An Introduction to Machine Learning

Fabio A. González Ph.D.

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

August 10, 2011

Content

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

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?

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

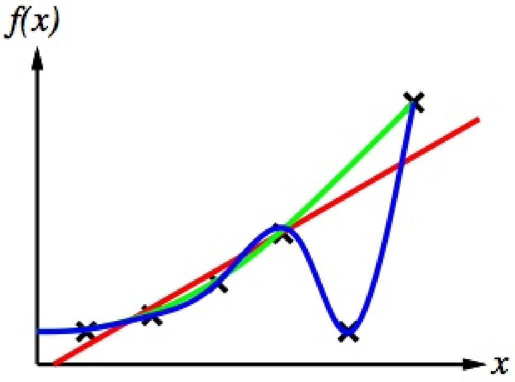
- In general, the learning problem is *ill-posed* (more than one possible solution for the same particular problem, solutions are sensitive to small changes on the problem)
- It is necessary to make additional assumptions about the kind of pattern that we want to learn

Inductive bias

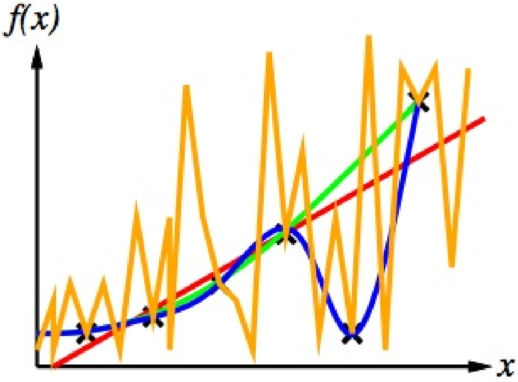
- In general, the learning problem is *ill-posed* (more than one possible solution for the same particular problem, solutions are sensitive to small changes on the problem)
- It is necessary to make additional assumptions about the kind of pattern that we want to learn
- **Hypothesis space**: set of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

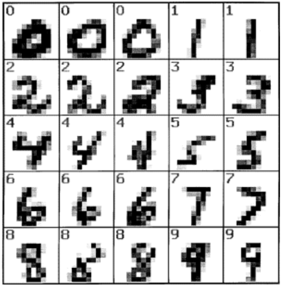
Types

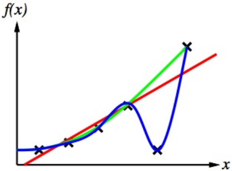
- Supervised learning
- Non-supervised learning
- Semi-supervised learning
- Active learning
- On-line learning

Outline

Supervised learning

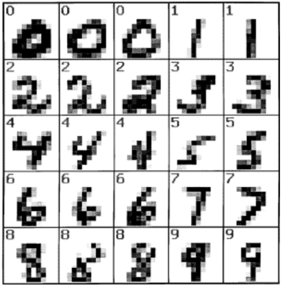
- **Fundamental problem**: to find a function that relates a set of inputs with a set of outputs

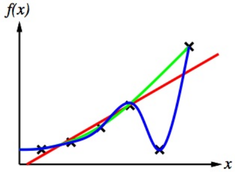




Supervised learning

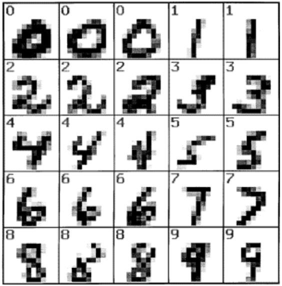
- **Fundamental problem**: to find a function that relates a set of inputs with a set of outputs
- Typical problems:

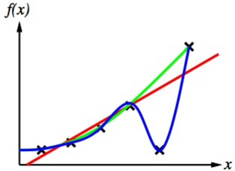




Supervised learning

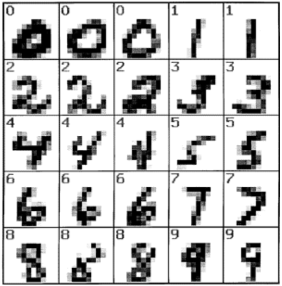
- **Fundamental problem**: to find a function that relates a set of inputs with a set of outputs
- Typical problems:
  - Classification

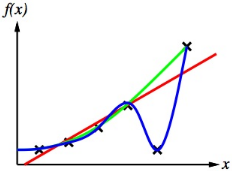




Supervised learning

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

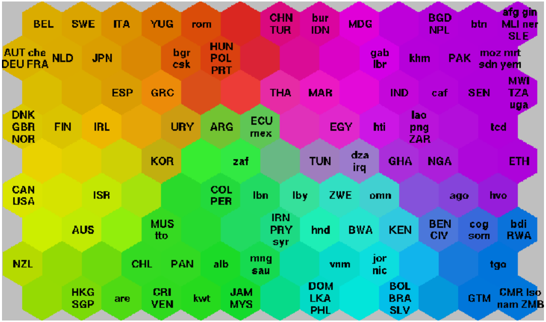




Outline

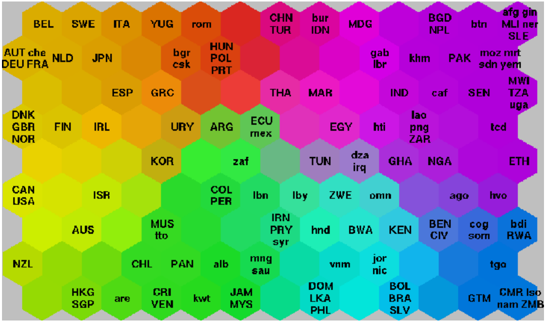
Non-supervised learning

- There are not labels for the training samples



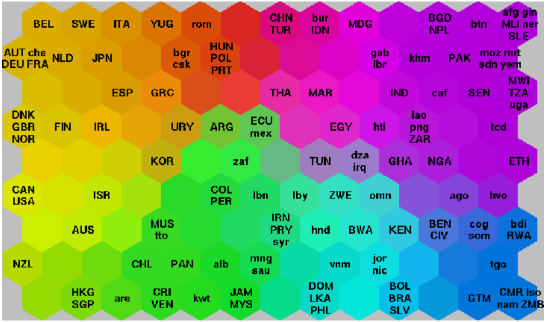
Non-supervised learning

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



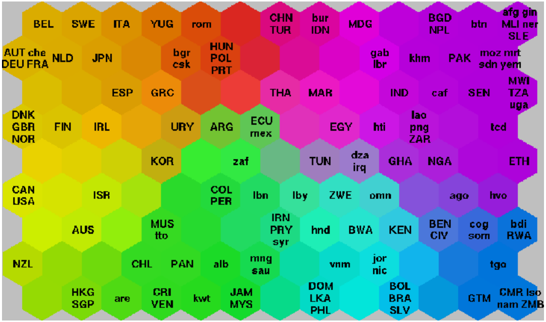
Non-supervised learning

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



Non-supervised learning

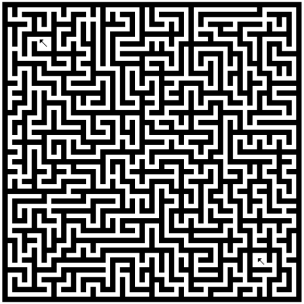
- There are not labels for the training samples
- **Fundamental problem**: to find the subjacent structure of a training data set
- Typical problems: clustering, data compression
- Some samples may have labels, in that case it is called semi-supervised learning



Outline

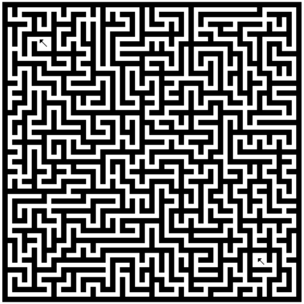
Active/reinforcing learning

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



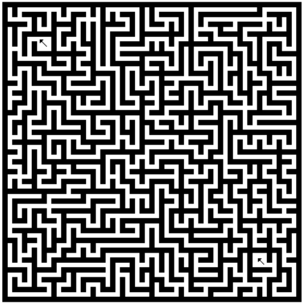
Active/reinforcing learning

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



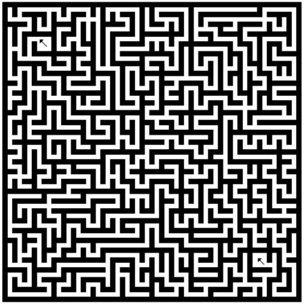
Active/reinforcing learning

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



Active/reinforcing learning

- Generally, it happens in the context of an agent acting in an environment
- The agent is not told whether it has make 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 positive stimulus (reward)



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

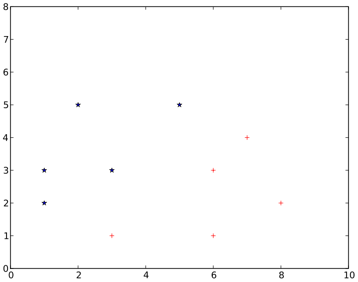
Representative techniques

- 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

Outline

Outline

Two Class Classification Problem



- The idea is to buid a linear classifier function, $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, such that:

$$f(x, y) = \begin{cases} < 0 & \text{if } (x, y) \in C_0 \\ \geq 0 & \text{if } (x, y) \in C_1 \end{cases}$$

Loss Function

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

Loss Function

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

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

Loss Function

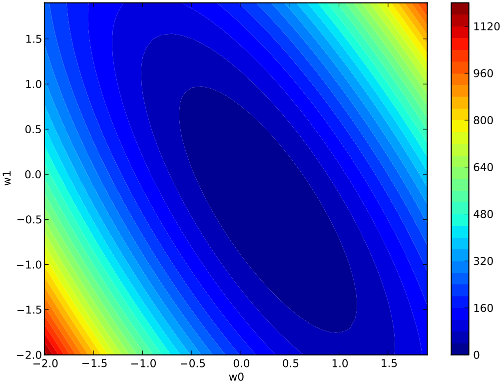
- Training set: $S = \{((x_1, y_1), l_1), \ldots, ((x_n, y_n), l_n)\}$
  - Example: $S = \{((1,2), -1), ((1,3), -1), ((3,1), 1), \ldots\}$
- Loss function:

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

- Are there other alternative loss functions?

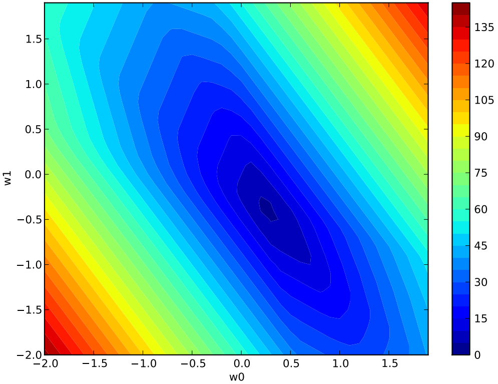
Square Error Loss

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



L_1 Error Loss

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



Learning as Optimization

- General optimization problem:

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

Learning as Optimization

- General optimization problem:

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

- Two Class 2D Classification:

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

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

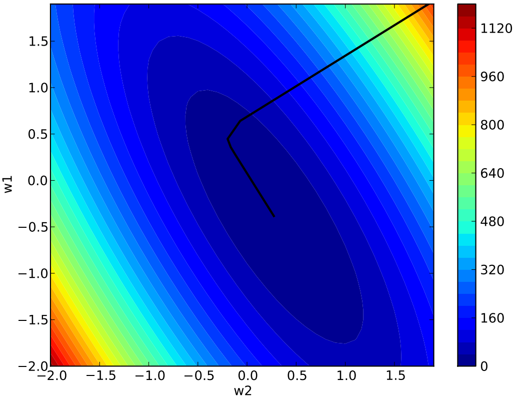
Outline

Gradient Descent

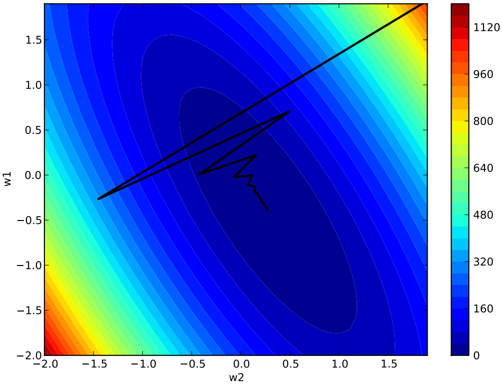
Iterative optimization of the loss function:

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

Gradient Descent Iteration Example (1)



Gradient Descent Iteration Example (2)



Outline

Training Error vs Generalization Error

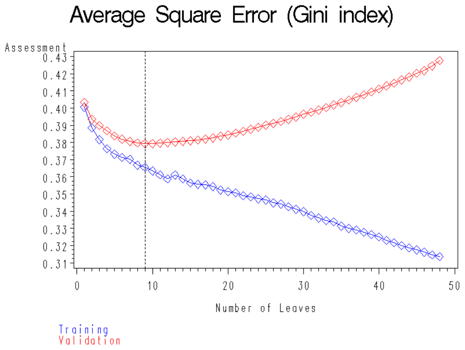
- The loss function measures the error in the training set

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?



-

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:

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

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

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

Alpaydin, E. 2004 Introduction to Machine Learning (Adaptive Computation and Machine Learning). The MIT Press. (Cap 1,2)