An Introduction to Machine Learning

Fabio A. González Ph.D.

Depto. de Ing. de Sistemas e Industrial
Universidad Nacional de Colombia, Bogotá

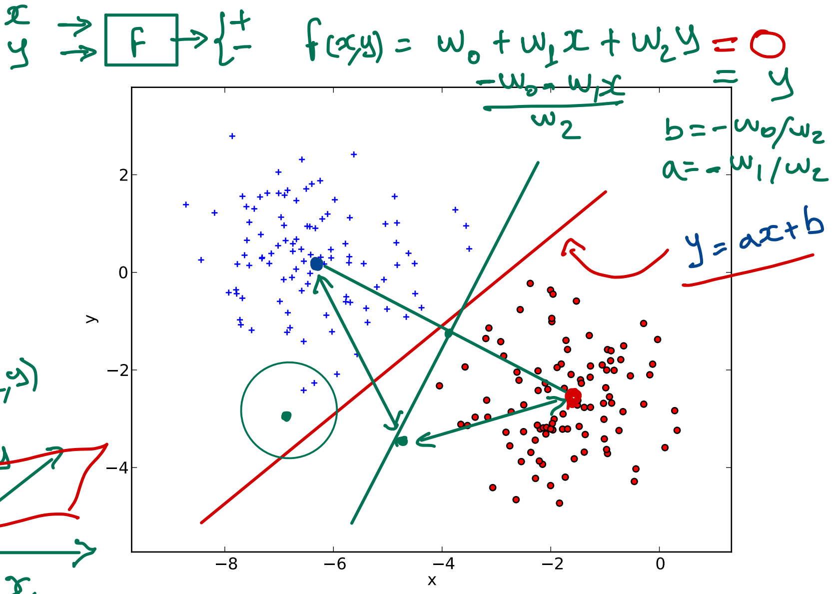
February 25, 2021

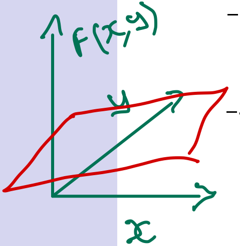
Content

Outline

Outline

Two class classification problem





Outline

How to solve it?

- We need to build a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ such that::

$$\text{Prediction}(x, y) = \begin{cases} C_+ & \text{if } f(x,y) \geq 0 \\ C_- & \text{if } f(x,y) < 0 \end{cases}$$

How to solve it?

- We need to build a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ such that::

$$\text{Prediction}(x, y) = \begin{cases} C_+ & \text{if } f(x, y) \geq 0 \\ C_- & \text{if } f(x, y) < 0 \end{cases}$$

- Training set: $D = \{((x_1, y_1), l_1), \ldots, ((x_n, y_n), l_n)\}$
    - Example: $D = \{((1, 2), -1), ((1, 3), -1), ((3, 1), 1), \ldots\}$

How to solve it?

- We need to ~~build a~~ find function $f:\mathbb{R}^2 \rightarrow \mathbb{R}$ such that::

$$\text{Prediction}(x,y) = \begin{cases} C_+ & \text{if } f(x,y) \geq 0 \\ C_- & \text{if } f(x,y) < 0 \end{cases}$$

- Training set: $D = \{((x_1,y_1),l_1),\ldots,((x_n,y_n),l_n)\}$
  - Example:
    $D = \{((1,2),-1),((1,3),-1),((3,1),1),\ldots\}$
- Loss function:

$$L(f,D) = \sum_{(x_i,y_i,l_i)\in D} \frac{|\text{sign}(f(x_i,y_i)) - l_i|}{2}$$

$\to \begin{cases} 1 & \text{if wrong} \\ 0 & \text{if right} \end{cases}$

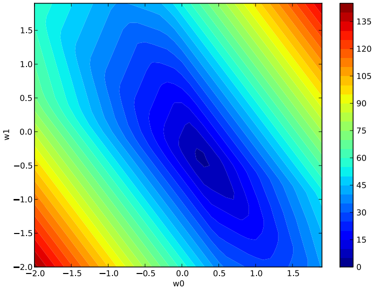
$\text{sign}(x) = \begin{cases} 1 & x>0 \\ 0 & x=0 \\ -1 & x<0 \end{cases}$

L_1 Error loss

$$f(x,y) = w_1 x + w_0 y$$

$$L(f,D) = \frac{1}{2}\sum_{(x_i,y_i,l_i)\in D} |f(x_i,y_i)) -$$

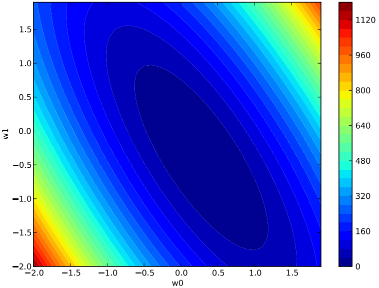
- Are there other alternative loss functions?



Square error loss

$$f(x, y) = w_1 x + w_0 y$$

$$L(f, D) = \frac{1}{2} \sum_{(x_i, y_i, l_i) \in D} (f(x_i, y_i)) - l_i)^2$$



Logistic Regression

$$f(x,y) = \sigma(w_0 + w_1 x + w_2 y) = P(l \mid x, y)$$

$$L(f, D) = -\sum_i \left[l_i \log(f(x_i, y_i)) + (1 - l_i) \log(1 - f(x_i, y_i)) \right]$$

Cross entropy

Learning as optimization

- General optimization problem:

$$\min_{f \in H} L(f, D)$$

Learning as optimization

- General optimization problem:

$$\min_{f \in H} L(f, D)$$

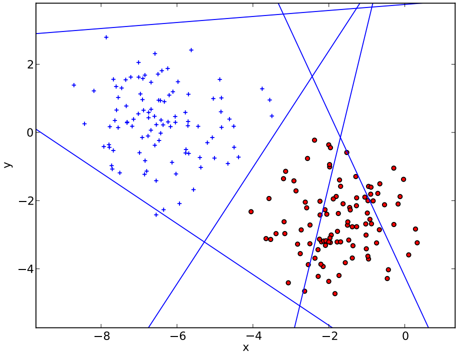
- Two Class 2D classification using linear functions:

Hypothesis space

$$H = \{f : f(x, y) = w_2 x + w_1 y + w_0, \forall w_0, w_1, w_2 \in \mathbb{R}\}$$

$$\min_{f \in H} L(f, D) = \min_{W \in \mathbb{R}^3} \frac{1}{2} \sum_{(x_i, y_i) \in D} (w_2 x_i + w_1 y_i + w_0 - l_i)^2$$

Hypothesis space

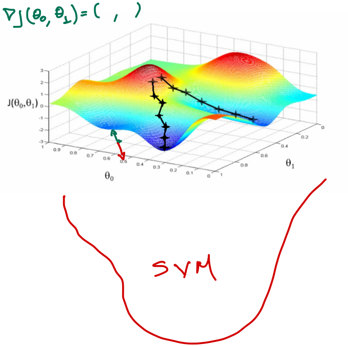


Outline

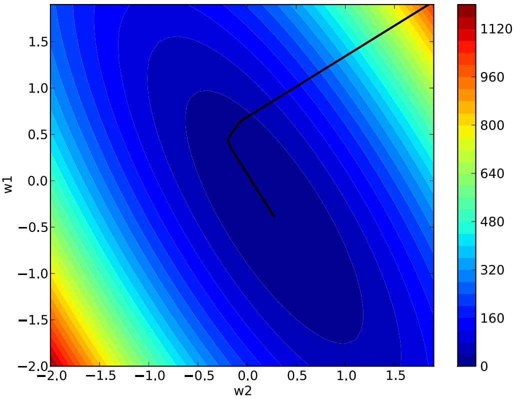
Gradient descent

Iterative optimization of the loss function:

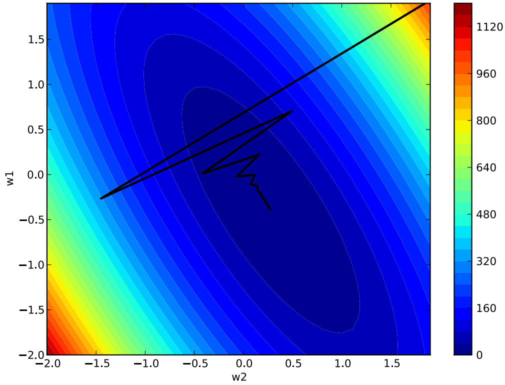
initialize $W^0 = w_0, w_1, w_2$
$k \leftarrow 0$
repeat
  $k \leftarrow k + 1$
  $W^k \leftarrow W^{k-1} - \eta(k)\nabla L(f_{W^{k-1}}, S)$
until $|\eta(k)\nabla L(f_{W^{k-1}}, S)| < \Theta$



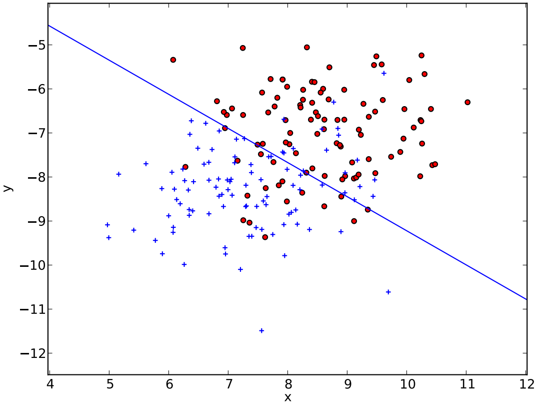
Gradient descent iteration example (1)



Gradient descent iteration example (2)



Non-separable data

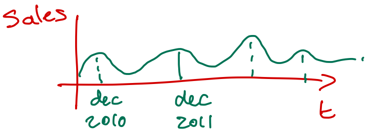


Outline

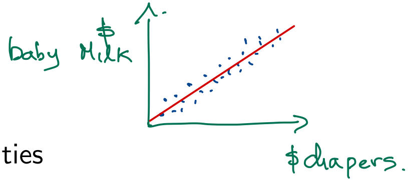
Outline

What is a pattern?

- Data regularities



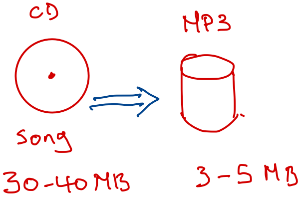
What is a pattern?



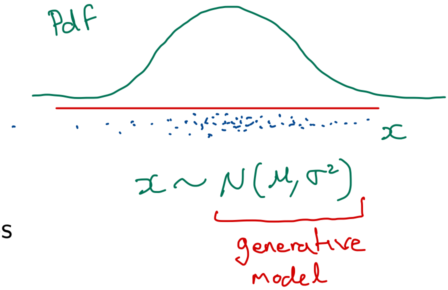
- Data regularities
- Data relationships

What is a pattern?

- Data regularities
- Data relationships
- Redundancy



What is a pattern?



- Data regularities
- Data relationships
- Redundancy
- Generative model

Learning a boolean function

Binary Features

x_1	x_2	f_1	f_2	...	f_{16}
0	0	0	0	...	1
0	1	0	0	...	1
1	0	0	0	...	1
1	1	0	1	...	1

$= 2^4$

- How many Boolean functions of n variables are? 2^{2^n}

Learning a boolean function

x_1	x_2	f_1	f_2	...	f_{16}
0	0	0	0	...	1
0	1	0	0	...	1
1	0	0	0	...	1
1	1	0	1	...	1

- How many Boolean functions of n variables are?
- How many candidate functions are removed by a sample?

Learning a boolean function

x_1	x_2	f_1	f_2	...	f_{16}
0	0	0	0	...	1
0	1	0	0	...	1
1	0	0	0	...	1
1	1	0	1	...	1

- How many Boolean functions of n variables are?
- How many candidate functions are removed by a sample?
- Is it possible to generalize?

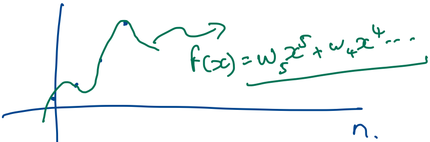
Inductive bias

- In general, the learning problem is *ill-posed* (more than one possible solution for the same particular problem, solutions are sensitive to small changes on the problem)

Inductive bias

n	0	1	2	3	4	5
$f(n)$	1	2	4	8	16	31 (32)

$f(n) = 2^n$



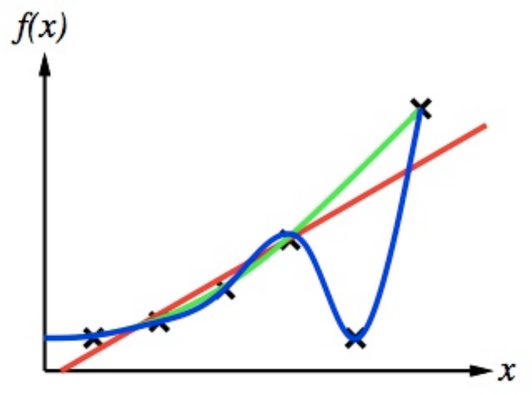
- In general, the learning problem is *ill-posed* (more than one possible solution for the same particular problem, solutions are sensitive to small changes on the problem)
- It is necessary to make additional assumptions about the kind of pattern that we want to learn

Inductive bias

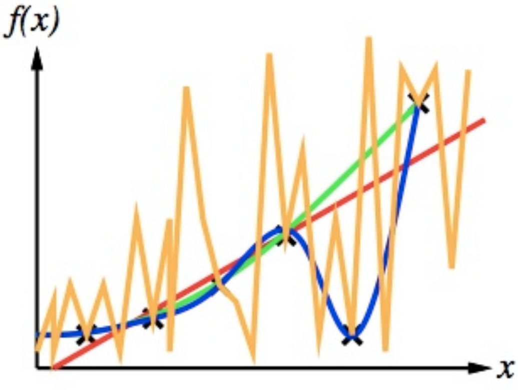
- In general, the learning problem is *ill-posed* (more than one possible solution for the same particular problem, solutions are sensitive to small changes on the problem)
- It is necessary to make additional assumptions about the kind of pattern that we want to learn
- **Hypothesis space**: set of valid patterns that can be learnt by the algorithm

Outline

What is a good pattern?



What is a good pattern?



Occam's razor

from Wikipedia:

Occam's razor (also spelled Ockham's razor) is a principle attributed to the 14th-century English logician and Franciscan friar William of Ockham. The principle states that the explanation of any phenomenon should make as few assumptions as possible, eliminating, or "shaving off", those that make no difference in the observable predictions of the explanatory hypothesis or theory. The principle is often expressed in Latin as the *lex parsimoniae* (law of succinctness or parsimony).

"All things being equal, the simplest solution tends to be the best one."

Outline

Training error vs generalization error

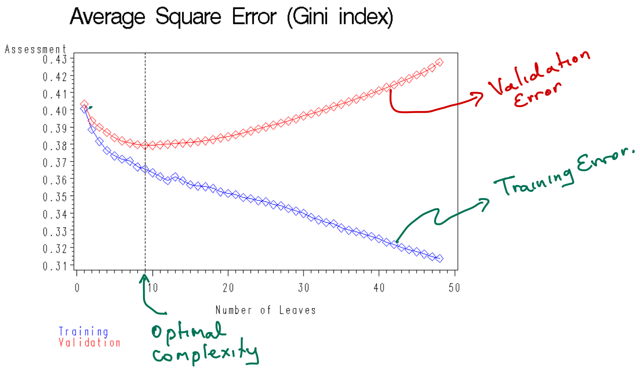
- The loss function measures the error in the training set

Training error vs generalization error

- The loss function measures the error in the training set
- Is this a good measure of the quality of the solution?

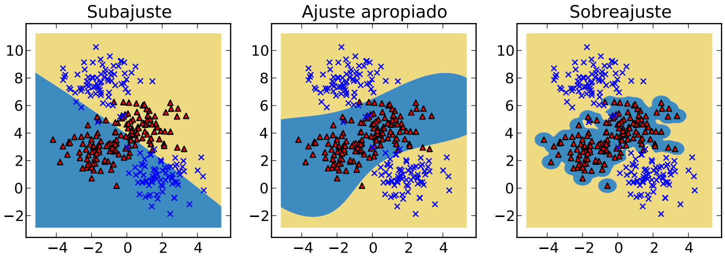
Training error vs generalization error

- The loss function measures the error in the training set
- Is this a good measure of the quality of the solution?



-

Over-fitting and under-fitting



Generalization error

- Generalization error:

$$E[(L(f_w, S)]$$

Generalization error

- Generalization error:

$$E[(L(f_w, S)]$$

- How to control the generalization error during training?

Generalization error

- Generalization error:

$$E[(L(f_w, S)]$$

- How to control the generalization error during training?
  - Cross validation

Generalization error

- Generalization error:

$$E[(L(f_w, S)]$$

- How to control the generalization error during training?
  - Cross validation
  - Regularization

Regularization

- Vapnik, 1995:

Real label

Prediction

Expected Risk

$$R(\alpha) = \int \frac{1}{2}|y - f(\mathbf{x}, \alpha)|dP(\mathbf{x}, y) = \frac{1}{2}\int p(x,y)\,|y - f(x,\alpha)|\,dx$$

real label for sample i

Prediction

Empirical Risk/loss

$$R_{emp}(\alpha) = \frac{1}{2l}\sum_{i=1}^{l}|y_i - f(\mathbf{x}_i, \alpha)|.$$

samples

VC-dimension (measure of complexity)

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\left(\frac{h(\log(2l/h) + 1) - \log(\eta/4)}{l}\right)}$$

Regularized optimization problem

- Optimization problem:

$$\min_{f \in H} L(f, D) + R(f)$$

Regularized optimization problem

- Optimization problem:

$$\min_{f \in H} L(f, D) + R(f)$$

- Two Class 2D classification using linear functions:

$$H = \{f : f(x, y) = w_2 x + w_1 y + w_0, \forall w_0, w_1, w_2 \in \mathbb{R}\}$$

$$\min_{f \in H} L(f, D) = \min_{W \in \mathbb{R}^3} \frac{1}{2} \sum_{(x_i, y_i) \in D} (w_2 x_i + w_1 y_i + w_0 - l_i)^2 + \alpha \| W \|$$

Outline

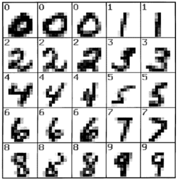
Types

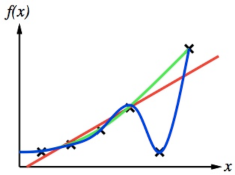
- Supervised learning
- Non-supervised learning
- Semi-supervised learning
- Active/reinforcement learning
- On-line learning

Outline

Supervised learning

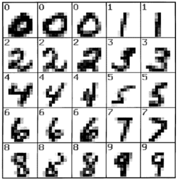
- **Fundamental problem**: to find a function that relates a set of inputs with a set of outputs

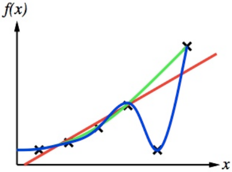




Supervised learning

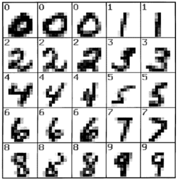
- **Fundamental problem**: to find a function that relates a set of inputs with a set of outputs
- Typical problems:

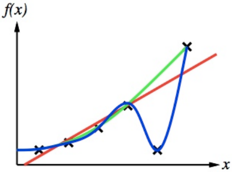




Supervised learning

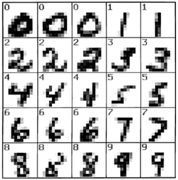
- **Fundamental problem**: to find a function that relates a set of inputs with a set of outputs
- Typical problems:
  - Classification

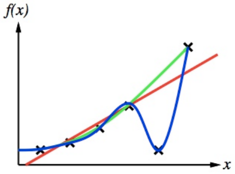




Supervised learning

- **Fundamental problem**: to find a function that relates a set of inputs with a set of outputs
- Typical problems:
  - Classification
  - Regression

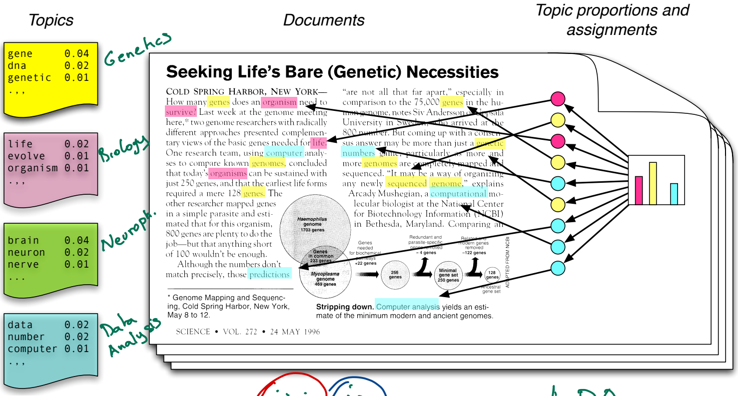


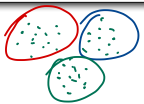


Outline

Non-supervised learning

Latent Topic Analysis





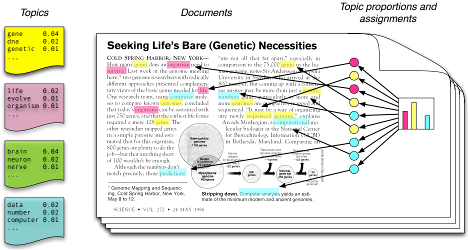
LDA

k-means

⇓

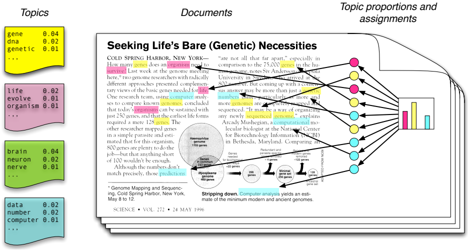
Fuzzy c-means

Non-supervised learning



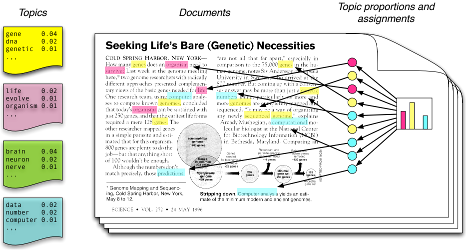
- There are not labels for the training samples

Non-supervised learning



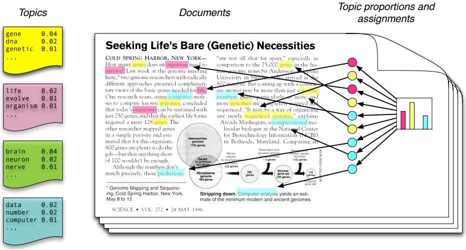
- There are not labels for the training samples
- **Fundamental problem**: to find the subjacent structure of a training data set

Non-supervised learning



- There are not labels for the training samples
- **Fundamental problem**: to find the subjacent structure of a training data set
- Typical problems: clustering, probability density estimation, dimensionality reduction, latent topic analysis, data compression, manifold learning

Non-supervised learning



- There are not labels for the training samples
- **Fundamental problem**: to find the subjacent structure of a training data set
- Typical problems: clustering, probability density estimation, dimensionality reduction, latent topic analysis, data compression
- Some samples may have labels, in that case it is called semi-supervised learning

Outline

Active/reinforcement learning

- Generally, it happens in the context of an agent acting in an environment

https://www.youtube.com/watch?v=iqXKQf2BOSE

Active/reinforcement learning

- Generally, it happens in the context of an agent acting in an environment
- The agent is not told whether it has made the right decision or not

https://www.youtube.com/watch?v=iqXKQf2BOSE

Active/reinforcement learning

- Generally, it happens in the context of an agent acting in an environment
- The agent is not told whether it has made the right decision or not
- The agent is punished or rewarded (not necessarily in an immediate way)

https://www.youtube.com/watch?v=iqXKQf2BOSE

Active/reinforcement learning

- Generally, it happens in the context of an agent acting in an environment
- The agent is not told whether it has made the right decision or not
- The agent is punished or rewarded (not necessarily in an immediate way)
- **Fundamental problem**: to define a policy that allows to maximize the expected positive stimulus (reward)

https://www.youtube.com/watch?v=iqXKQf2BOSE

Outline

On-line learning

- Only one pass through the data

On-line learning

- Only one pass through the data
  - big data volume

On-line learning

- Only one pass through the data
  - big data volume
  - real time

On-line learning

- Only one pass through the data
  - big data volume
  - real time
- It may be supervised or unsupervised

On-line learning

- Only one pass through the data
  - big data volume
  - real time
- It may be supervised or unsupervised
- **Fundamental problem**: to extract the maximum information from data with minimum number of passes

Outline

Representative techniques

- Computational
  - Decision trees
  - Nearest-neighbor classification
  - Graph-based clustering
  - Association rules
- Statistical
  - Multivariate regression
  - Linear discriminant analysis
  - Bayesian decision theory
  - Bayesian networks
  - K-means
- Computational-Statistical
  - SVM
  - AdaBoost
- Bio-inspired
  - Neural networks
  - Genetic algorithms
  - Artificial immune systems

Alpaydin, E. 2010 Introduction to Machine Learning (Adaptive Computation and Machine Learning). The MIT Press. (Chap 1,2)