

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

1 Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

2 Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

3 Learning Problems

Supervised

Non-supervised

Active

On-line

4 Learning Techniques

1 Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

2 Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

3 Learning Problems

Supervised

Non-supervised

Active

On-line

4 Learning Techniques

1 Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

2 Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

3 Learning Problems

Supervised

Non-supervised

Active

On-line

4 Learning Techniques

Two class classification problem

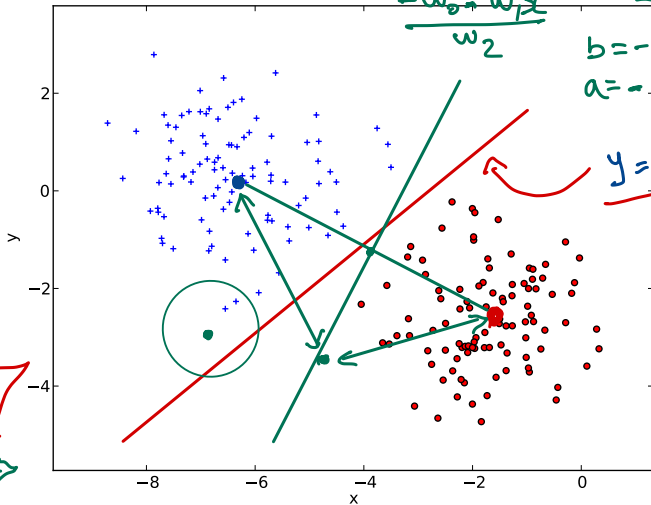
$$\begin{matrix} x \\ y \end{matrix} \Rightarrow \boxed{F} \Rightarrow \begin{Bmatrix} + \\ - \end{Bmatrix} \quad f(x,y) = w_0 + w_1x + w_2y = 0$$

$$\frac{-w_0 - w_1x}{w_2} = y$$

$$b = -w_0/w_2$$

$$a = -w_1/w_2$$

$$y = ax + b$$



Introduction

Example

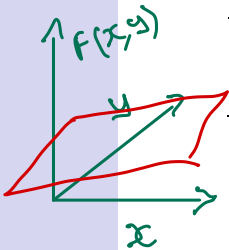
How to State the Learning Problem?

How to Solve the Learning Problem?

Patterns and Generalization

Learning Problems

Learning Techniques



Outline

1 Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

2 Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

3 Learning Problems

Supervised

Non-supervised

Active

On-line

4 Learning Techniques

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

How to solve it?

find

- 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: $D = \{((x_1, y_1), l_1), \dots, ((x_n, y_n), l_n)\}$

- Example:

$$D = \{((1, 2), -1), ((1, 3), -1), ((3, 1), 1), \dots\}$$

- Loss function:

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

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

$\left\{ \begin{array}{l} 1 \text{ if wrong} \\ 0 \text{ if right} \end{array} \right.$

Introduction

Example

How to State the
Learning Problem?

How to Solve the
Learning Problem?

Patterns and Generalization

Learning Problems

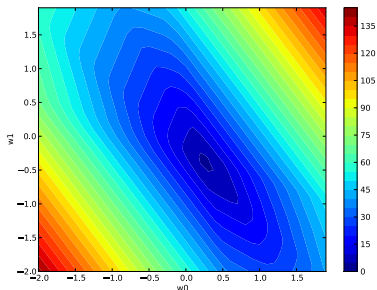
Learning Techniques

L_1 Error loss

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

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

- Are there other alternative loss functions?



Square error loss

Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

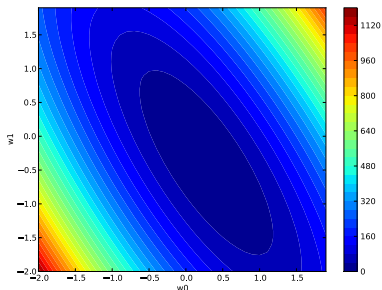
Patterns and Generalization

Learning Problems

Learning Techniques

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

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



Logistic Regression

$$f(x, y) = \sigma(w_0 + w_1 x + w_2 y) = P(\underline{l} | x, y)$$

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

Cross entropy

Learning as optimization

- General optimization problem:

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

Introduction

Example

How to State the
Learning Problem?

How to Solve the
Learning Problem?

Patterns and
Generalization

Learning
Problems

Learning
Techniques

Learning as optimization

Introduction

Example

How to State the
Learning Problem?

How to Solve the
Learning Problem?

Patterns and Generalization

Learning Problems

Learning Techniques

- General optimization problem:

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

- Two Class 2D classification using linear functions:

Hypothesis
space

$$H = \{f : f(x, y) = w_2x + w_1y + w_0, \forall w_0, w_1, w_2 \in \mathbb{R}\}$$

$$\min_{f \in H} L(f, D) = \min_{W \in \mathbb{R}^3} \frac{1}{2} \sum_{(x_i, y_i) \in D} (w_2x_i + w_1y_i + w_0 - l_i)^2$$

Hypothesis space

Introduction

Example

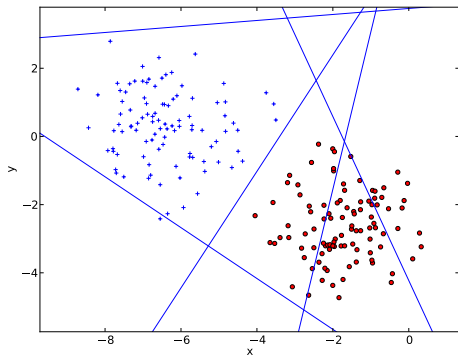
How to State the
Learning Problem?

How to Solve the
Learning Problem?

Patterns and Generalization

Learning Problems

Learning Techniques



Outline

1 Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

2 Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

3 Learning Problems

Supervised

Non-supervised

Active

On-line

4 Learning Techniques

Gradient descent

Iterative optimization of the loss
function:

initialize $W^0 =$

w_0, w_1, w_2

$k \leftarrow 0$

repeat

$k \leftarrow k + 1$

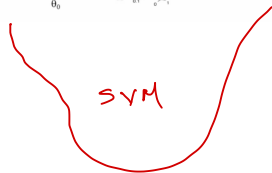
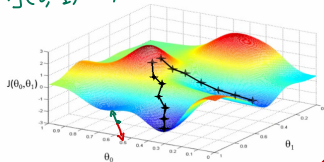
$W^k \leftarrow W^{k-1} -$

$\eta(k) \nabla L(f_{W^{k-1}}, S)$

until $|\eta(k) \nabla L(f_{W^{k-1}}, S)| <$

Θ

$$\nabla J(\theta_0, \theta_1) = (\quad , \quad)$$



Gradient descent iteration example (1)

Introduction

Example

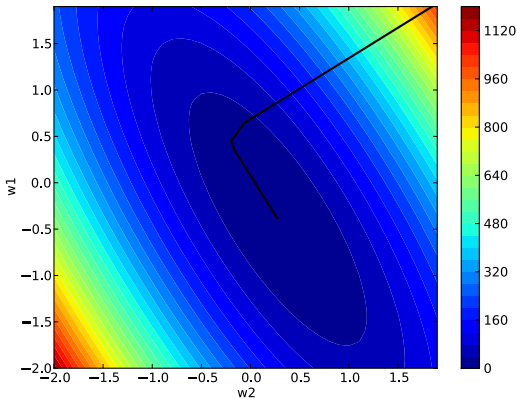
How to State the
Learning Problem?

How to Solve the
Learning Problem?

Patterns and Generalization

Learning Problems

Learning Techniques



Gradient descent iteration example (2)

Introduction

Example

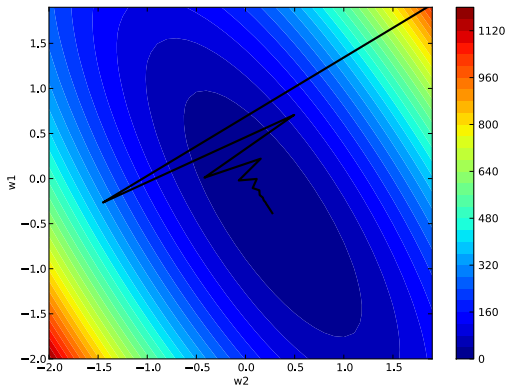
How to State the
Learning Problem?

How to Solve the
Learning Problem?

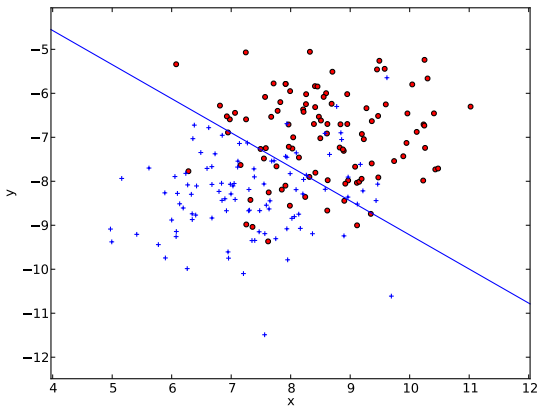
Patterns and Generalization

Learning Problems

Learning Techniques



Non-separable data



Introduction

Example

How to State the
Learning Problem?

How to Solve the
Learning Problem?

Patterns and Generalization

Learning Problems

Learning Techniques

Outline

1 Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

2 Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

3 Learning Problems

Supervised

Non-supervised

Active

On-line

4 Learning Techniques

Outline

1 Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

2 Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

3 Learning Problems

Supervised

Non-supervised

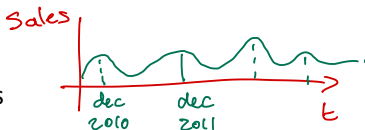
Active

On-line

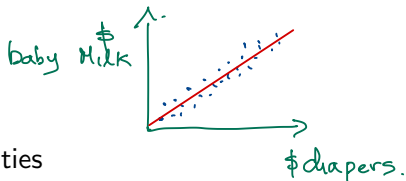
4 Learning Techniques

What is a pattern?

- Data regularities



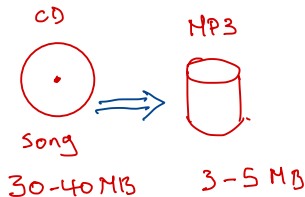
What is a pattern?



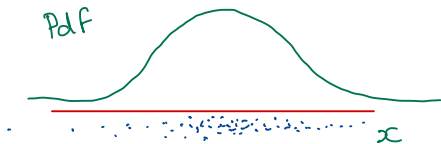
- Data regularities
- Data relationships

What is a pattern?

- Data regularities
- Data relationships
- Redundancy



What is a pattern?



- Data regularities
- Data relationships
- Redundancy
- Generative model

$$x \sim \mathcal{N}(\mu, \sigma^2)$$

generative
model

Learning a boolean function

Binary Features

| x_1 | x_2 | f_1 | f_2 | ... | $f_{16} = 2^4$ |
|-------|-------|-------|-------|-----|----------------|
| 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? 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 |

Annotations: Green checkmarks above f_1 and f_2 in the first row. A red 'X' above f_{16} in the first row. A green circle around the 0 in f_1 for (0,0). A red circle around the 0 in f_1 for (0,1). A blue circle around the 0 in f_1 for (1,0). A red box around the (0,1) row. A blue box around the (1,0) row.

- 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?

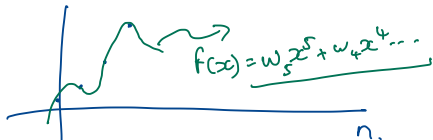
Inductive bias

- 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)

| | | | | | | |
|--------|---|---|---|---|----|-----------|
| n | 0 | 1 | 2 | 3 | 4 | 5 |
| $f(n)$ | 1 | 2 | 4 | 8 | 16 | <u>31</u> |
| | | | | | | 32 |

$f(n) = 2^n$

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

① Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

② Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

③ Learning Problems

Supervised

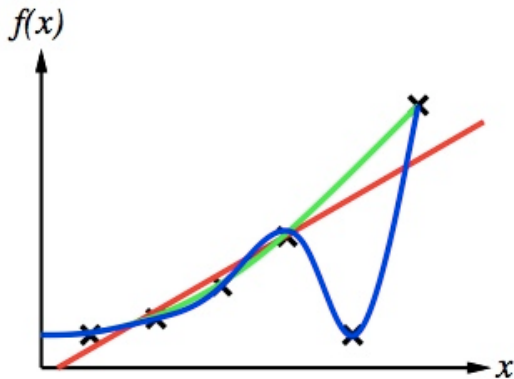
Non-supervised

Active

On-line

④ Learning Techniques

What is a good pattern?



Introduction

Patterns and
Generalization

Generalizing from
patterns

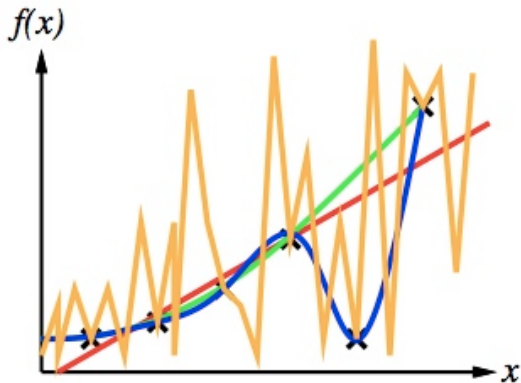
**Overfitting/
Overlearning**

How to Measure the
Quality of a
Solution?

Learning
Problems

Learning
Techniques

What is a good pattern?



Introduction

Patterns and
Generalization

Generalizing from
patterns

**Overfitting/
Overlearning**

How to Measure the
Quality of a
Solution?

Learning
Problems

Learning
Techniques

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

① Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

② Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

③ Learning Problems

Supervised

Non-supervised

Active

On-line

④ Learning Techniques

Training error vs generalization error

- The loss function measures the error in the training set

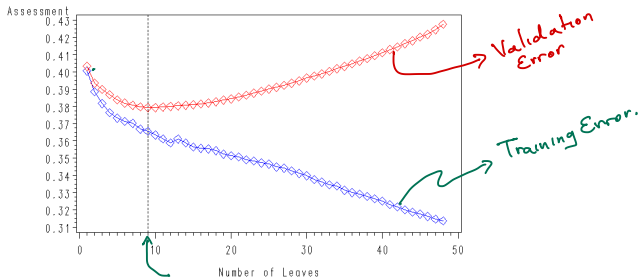
Training error vs generalization error

- 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?

Average Square Error (Gini index)



- Training Validation

optimal complexity

Over-fitting and under-fitting

Introduction

Patterns and Generalization

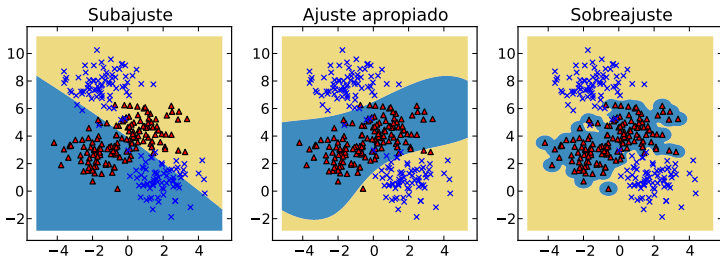
Generalizing from patterns

Overfitting/Overlearning

How to Measure the Quality of a Solution?

Learning Problems

Learning Techniques



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:

Expected Risk

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

Real label
Prediction

Empirical Risk/Loss

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

samples
real label for sample i
Prediction
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:

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

Introduction

Patterns and
Generalization

Generalizing from
patterns

Overfitting/
Overlearning

How to Measure the
Quality of a
Solution?

Learning
Problems

Learning
Techniques

Regularized optimization problem

Introduction

Patterns and
Generalization

Generalizing from
patterns

Overfitting/
Overlearning

How to Measure the
Quality of a
Solution?

Learning
Problems

Learning
Techniques

- Optimization problem:

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

- Two Class 2D classification using linear functions:

$$H = \{f : f(x, y) = w_2x + w_1y + w_0, \forall w_0, w_1, w_2 \in \mathbb{R}\}$$

$$\min_{f \in H} L(f, D) = \min_{W \in \mathbb{R}^3} \frac{1}{2} \sum_{(x_i, y_i) \in D} (w_2x_i + w_1y_i + w_0 - l_i)^2 + \alpha \|W\|$$

Outline

1 Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

2 Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

3 Learning Problems

Supervised

Non-supervised

Active

On-line

4 Learning Techniques

Types

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

Outline

① Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

② Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

③ Learning Problems

Supervised

Non-supervised

Active

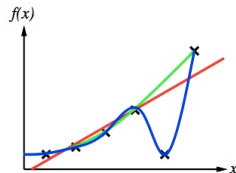
On-line

④ Learning Techniques

Supervised learning

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

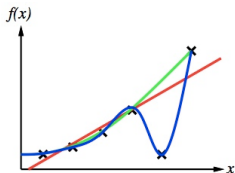
| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 2 | 2 | 2 | 3 | 3 |
| 2 | 2 | 2 | 3 | 3 |
| 4 | 4 | 4 | 5 | 5 |
| 4 | 4 | 4 | 5 | 5 |
| 6 | 6 | 6 | 7 | 7 |
| 6 | 6 | 6 | 7 | 7 |
| 8 | 8 | 8 | 9 | 9 |
| 8 | 8 | 8 | 9 | 9 |



Supervised learning

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

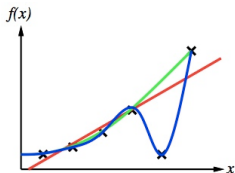
| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 2 | 2 | 2 | 3 | 3 |
| 4 | 4 | 4 | 5 | 5 |
| 6 | 6 | 6 | 7 | 7 |
| 8 | 8 | 8 | 9 | 9 |



Supervised learning

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

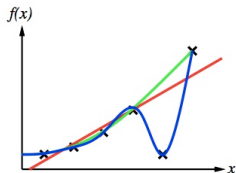
| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 2 | 2 | 2 | 3 | 3 |
| 4 | 4 | 4 | 5 | 5 |
| 6 | 6 | 6 | 7 | 7 |
| 8 | 8 | 8 | 9 | 9 |



Supervised learning

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

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 2 | 2 | 2 | 3 | 3 |
| 4 | 4 | 4 | 5 | 5 |
| 6 | 6 | 6 | 7 | 7 |
| 8 | 8 | 8 | 9 | 9 |



Outline

1 Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

2 Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

3 Learning Problems

Supervised

Non-supervised

Active

On-line

4 Learning Techniques

Non-supervised learning

Latent Topic Analysis

- Introduction
- Patterns and Generalization
- Learning Problems
 - Supervised
 - Non-supervised
 - Active
 - On-line
- Learning Techniques

Topics

| | |
|---------|------|
| gene | 0.04 |
| dna | 0.02 |
| genetic | 0.01 |
| ... | |

Genetics

| | |
|----------|------|
| life | 0.02 |
| evolve | 0.01 |
| organism | 0.01 |
| ... | |

Biology

| | |
|--------|------|
| brain | 0.04 |
| neuron | 0.02 |
| nerve | 0.01 |
| ... | |

Neuropt.

| | |
|----------|------|
| data | 0.02 |
| number | 0.02 |
| computer | 0.01 |
| ... | |

Data Analysis

Documents

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many **genes** does an **organism** need to **survive**? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for **life**. One research team, using **computer** analyses to compare known **genomes**, concluded that today's **organisms** can be sustained with just 250 genes, and that the earliest life forms required a mere 128 **genes**. The other researcher mapped genes in a simple parasite and estimated that for this organism, 802 genes are plenty to do the job—but that anything short of 100 wouldn't be enough. Although the numbers don't match precisely, those **predictions** are not all that far apart," especially in comparison to the 75,000 **genes** in the human genome, notes Siv Andersson, a postdoctoral fellow at the University in Sweden. "It arrived at the 802 number. But coming up with a concrete answer may be more than just a **number**. Some, particularly **more** and **more** **genes** are **needed** to **survive** and **sequenced** **genomes**," explains Arcady Mushegian, a **computational** molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comprising an

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

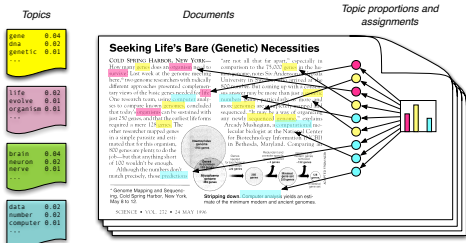
Topic proportions and assignments

k-means
 ↓
 Fuzzy c-means

LDA

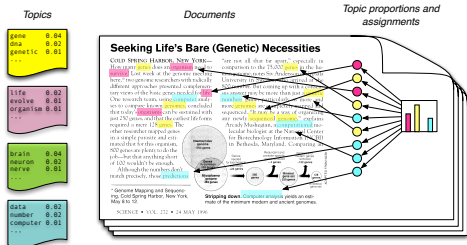
Navigation icons: back, forward, search, etc.

Non-supervised learning



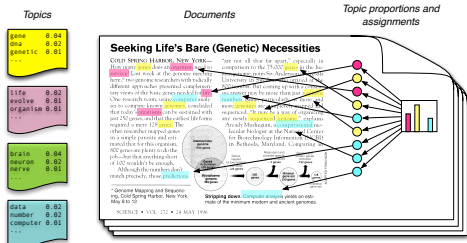
- There are not labels for the training samples

Non-supervised learning



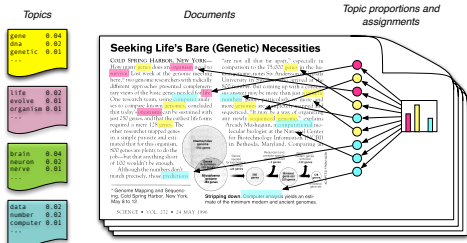
- There are not labels for the training samples
- **Fundamental problem:** to find the subjacent structure of a training data set

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

1 Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

2 Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

3 Learning Problems

Supervised

Non-supervised

Active

On-line

4 Learning Techniques

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

① Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

② Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

③ Learning Problems

Supervised

Non-supervised

Active

On-line

④ Learning Techniques

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

1 Introduction

Example

How to State the Learning Problem?

How to Solve the Learning Problem?

2 Patterns and Generalization

Generalizing from patterns

Overfitting/ Overlearning

How to Measure the Quality of a Solution?

3 Learning Problems

Supervised

Non-supervised

Active

On-line

4 Learning Techniques

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)

Introduction

Patterns and
Generalization

Learning
Problems

**Learning
Techniques**