An Introduction to Machine Learning

Fabio A. González Ph.D.

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

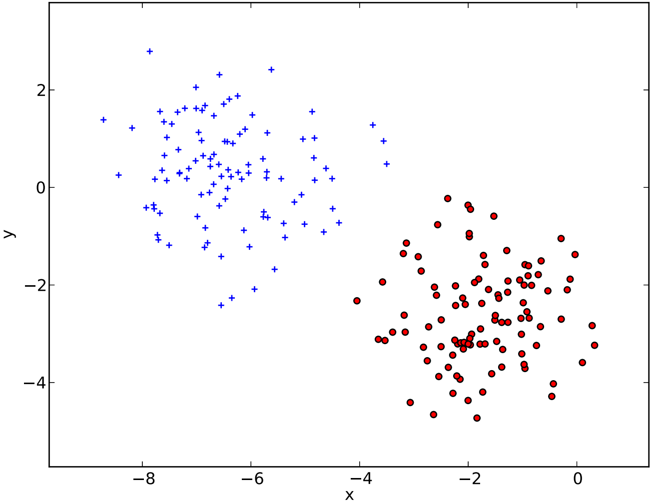
August 15, 2015

Content

Outline

Outline

Two class classification problem



Outline

How to solve it?

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

$$\text{Prediction}(x, y) = \begin{cases} C_1 & \text{si } f(x, y) \geq 0 \\ C_2 & \text{si } f(x, y) < 0 \end{cases}$$

How to solve it?

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

$$\text{Prediction}(x, y) = \begin{cases} C_1 & \text{si } f(x,y) \geq 0 \\ C_2 & \text{si } f(x,y) < 0 \end{cases}$$

- 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 prediction function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ such that::

$$\text{Prediction}(x, y) = \begin{cases} C_1 & \text{si } f(x, y) \geq 0 \\ C_2 & \text{si } f(x, y) < 0 \end{cases}$$

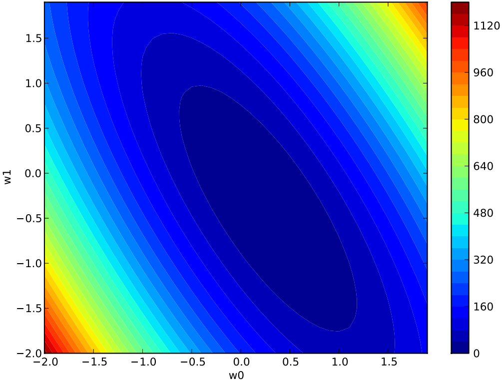
- Training set: $D = \{((x_1, y_1), l_1), \ldots, ((x_n, y_n), l_n)\}$
  - Example: $D = \{((1, 2), -1), ((1, 3), -1), ((3, 1), 1), \ldots\}$
- Loss function:

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

Square error loss

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

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

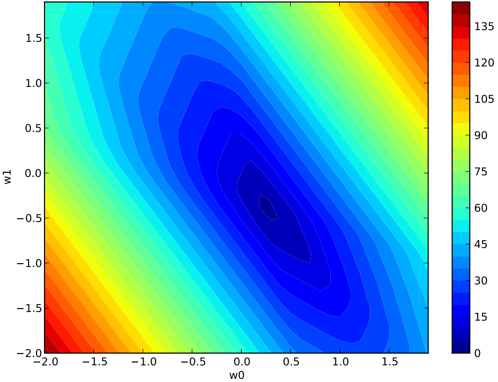


- Are there other alternative loss functions?

L_1 Error loss

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

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



Learning as optimization

- General optimization problem:

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

Learning as optimization

- General optimization problem:

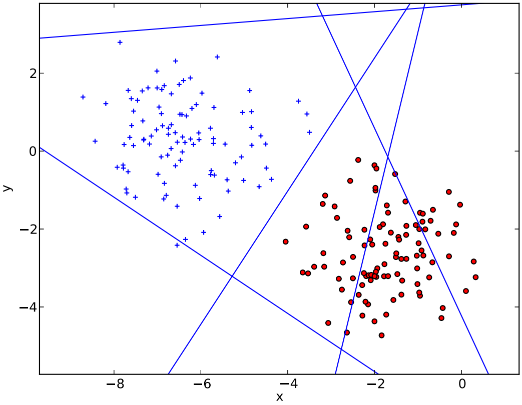
$$\min_{f \in H} L(f, D)$$

- Two Class 2D classification using linear functions:

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

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

Hypothesis space



Outline

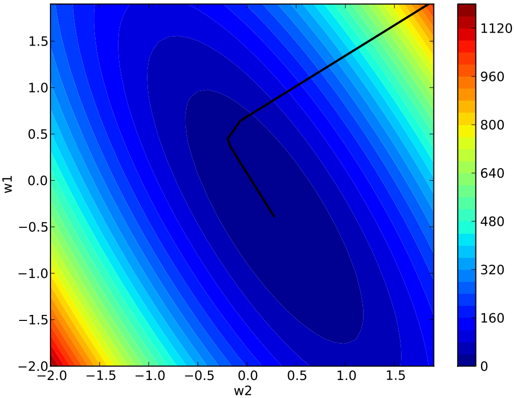
Gradient descent

Iterative optimization of the loss function:

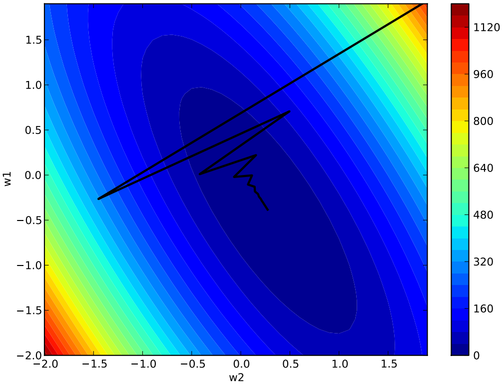
```
initialize W^0 = w_0, w_1, w_2
k ← 0
repeat
    k ← k + 1
    W^k ← W^{k-1} − η(k)∇L(f_{W^{k-1}}, S)
until |η(k)∇L(f_{W^{k-1}}, S)| < Θ
```

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

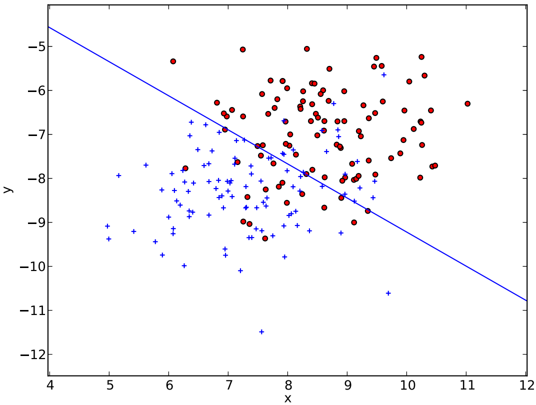
Gradient descent iteration example (1)



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

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

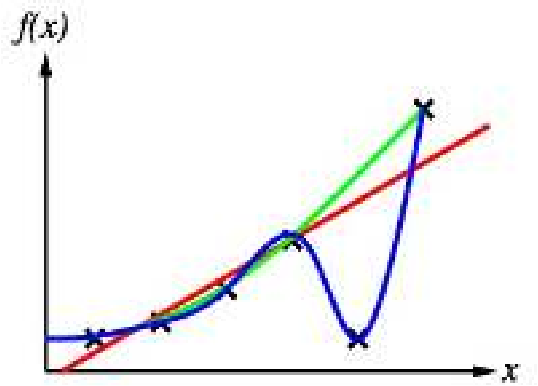
- In general, the learning problem is *ill-posed* (more than one possible solution for the same particular problem, solutions are sensitive to small changes on the problem)
- It is necessary to make additional assumptions about the kind of pattern that we want to learn

Inductive bias

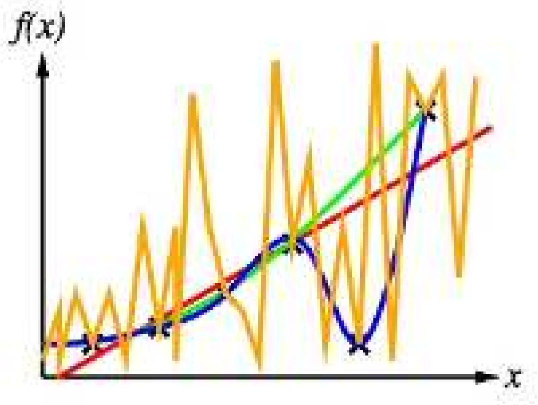
- In general, the learning problem is *ill-posed* (more than one possible solution for the same particular problem, solutions are sensitive to small changes on the problem)
- It is necessary to make additional assumptions about the kind of pattern that we want to learn
- **Hypothesis space**: set of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