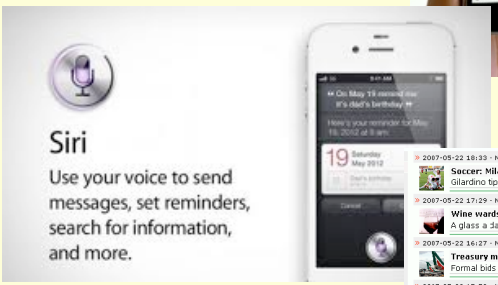
Machine Learning

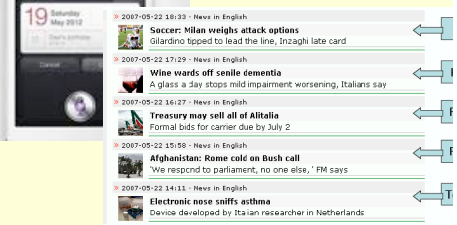
systems are trained
on *examples*
rather than being
programmed

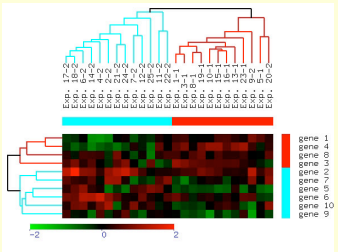


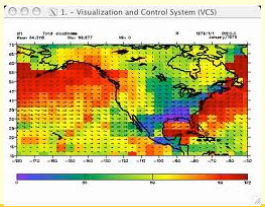












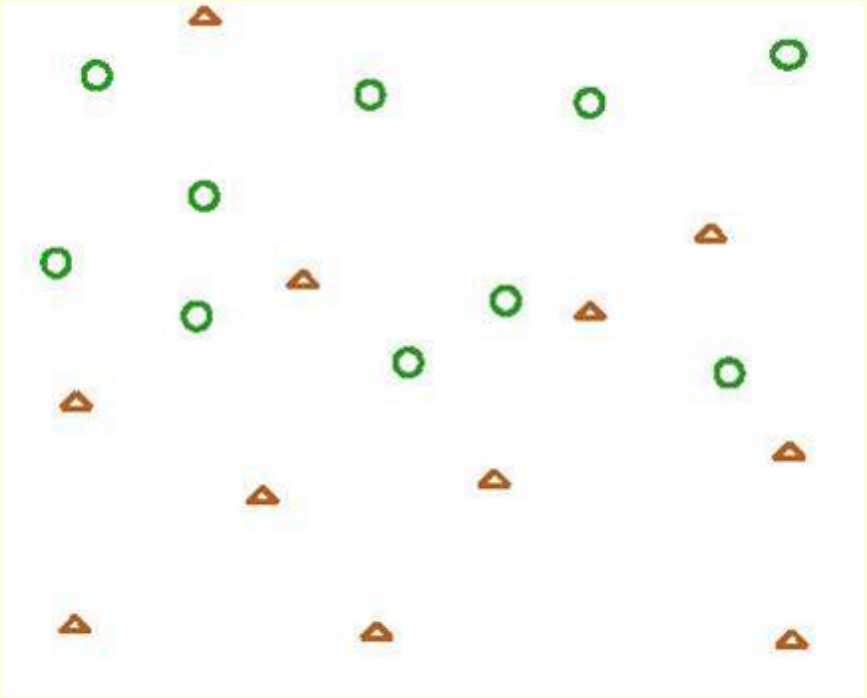
Menu'

Appetizer: AI some context and some history

Entree: Machine Learning at a Glance

Main Course: Intro to Statistical Learning Theory

Basic Setting: Classification

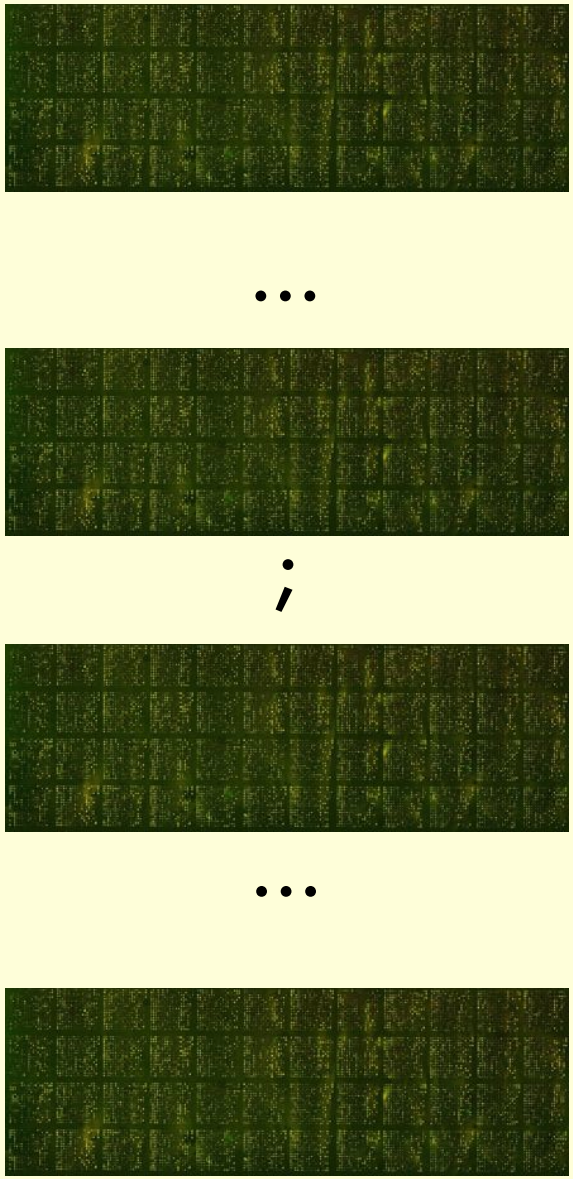


$(x_1, y_1), \ldots, (x_n, y_n)$

$x_i \in \mathbb{R}^p$ and $y_i \in Y = \{-1, 1\}$, $i = 1, \ldots, n$

$$X_n = \begin{pmatrix} x_1^1 & \ldots & \ldots & \ldots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \ldots & \ldots & \ldots & x_n^p \end{pmatrix} \qquad Y_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Genomics



n patients p gene expression measurements

$$X_n = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \dots & \dots & \dots & x_n^p \end{pmatrix}; Y_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Text Classification

» 2007-05-22 18:33 · News in English
Soccer: Milan weighs attack options ← Sport (football)
Gilardino tipped to lead the line, Inzaghi late card

» 2007-05-22 17:29 · News in English
Wine wards off senile dementia ← Health
A glass a day stops mild impairment worsening, Italians say

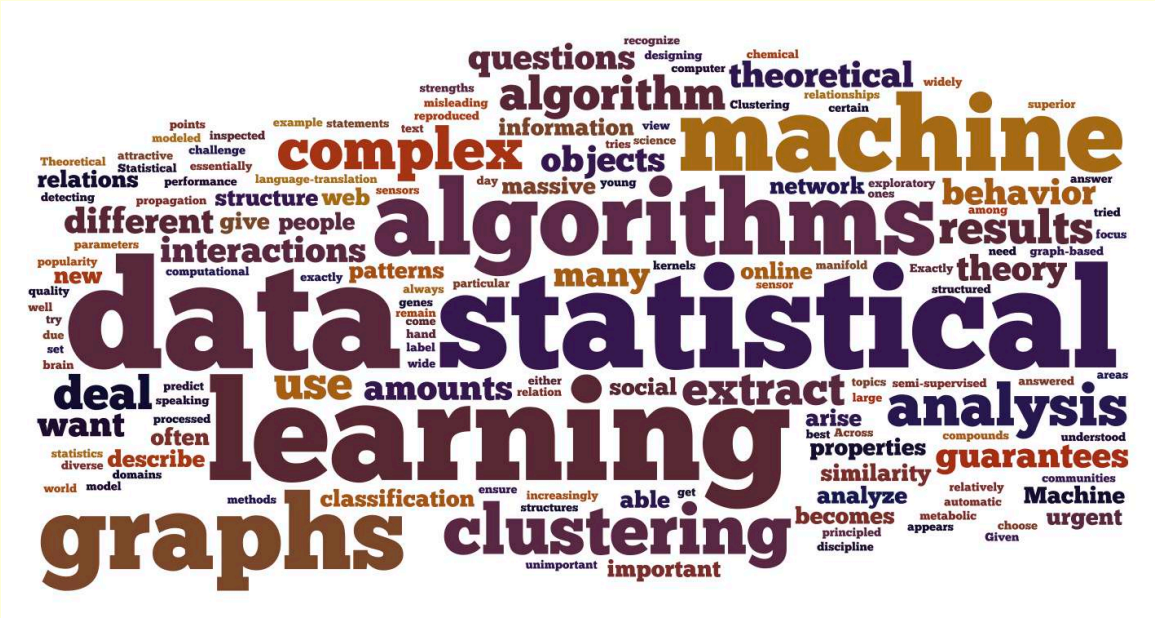
» 2007-05-22 16:27 · News in English
Treasury may sell all of Alitalia ← Politics (economy)
Formal bids for carrier due by July 2

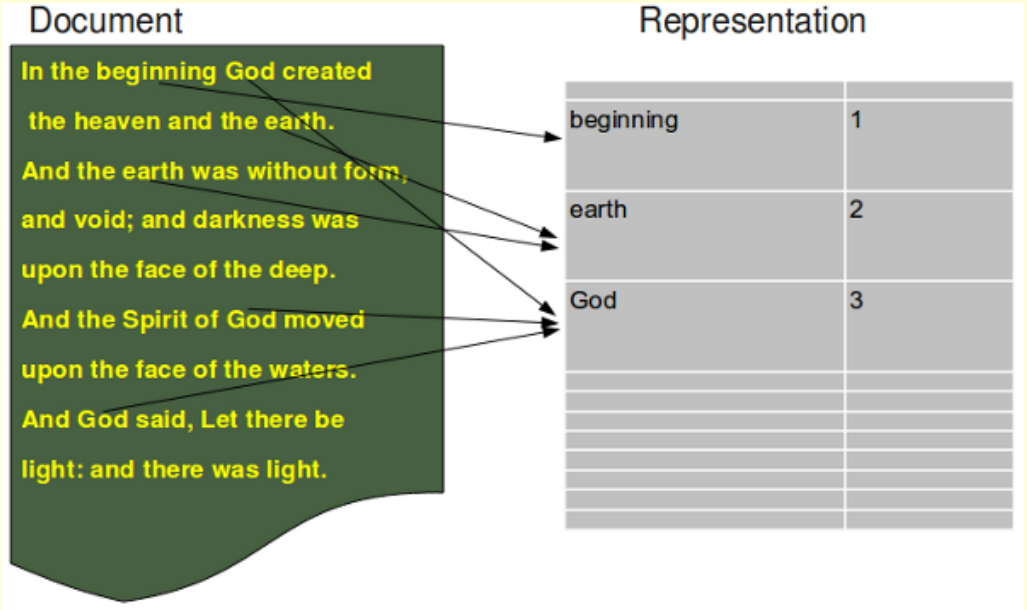
» 2007-05-22 15:58 · News in English
Afghanistan: Rome cold on Bush call ← Politics (foreign)
'We respcnd to parliament, no one else, ' FM says

» 2007-05-22 14:11 · News in English
Electronic nose sniffs asthma ← Technology
Device developed by Italian researcher in Netherlands

Subject	Date	Time	Body	Spam?
I has the viagra for you	03/12/1992	12:23 pm	Hi! I noticed that you are a software engineer so here's the pleasure you were looking for…	Yes
Important business	05/29/1995	01:24 pm	Give me your account number and you'll be rich. I'm totally serial	Yes
Business Plan	05/23/1996	07:19 pm	As per our conversation, here's the business plan for our new venture Warm regards…	No
Job Opportunity	02/29/1998	08:19 am	Hi !I am trying to fill a position for a PHP …	Yes
[A few thousand rows ommitted]				
Call mom	05/23/2000	02:14 pm	Call mom. She's been trying to reach you for a few days now	No

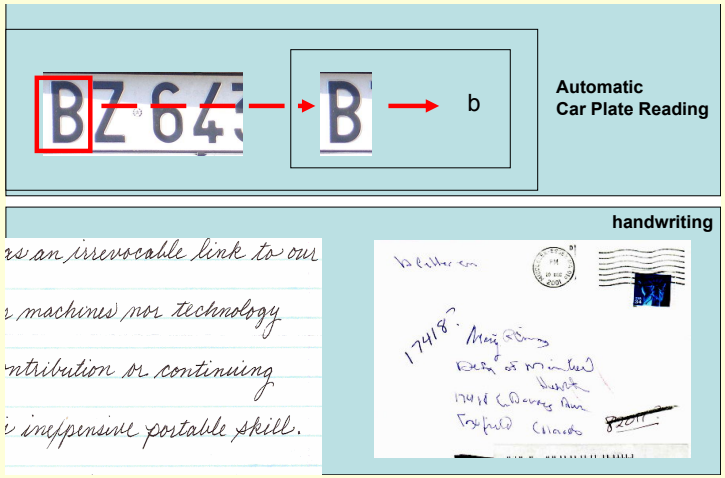
Text Classification: Bag of Words

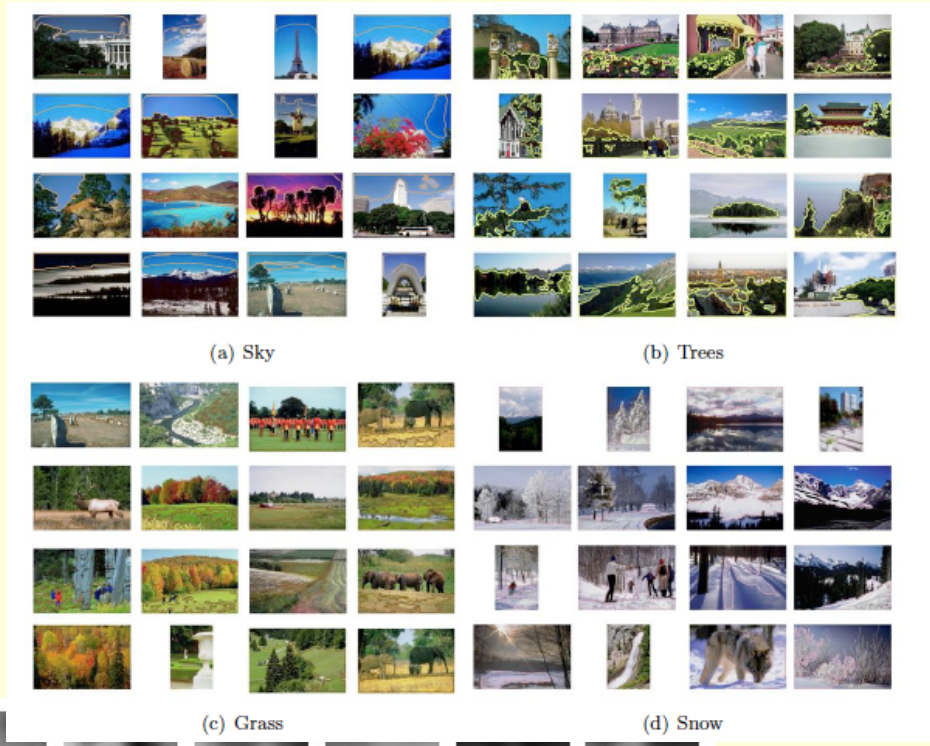




$$X_n = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \dots & \dots & \dots & x_n^p \end{pmatrix}$$

Image Classification





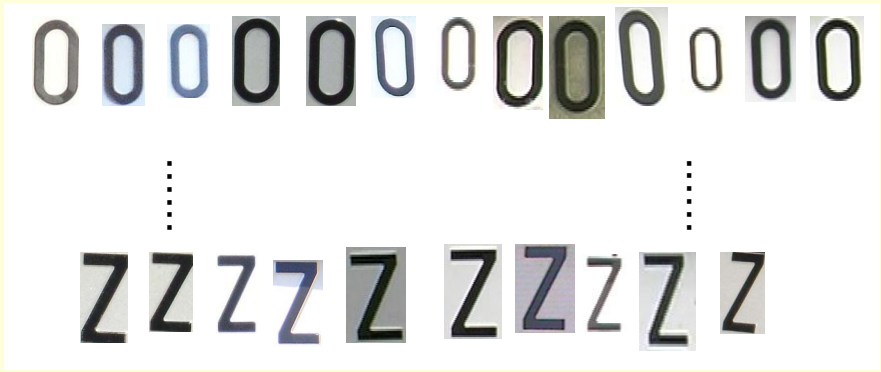
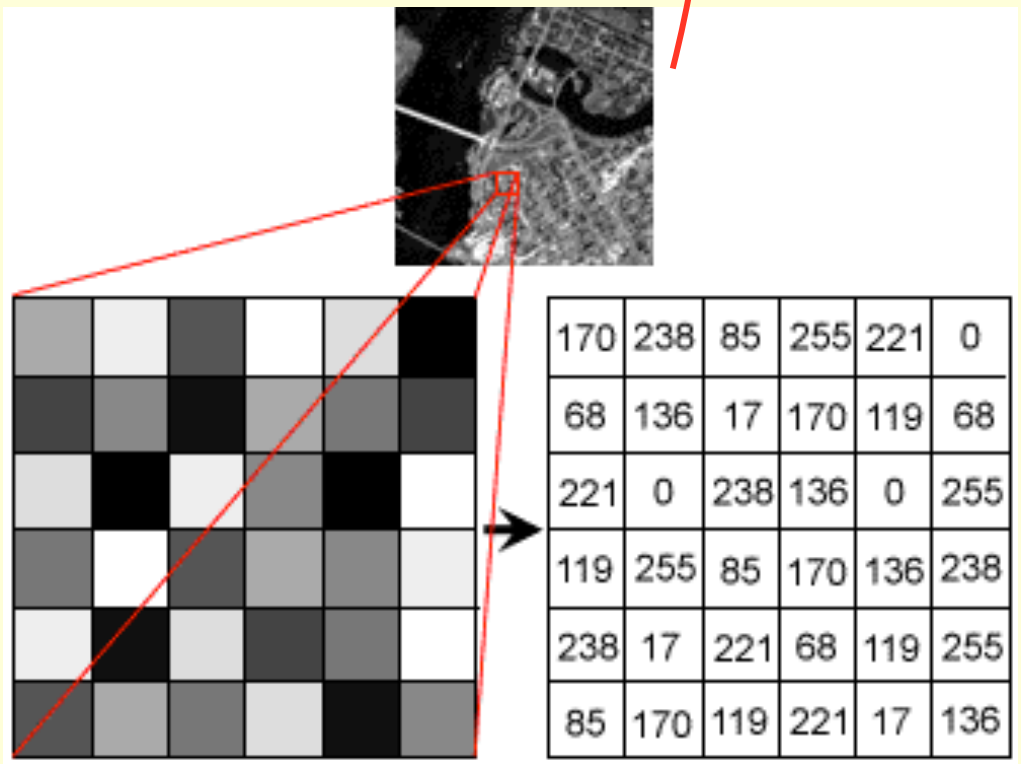


Image Classification

$$X_n = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \dots & \dots & \dots & x_n^p \end{pmatrix}$$



170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

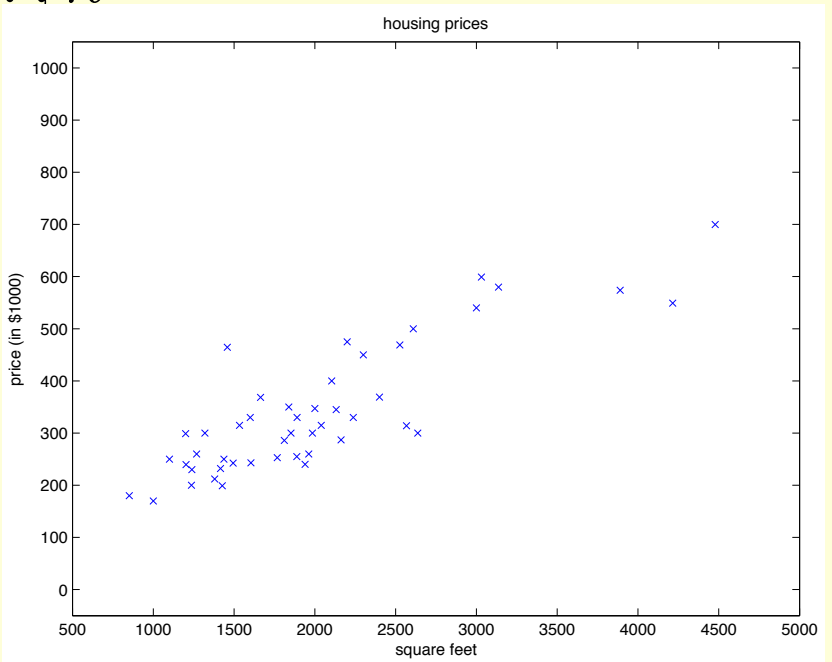
From classification to regression

$$(x_1, y_1), \ldots, (x_n, y_n)$$

$x_i \in \mathbb{R}^D$ and ~~$y_i \in Y = \{-1, 1\}, i = 1, \ldots, n$~~

$$y_i \in Y \in \mathbb{R}, i = 1, \ldots, n$$

Living area (feet^2)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
$\vdots$	$\vdots$



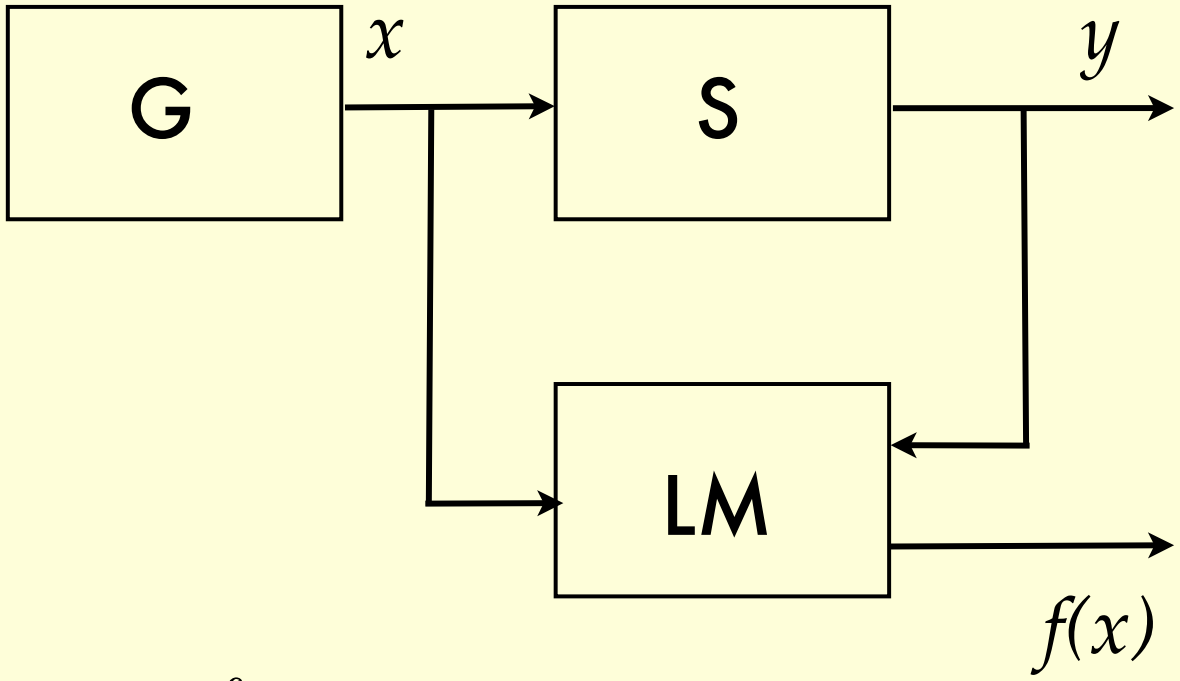
Living area (feet^2)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
$\vdots$	$\vdots$	$\vdots$

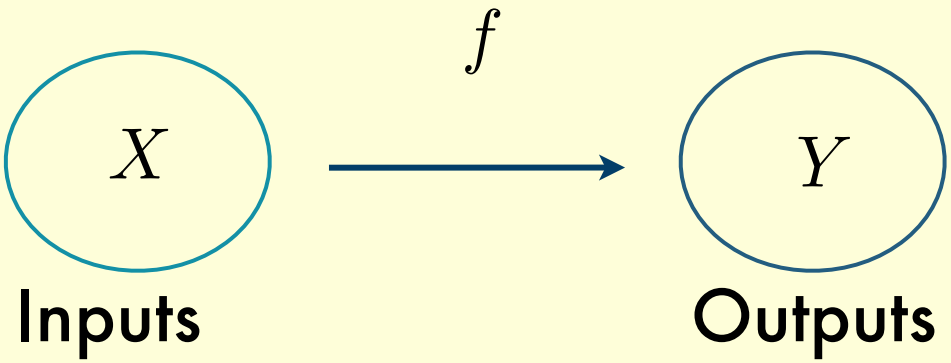
$$y_i = f(x_i) + \sigma\varepsilon_i, \quad \sigma > 0$$

$$\text{e.g.} \quad f(x) = w^T x, \quad \varepsilon_i \sim N(0, 1)$$

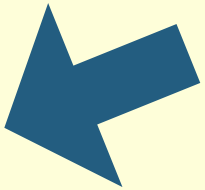
Batch Learning

$$(x_1, y_1), \ldots, (x_n, y_n)$$





Machine Learning: Problems and Approaches



Learning Problems

- Supervised Learning
- Semisupervised
- Online
- ....

Learning Approaches

- Batch Learning
- Online
- Active
- ...

Variations on a Theme

$$(x_1, y_1), \ldots, (x_n, y_n)$$

Multiclass: $x_i \in \mathbb{R}^D$ and $y_i \in Y = \{1, \ldots, T\}$, $i = 1, \ldots, n$

Multitask: $x_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}^T$, $i = 1, \ldots, n$

Learning a similarity function

$$(x_1, x_1; y_{1,1}), (x_1, x_2; y_{1,2}) \ldots, (x_n, x_n; y_{n,n})$$

$$x_j, x_i \in \mathbb{R}^D \text{ and } y_{i,j} \in [0, 1], j, i = 1, \ldots, n$$

Semisupervised Learning

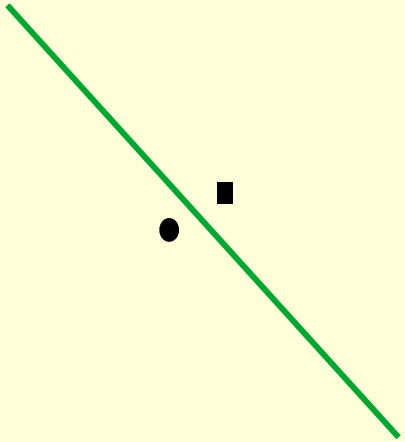
$$X_n = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \dots & \dots & \dots & x_n^p \end{pmatrix}; Y_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \cup \quad X_u = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_u^1 & \dots & \dots & \dots & x_u^p \end{pmatrix}$$

Semisupervised Learning

$$X_n = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \dots & \dots & \dots & x_n^p \end{pmatrix}; Y_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \cup \quad X_u = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_u^1 & \dots & \dots & \dots & x_u^p \end{pmatrix}$$

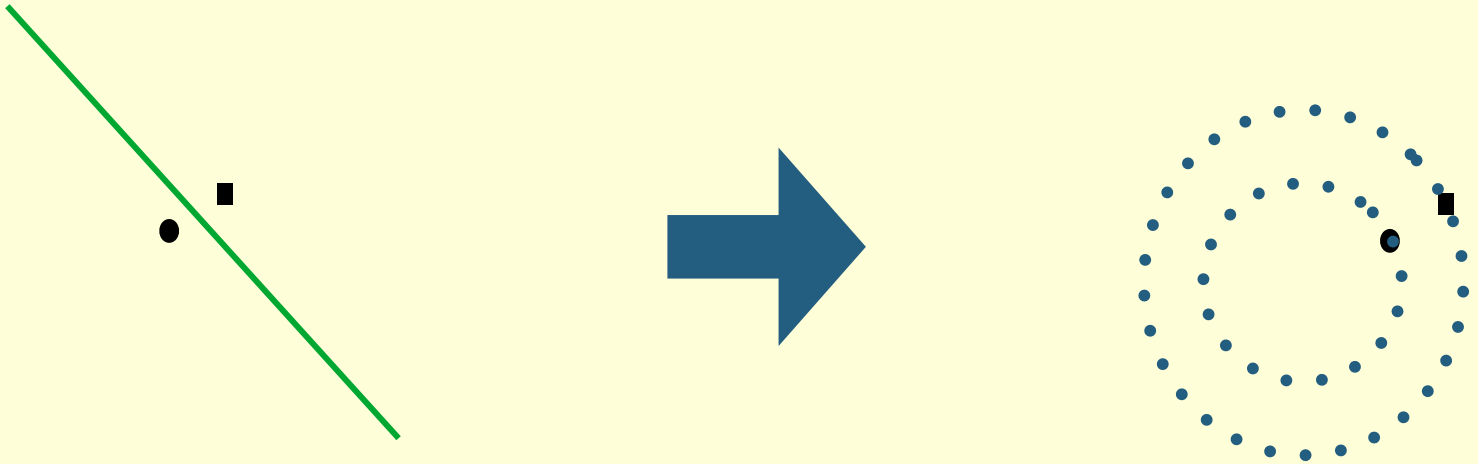
Semisupervised Learning

$$X_n = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \dots & \dots & \dots & x_n^p \end{pmatrix}; Y_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \cup \quad X_u = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_u^1 & \dots & \dots & \dots & x_u^p \end{pmatrix}$$



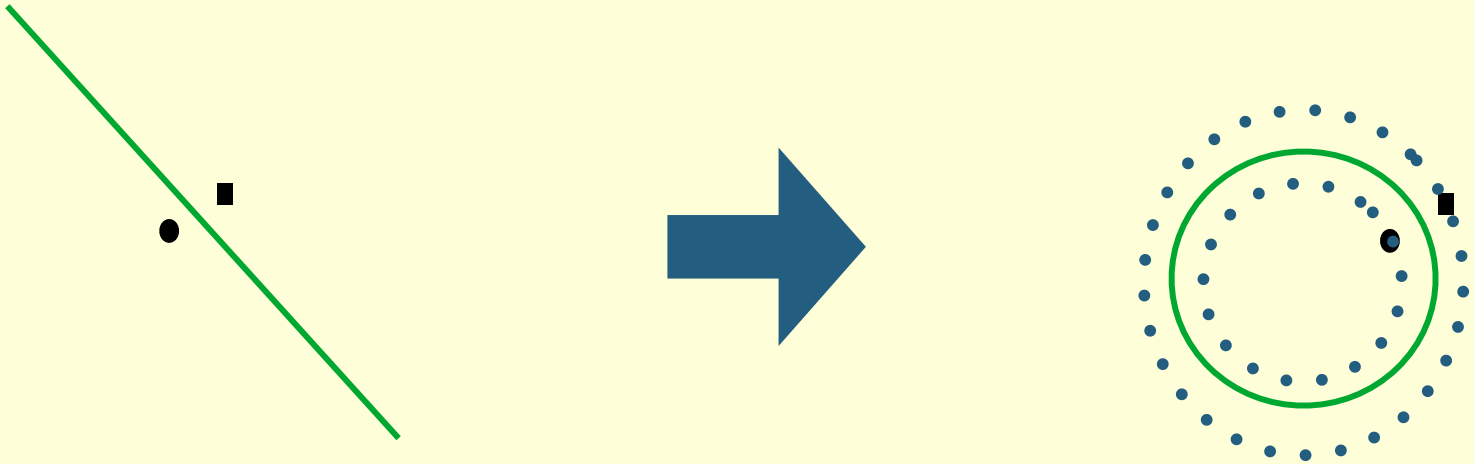
Semisupervised Learning

$$X_n = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \dots & \dots & \dots & x_n^p \end{pmatrix}; Y_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \cup \quad X_u = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_u^1 & \dots & \dots & \dots & x_u^p \end{pmatrix}$$



Semisupervised Learning

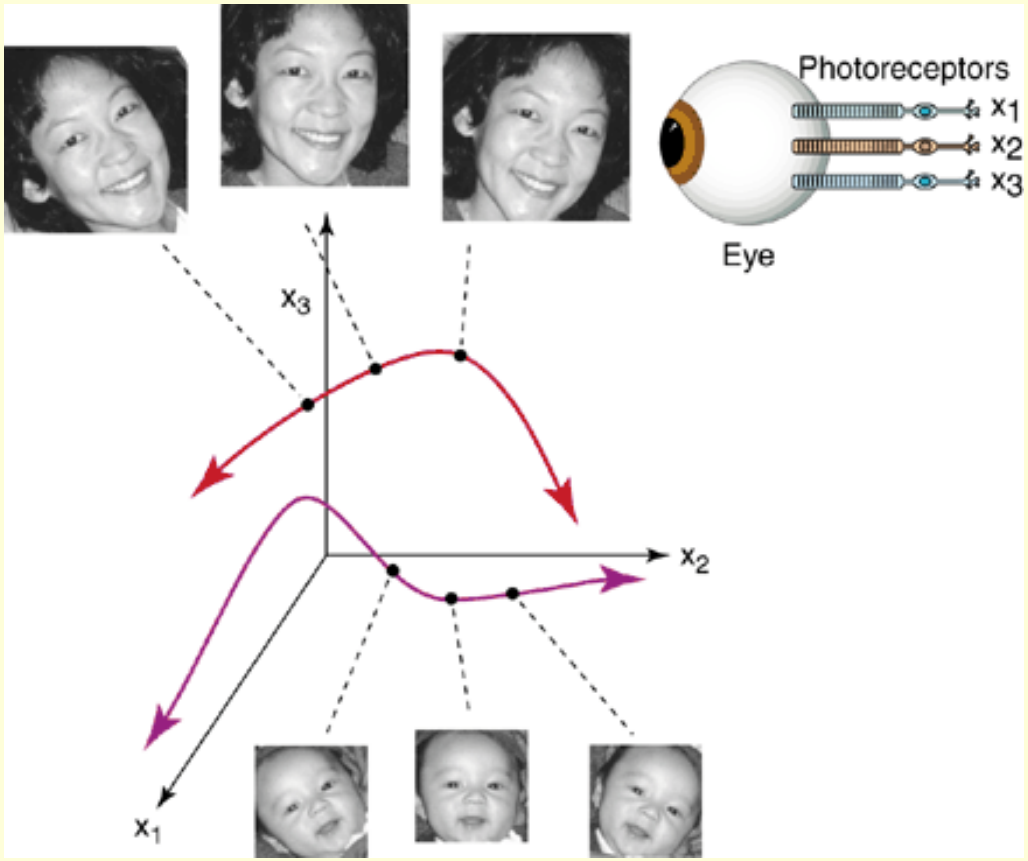
$$X_n = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \dots & \dots & \dots & x_n^p \end{pmatrix}; Y_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \cup \quad X_u = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_u^1 & \dots & \dots & \dots & x_u^p \end{pmatrix}$$



Semisupervised Learning

$$X_n = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \dots & \dots & \dots & x_n^p \end{pmatrix}; Y_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \cup \quad X_u = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_u^1 & \dots & \dots & \dots & x_u^p \end{pmatrix}$$

Manifold Learning



Online Learning

f_0

(x_1, y_1)

f_1

(x_2, y_2)

f_2

$\ldots$

(x_n, y_n)

f_n

$\ldots$

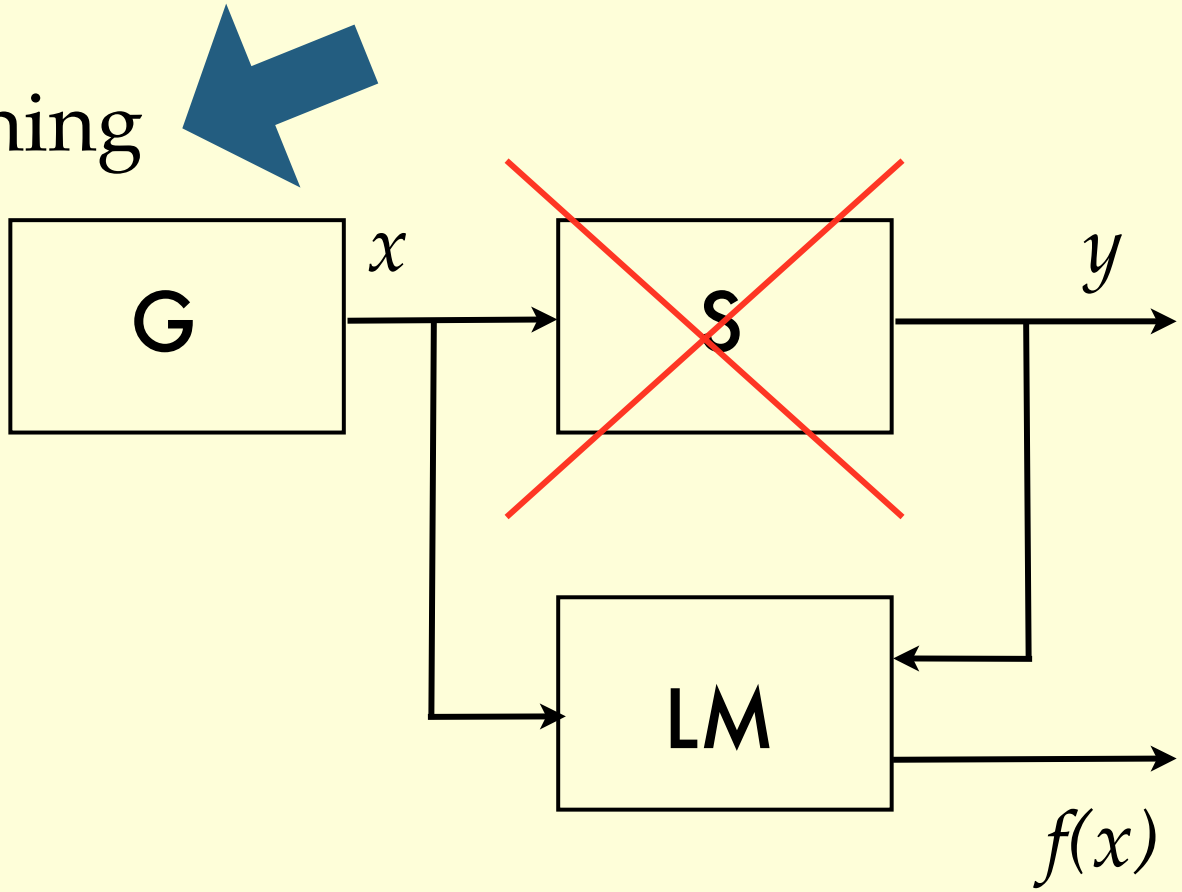
Machine Learning: Problems and Approaches

Learning Problems

- Supervised Learning
- Semisupervised
- Online
- Unsupervised Learning

Learning Approaches

- Batch Learning
- Online
- Active
- ...

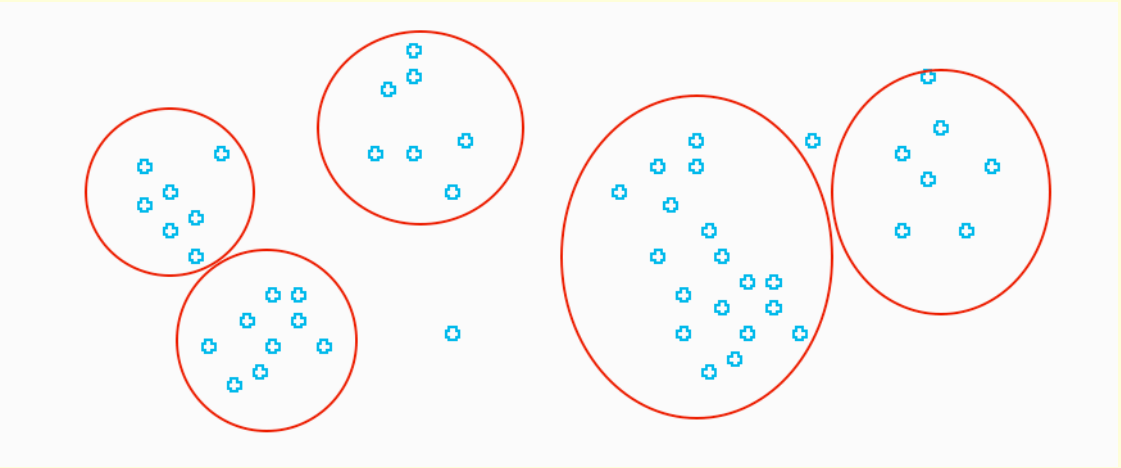


Unsupervised Learning

$$X_n = \begin{pmatrix} x_1^1 & \dots & \dots & \dots & x_1^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & \dots & \dots & \dots & x_n^p \end{pmatrix} \qquad Y_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Given $x_1, \dots, x_n$

Goal: Extract patterns...



Clustering

Dimensionality reduction

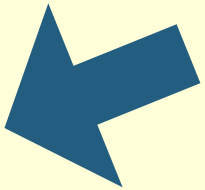
Learning Data Representation

....

Machine Learning: Problems and Approaches

Learning Problems

- Supervised Learning
- Semisupervised
- Online
- ....

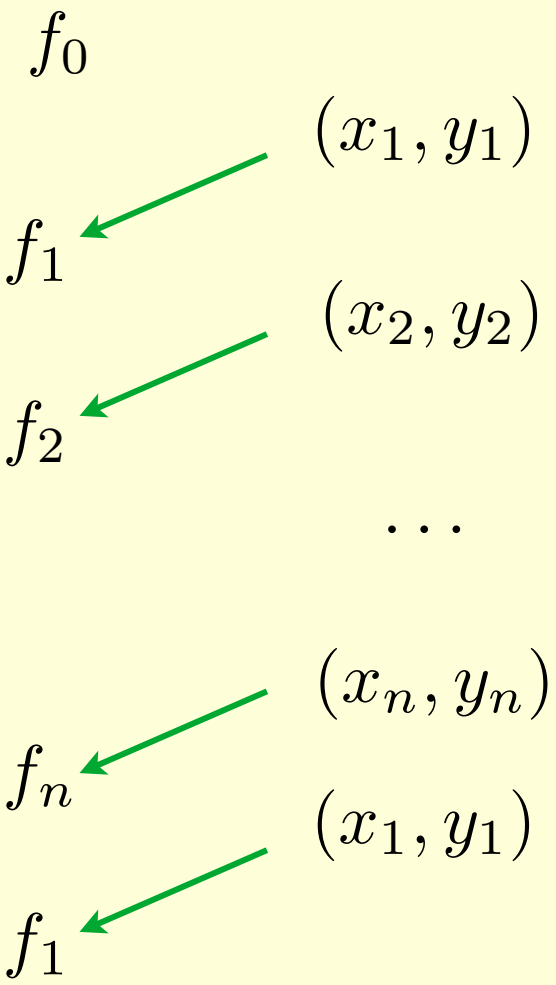


Learning Approaches

- Batch Learning
- Online
- Active
- ...

Online/Incremental Learning

$$(x_1, y_1), \ldots, (x_n, y_n)$$



Active Learning

Learner can query points

f_0

(x_1, y_1)

f_1

(x_2, y_2)

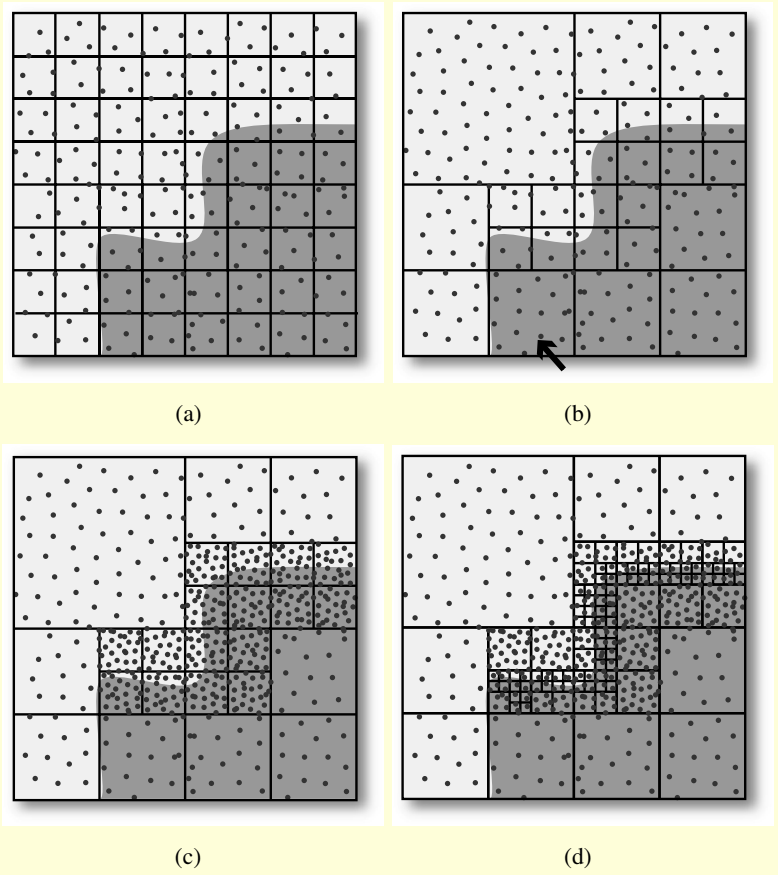
f_2

$\ldots$

(x_n, y_n)

f_n

$\ldots$



(a) (b) (c) (d)

Some Remarks

> ``*Learning is the acquisition of knowledge or skills through study, experience, or being taught*"

We look for computer **systems** that are **trained**, rather than programmed, to perform a task

Learning from examples is a unifying paradigm in AI:

It allows to exploit the availability of ***data*** *and* ***computational resources***

Menu'

Appetizer: AI some context and some history

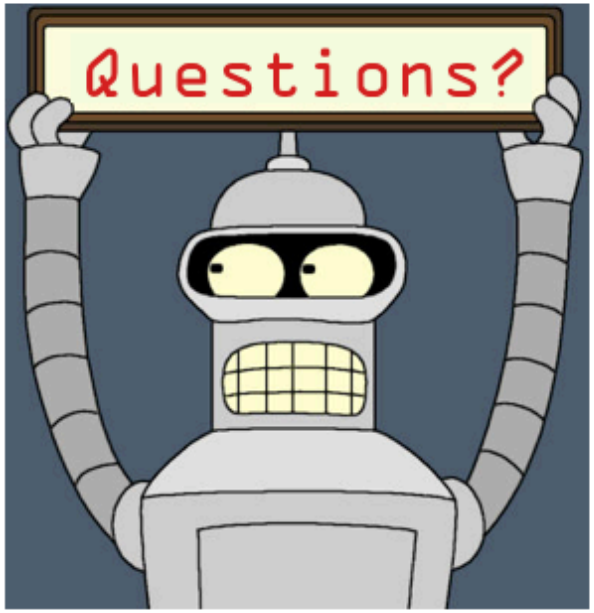
Entree: Machine Learning at a Glance

Main Course: Intro to Statistical Learning Theory

Warning

Math

The course contains many ideas and (quite) a bit of math, questions help prevent sleeping...



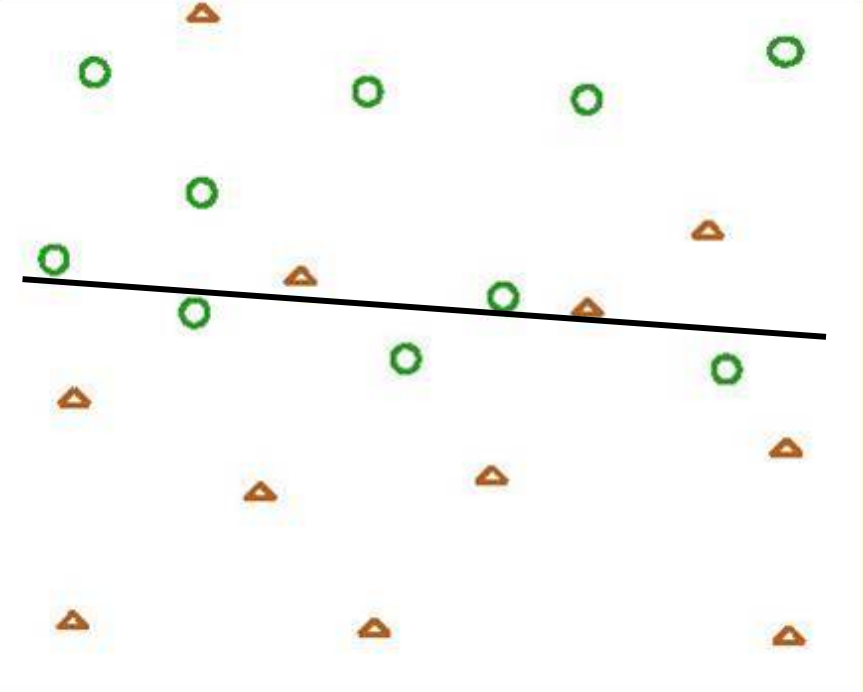
Training Set

Given a Training Set

$$S = (x_1, y_1), \ldots, (x_n, y_n)$$

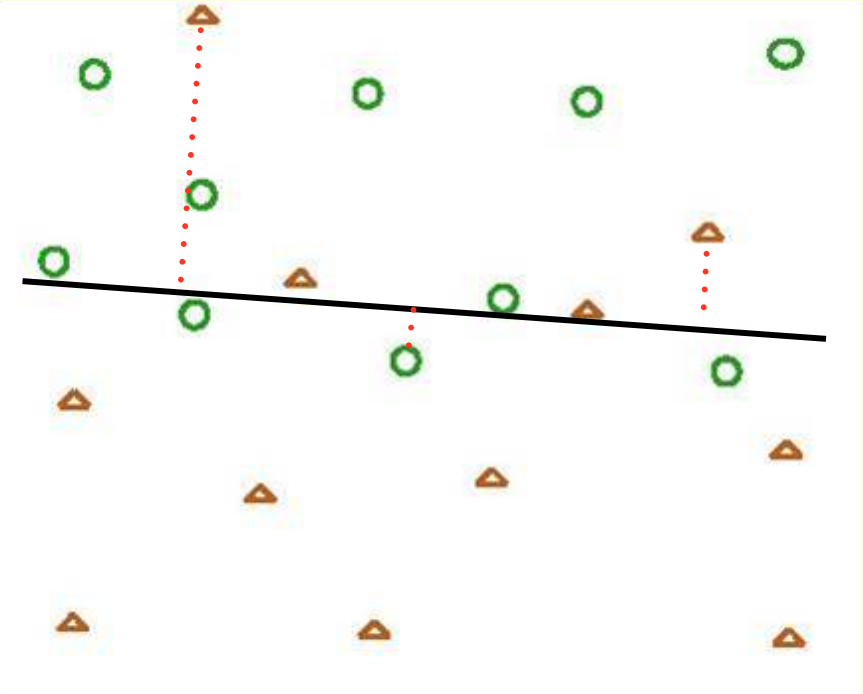
Find

$$f(x) \sim y$$



Loss function

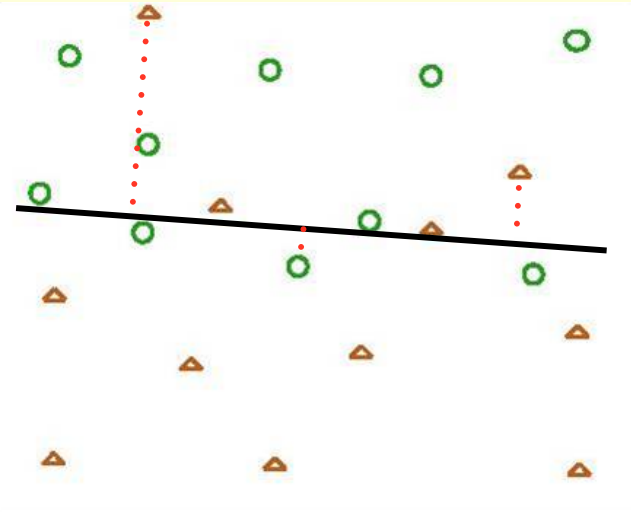
We need a way to measure errors

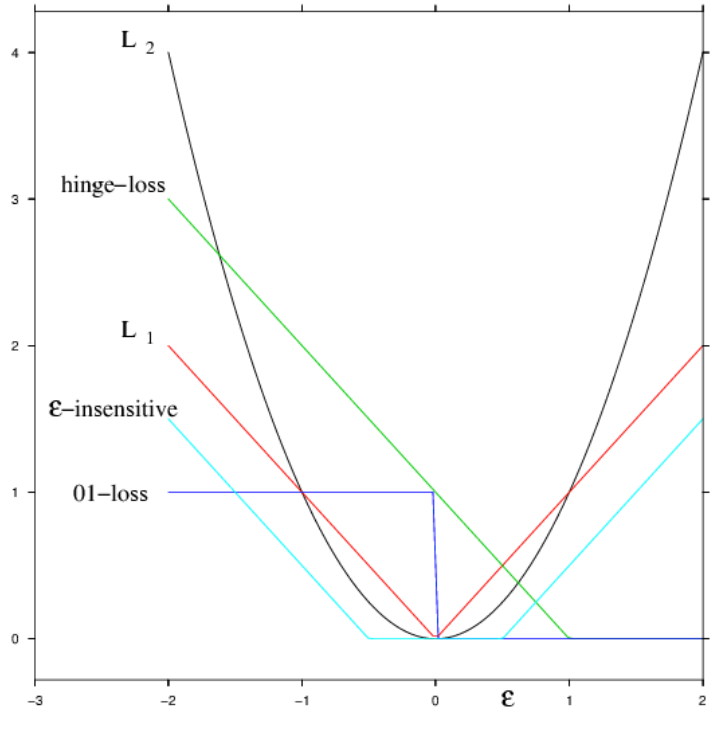


Loss function $V(f(x), y)$

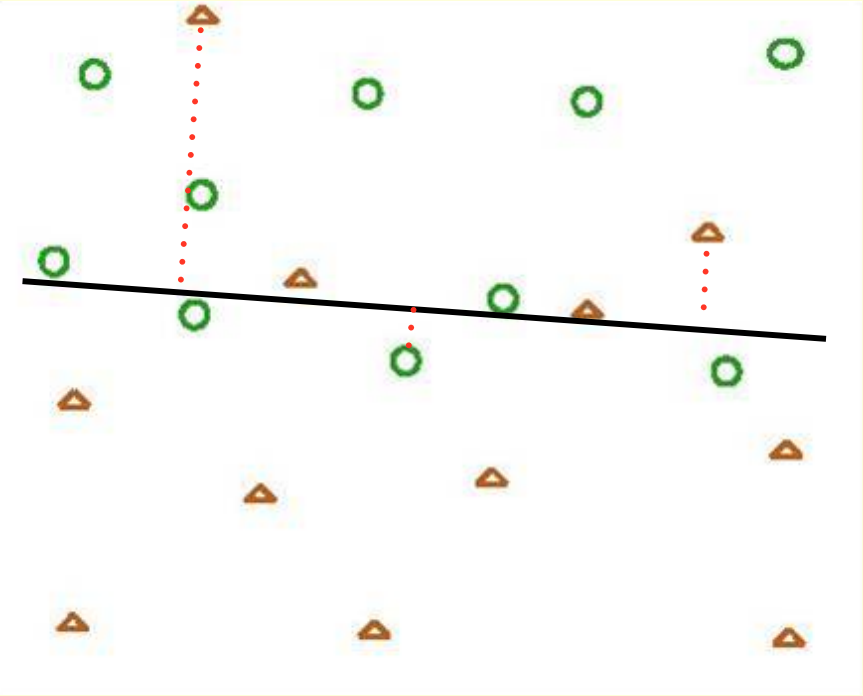
Loss function examples

- $0-1$**-loss** $V(f(x),y) = \theta(-yf(x))$ (θ is the step function)
- **square loss (L2)** $V(f(x),y) = (f(x)-y)^2 = (1-yf(x))^2$
- **absolute value (L1)** $V(f(x),y) = |f(x)-y|$
- Vapnik's ϵ-**insensitive loss** $V(f(x),y) = (|f(x)-y|-\epsilon)_+$
- **hinge loss** $V(f(x),y) = (1-yf(x))_+$
- **logistic loss** $V(f(x),y) = \log(1-e^{-yf(x)})$ logistic regression
- **exponential loss** $V(f(x),y) = e^{-yf(x)}$





Empirical error



Given a loss function $V(f(x), y)$

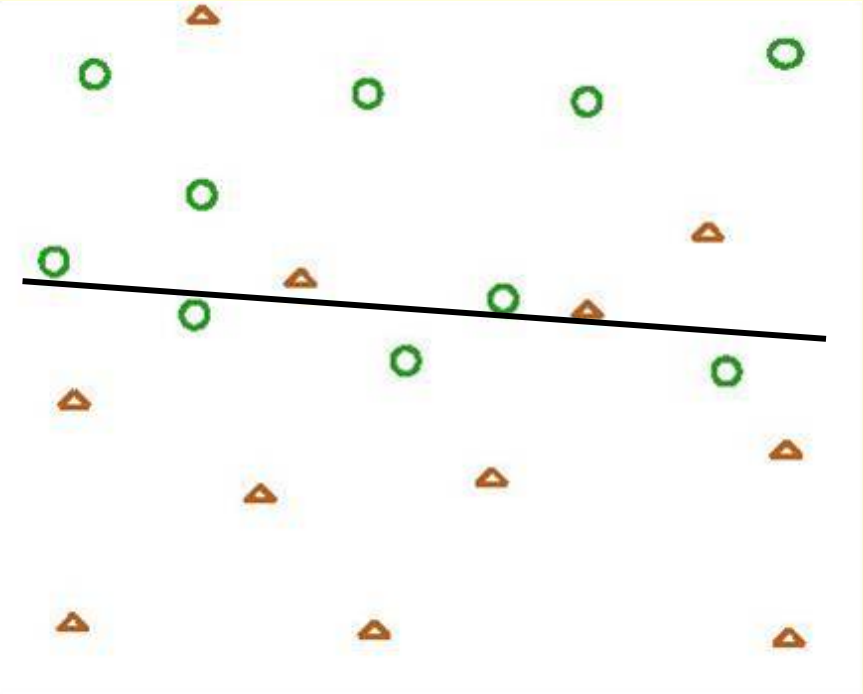
We can define the Empirical Error

$$I_S[f] = \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i)$$

Hypotheses Space

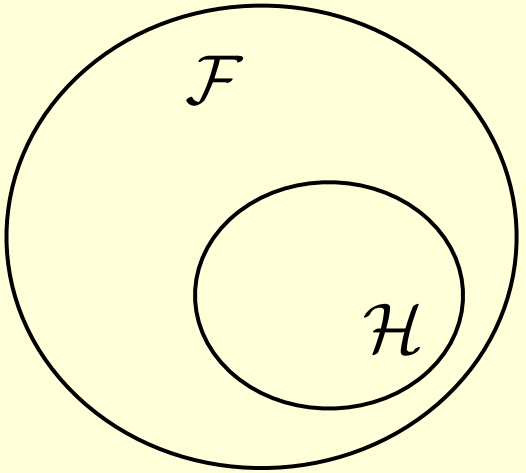
``Learning processes do not take place in vacuum."

Cucker and Smale, AMS 2001

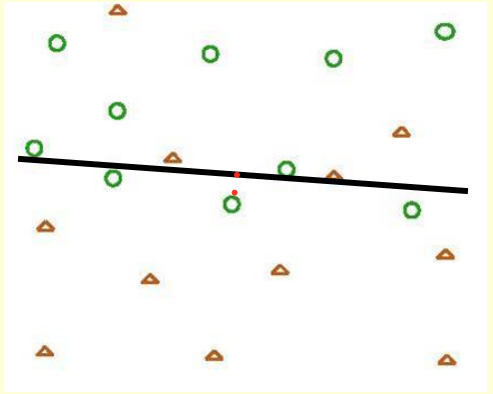


We need to fix a Hypotheses Space

$$\mathcal{H} \subset \mathcal{F} = \{f \mid f : X \to Y\}$$



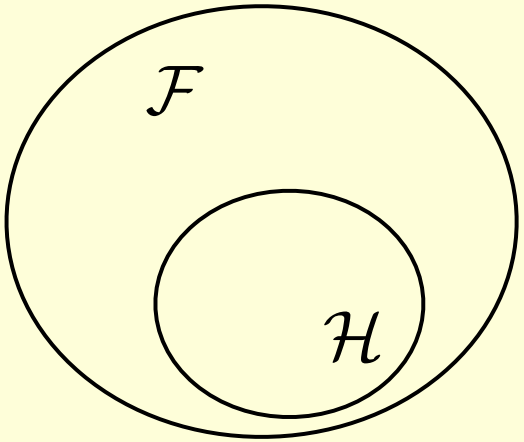
Hypotheses Space



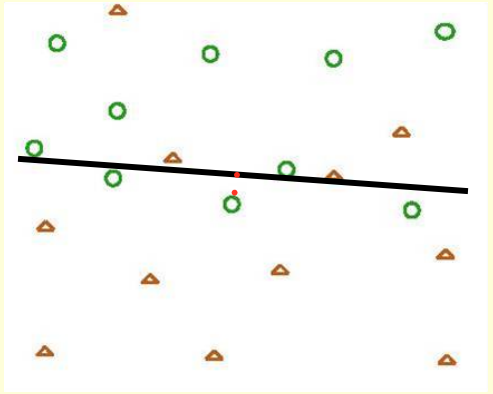
parametric

- **Linear model** $f(x) = \sum_{j=1}^{p} x^j w^j$

$$\mathcal{H} \subset \mathcal{F} = \{f \mid f : X \to Y\}$$



Hypotheses Space



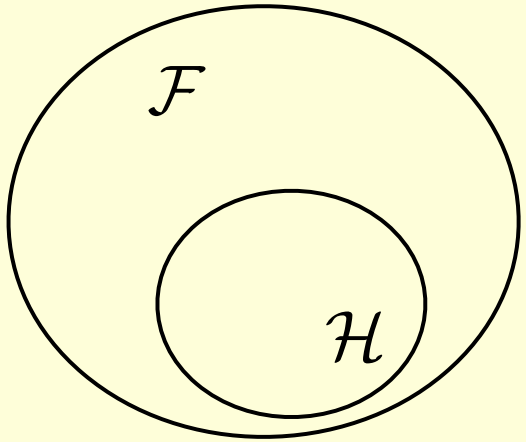
parametric

- **Linear model** $f(x) = \sum_{j=1}^{p} x^j w^j$

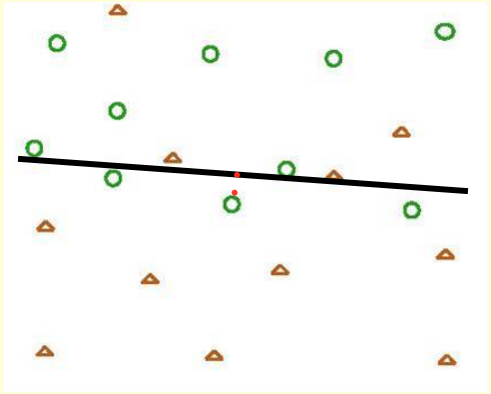
semi-parametric

- **Generalized linear models** $f(x) = \sum_{j=1}^{p} \Phi(x)^j w^j$

$$\mathcal{H} \subset \mathcal{F} = \{f \mid f : X \to Y\}$$



Hypotheses Space



parametric

- **Linear model** $f(x) = \sum_{j=1}^{p} x^j w^j$

semi-parametric

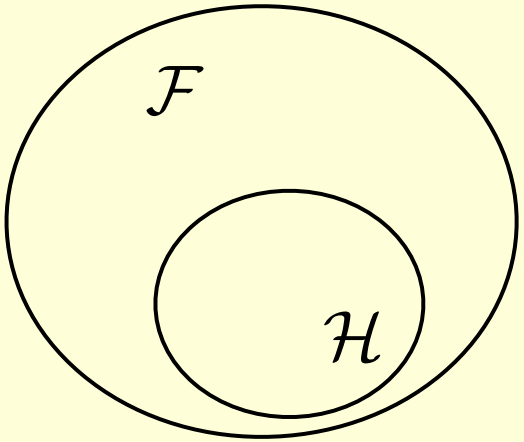
- **Generalized linear models** $f(x) = \sum_{j=1}^{p} \Phi(x)^j w^j$

non-parametric

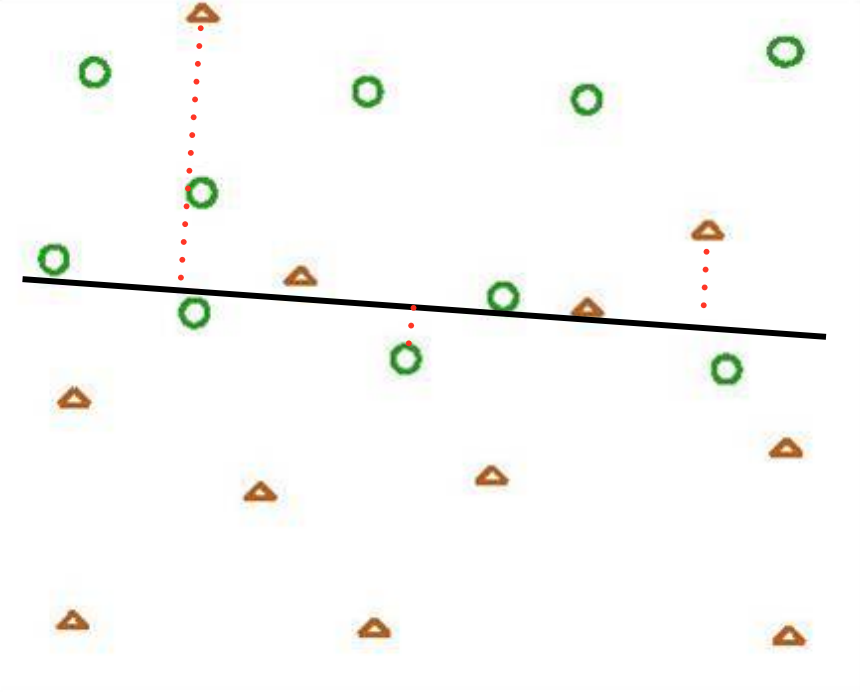
- **Reproducing kernel Hilbert spaces** $f(x) = \sum_{j \geq 1} \Phi(x)^j w^j = \sum_{i \geq 1} K(x, x_i)\alpha_i$

$K(x, x')$ is a symmetric positive definite function called reproducing kernel

$$\mathcal{H} \subset \mathcal{F} = \{f \mid f : X \to Y\}$$



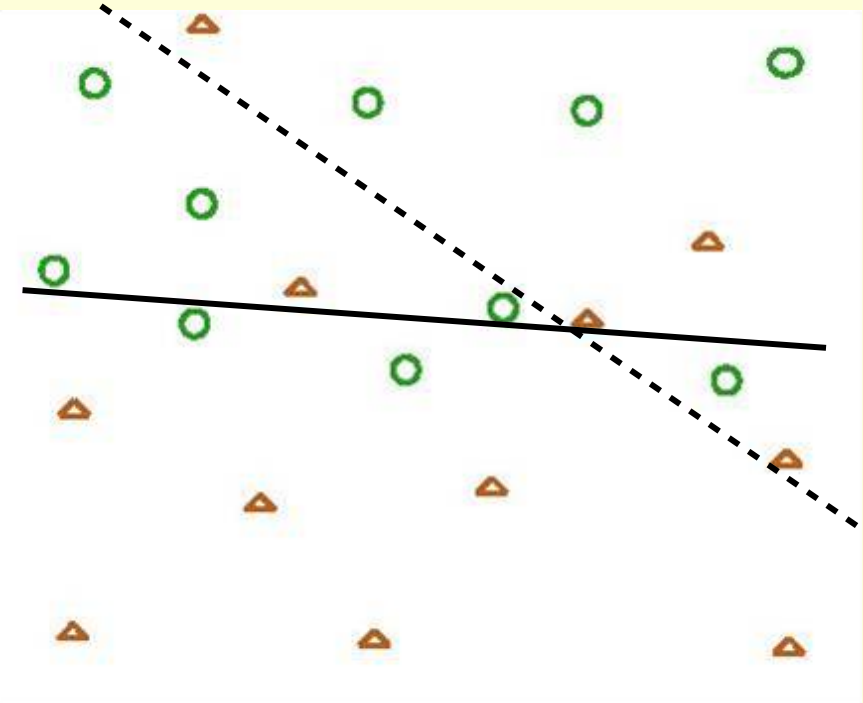
Minimizing the empirical error



Empirical Risk Minimization (ERM)

$$\min_{f \in \mathcal{H}} I_S[f] = \min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i)$$

Minimizing the empirical error

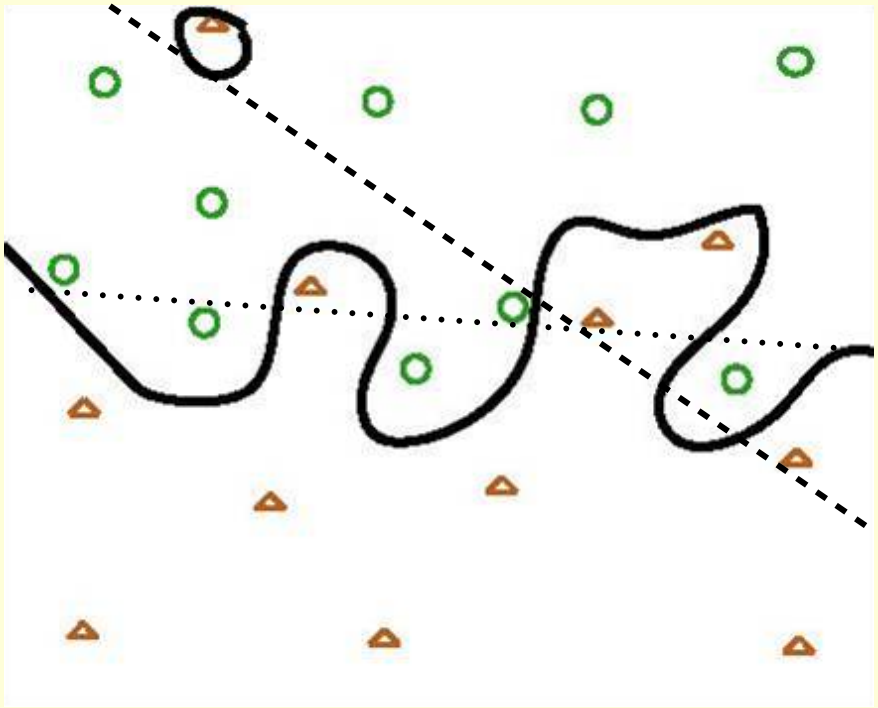


Empirical Risk Minimization (ERM)

$$\min_{f \in \mathcal{H}} I_S[f] = \min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i)$$

Minimizing the empirical error

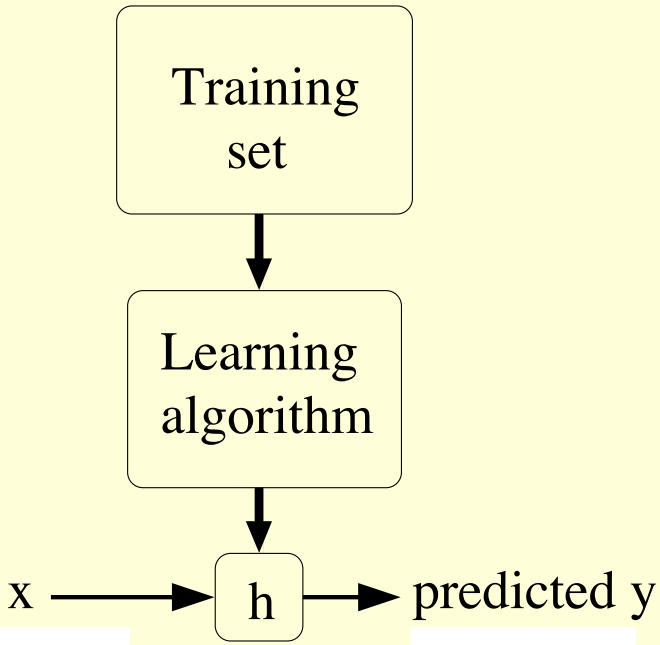
Which one is a good solution?

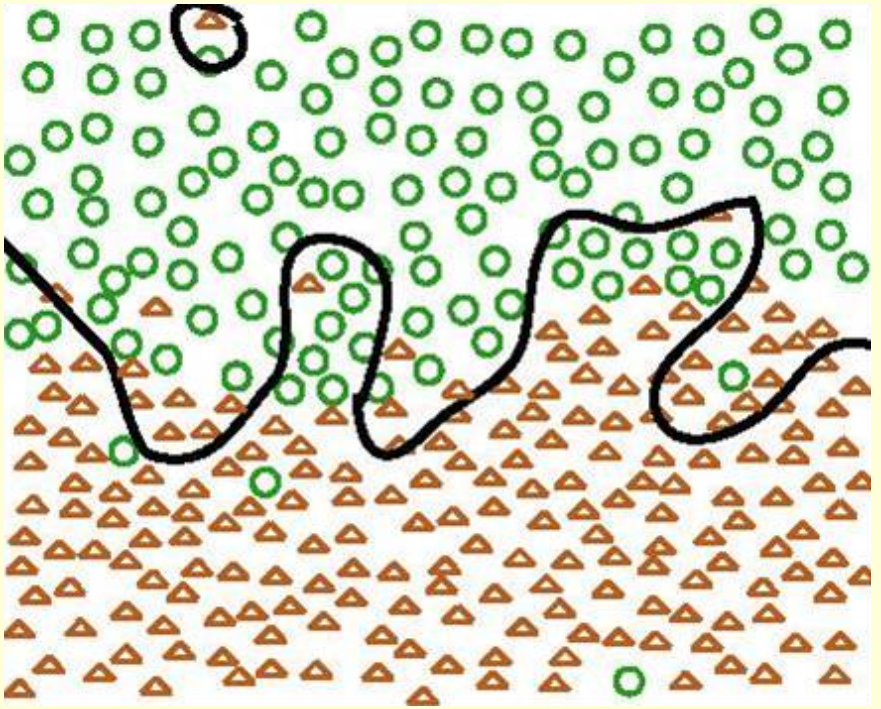


Empirical Risk Minimization (ERM)

$$\min_{f \in \mathcal{H}} \mathcal{E}_S[f] = \min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i)$$

Statistical Learning: Overfitting and Generalization





The training set

$$S = (x_1, y_1), \ldots, (x_n, y_n)$$

is sampled identically and independently (i.i.d) from a fixed unknown probability distribution $p(x, y) = p(x)p(y|x)$

Generalization and Stability

Learning is an ill-posed problem

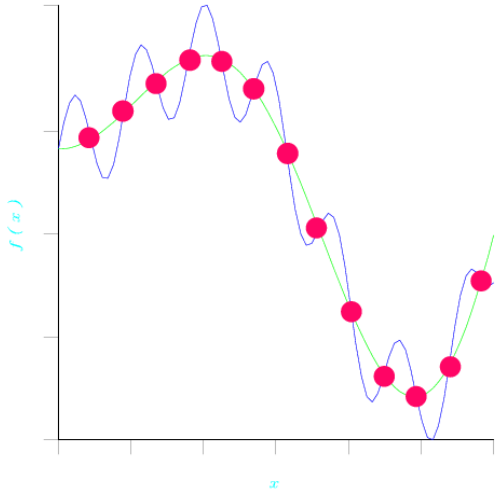
Jacques Hadamard

Ill posed problems often arise if one tries to infer general laws from few data

- the hypothesis space is too large
- there are not enough data

In general ERM leads to ill-posed solutions because

- the solution may be too complex
- it may be not unique
- it may change radically when leaving one sample out



Regularization Theory provides results and techniques to restore well-posedness, that is stability (hence generalization)

Theory of Machine Learning

- Beyond drawings & intuitions (...) there is a deep, **rigorous mathematical foundation** of regularized learning algorithms (Cucker and Smale, Vapnik and Chervonenkis,).

Theory of learning is a synthesis of different fields, e.g. Computer Science (Algorithms, Complexity) and Mathematics (Optimization, Probability, Statistics).

- Central to the Theory of Machine Learning is the problem of understanding condition under which ERM can solve

$$\inf \mathcal{E}(f), \quad \mathcal{E}(f) = \mathbb{E}_{(x,y)} V(y, f(x))$$

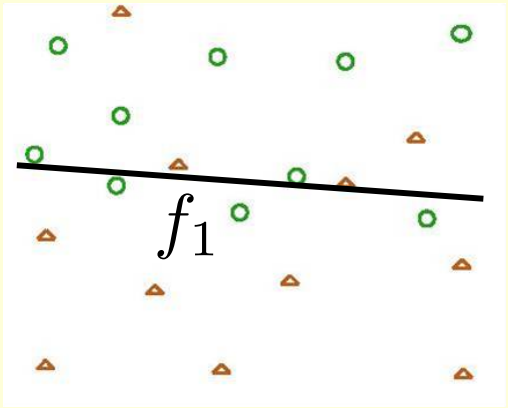
(Tikhonov) Regularization

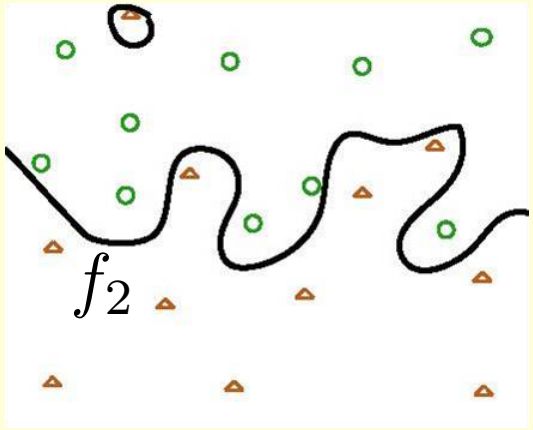
regularization parameter

$$\min_{f \in \mathcal{H}} \{ \frac{1}{n} \sum_{i=1}^{n} V(y_i, f(x_i)) + \lambda R(f) \} \rightarrow f_S^{\lambda}$$

regularizer

- The regularizer describes the *complexity* of the solution





$R(f_2)$ is bigger than $R(f_1)$

- The regularization parameter determines the trade-off between complexity and empirical risk

Some Remarks and Some Questions

- Supervised learning in statistical learning theory: basic concepts/ notation.

- The regularization approach:
  - Which hypotheses space? Which regularizer?
  - How can we find a solution in an efficient way?
  - How do we solve the fitting/regularizing trade-off?