Chapter 1
Basics: Introduction to Pattern Recognition and to structured data

Manuele Bicego
Computer Science Dept.
University of Verona
Pattern Recognition
Summary

- Introduction to Pattern Recognition
- The pipeline of Pattern Recognition
  - Representation
  - Modelling
  - Evaluation
- A summarizing example: analysis of expression data
Pattern Recognition

• A possible definition:
  [R. Duda et al : Pattern Classification, 2001]

“the act of taking in raw data and making an action based on the category of the pattern”
Pattern Recognition

- A more explicit definition
  [S. Theodoridis et al. : Pattern Recognition, 2003]

“Pattern recognition is the scientific discipline whose goal is the classification of objects into a number of categories or classes.

Depending on the application, these objects can be images or signal waveforms or any type of measurements that need to be classified. We will refer to these objects using the generic term patterns.”
Pattern Recognition

Example: handwritten characters recognition

Optical Character Recognition is designed to convert your handwriting into text.
Pattern Recognition

- Easy for humans
  - It represents a fundamental process in everyday life of human beings

- Automatic Pattern Recognition: perform Pattern Recognition with machines
  - The goal is replicate this skill of human beings in an automatic way
Pattern Recognition

- Typically solved using the “learning from examples” paradigm
  - Distill the knowledge (the model) from a set of instances sampled from the problem (the training set)
  - The goal is not “to memorize” but “to learn”
    - Successful learning: capability of explaining also objects not present in the training set
    - Overtraining: the model “memorizes” the training set

- For machines Pattern Recognition is very difficult, due to different factors
Challenges in *automatic* pattern recognition

High variability among objects inside the same class

Face recognition: expression variations
Challenges in *automatic* pattern recognition

Similarity between objects of different classes

Face recognition
Challenges in *automatic* pattern recognition

Often humans have better “sensors”
Challenges in *automatic* pattern recognition

Humans are able to capture the “whole picture”

What a human sees:

What a pc sees:
Challenges in *automatic* pattern recognition

Humans are better in dealing with changes in external conditions.
Pattern Recognition in bioinformatics

- PR tools and techniques are of paramount importance for analysing and understanding biological data
- Why: many problems in bioinformatics deal with the concept of groups/classes

“To which family does this protein belong to?”
“Which are the co-regulated genes?”
“Does this piece of DNA identify a gene?”
Pattern Recognition in bioinformatics

- Another reason: the possibility of exploiting the “learning paradigm” to create models for biological data or to learn a time-consuming operation

“Create a computational model able to capture the time evolution of the expression of a gene”

“Create an automatic system able to identify a brain anomaly in a MRI brain scan”
PR in bioinformatics: challenges

- Huge complexity of biological systems and problems

  “recognize an apple from an image” vs “identify a morphological change in a brain scan”
PR in bioinformatics: challenges

- Need for integration of different sources of knowledge
- A given problem is typically analysed from many different view points (e.g. schizophrenia is analysed via personal interviews or via MRI brain scans)
PR in bioinformatics: challenges

• The problem of data dimension:

  • For some problems we have **a too large amount** of data
    • e.g. Finding relevant information within genomes may be problematic

  • For some problems we have **a too little amount** of data
    • e.g. It's difficult to characterize a class of rare tumors with only few examples
PR in bioinformatics: challenges

- Often data coming from bio-medical experiments are of poor quality
  - Low resolution (e.g. MRI scans)
  - Corrupted with noise (e.g. sequencing data)
  - Prone to significant changes depending from experimental conditions (e.g. NMR)

- It is fundamental to interact with medical/biological partners, who have different background – expectations – knowledge – language!
The Pattern Recognition Pipeline

Data acquisition: How to collect measurements from the entities of the problem

Representation: How to represent the objects to build the PR system

Modeling: How to create a model from the training set

Evaluation: How to extract information
The Pattern Recognition Pipeline

Data acquisition
- How to collect measurements from the entities of the problem

Representation
- How to represent the objects to build the PR system

Modeling
- How to create a model from the training set

Evaluation
- How to extract information
Data acquisition

- The problem of sampling: how to collect data?

Which sensor?

Issues: Resolution, cost, time, reliability, robustness to changes
Data acquisition

• The problem of sampling: how to collect data?

Which conditions?

Controlled (easier) or into the wild (more difficult but more realistic)?
The Pattern Recognition Pipeline

Data acquisition

Representation

Modeling

Evaluation

How to collect measurements from the entities of the problem

How to represent the objects to build the PR system

How to create a model from the training set

How to extract information
Data representation

- Data representation: the problem of finding a proper digital representation for the collected objects
- This represents the most crucial problem in the design of Pattern Recognition systems

Crucial concepts: **feature** and **pattern**
Feature

- Feature: characteristic of the object of the problem under consideration
  - It can be directly a measure of the sensor (e.g. graylevel of a pixel in an image)
  - It can also be an “elaboration” of (raw) measurements

Example.
From the image of a fish (matrix containing the raw measurements) we can extract a feature measuring its length
Pattern

- Pattern: `collection` of features (a way of putting together different features)

Example

We can concatenate the two features to form a vector

Pattern: [ length, height ]
Feature / Pattern

- Features can be of different type:
  - Numeric, categorical, binary (symmetric or asymmetric), continuous, discrete ...
- Features can also derive from complex operations
  - Example 1: Mathematical transforms (e.g. Fourier, wavelets)
  - Example 2: Image processing (e.g. contour of an object, segmentation)

Crucial choice!
Feature / Pattern

- The pattern is a collection of features
- Many different possibilities:
  - Vectors, sequences, strings, sets, graphs, transactions (the concept will be clearer later during the course)

- Common choice: vectors!
  - Many approaches in Pattern Recognition deal with vectorial representations
The issue of representation

- Choosing the proper representation is fundamental, and represents the core of any Pattern Recognition system

  "Proper representation" = representation which permits the generalization

Remind: The generalization represents the capability of a given PR system to explain (and classify) also objects not present in the training set
The issue of representation

Good representation

• CLASS SPECIFIC: Different classes should be represented in different positions in the representation space.

• COMPACT: Every class should be represented in a small set of finite domains.
The issue of representation

The **Compactness Hypothesis**

*Representations of real world similar objects are close. There is no ground for any generalization (induction) on representations that do not obey this demand.*

The issue of representation

**Important Note.** The compactness hypothesis requires that:

*Similar objects → close representations*

The hypothesis is not sufficient for perfect classification as dissimilar objects may be close.
The issue of representation

Direct consequence: classes can overlap, we need classifiers able to deal with this!
The issue of representation

TRUE REPRESENTATIONS:
Similar objects are close and Dissimilar objects are distant.

Linear classifiers (or domain-based classifiers) are enough!
The curse of dimensionality

- **Curse of dimensionality (features):** “problems which may occur when **the number of features is too high** with respect to the number of objects which are present in the training set”

NOTE: This holds for vectorial spaces; for other representations the dual version holds

- **Curse of dimensionality (model):** “problems which may occur when **the model is too complex** with respect to the number of objects which are present in the training set”
The curse of dimensionality

Peaking Phenomenon, Overtraining, Curse of Dimensionality, Rao's Paradox
The curse of dimensionality

Curse of dimensionality: growing the complexity of the model
The curse of dimensionality

![Graph showing classification error vs degree of polynomial classifier complexity for training and test sets.](image-url)
The curse of dimensionality

- Problems which may occur:
  - With many features the space is almost empty, it is difficult to perform an accurate estimation of models
    - Overtraining may occur (the model is too specialized on the training set, and does not generalize to other objects)
  - Computational problems (large representations)
Reduction of dimensionality

- One possible solution to prevent the curse of dimensionality is to reduce the number of features

First approach: **Feature transformation**.

- Goal: extract a more compact representation by **transforming** the original set of features
- Examples: Principal Component Analysis, Fisher Discriminant Analysis, Independent Component Analysis ...
Reduction of dimensionality

Second approach: **Feature selection**

- Goal: extract a more compact representation by **selecting** some features (and discarding the others)
- Features are selected on the basis of different criteria, like redundancy, variability, discriminative power, entropy, ...
- To obtain the set of relevant features different algorithms can be exploited
  - Sequential Forward FS, lasso, SVMRelief ...
The Pattern Recognition Pipeline

- Data acquisition: How to collect measurements from the entities of the problem
- Representation: How to represent the objects to build the PR system
- Modeling: How to create a model from the training set
- Evaluation: How to extract information
Modeling

- Given the representation, the goal is to derive a model for the problem under examination.

- The model is built starting from the training set and some a priori knowledge.

- The goal is to derive a model able to generalize also to objects not present in the training set (learn, not memorize!)
The training set

- The training set contains **examples** sampled from the problem at hand

- We should have a “good” training set:
  - **Large** (many patterns)
  - **Complete** (all categories should be represented)
  - **Variable** (it should contains the principal variations of the patterns)

- The training set is crucial, since it represents the basis on which the model is built
The a priori information

• The a priori information depends on the task we are aiming at solving
• We can reverse the perspective: depending on the available a priori information we can realize different Pattern Recognition systems
  • Different learning problems

Examples: Classification, clustering, detection
Classification

- **Classifier**: a model able to assign an unknown pattern to one among a set of pre-specified categories

**Question**: to which category does this object belong to?

**Note**: The possible categories are known in advance: “apples”, “pears”
Classification

The training of the model is performed via **Supervised learning**: for each object of the training set we know the exact label.

Training set: some apples and some pears

Each object in the training set has its own label.
Classification

Feature Space

Test object classified as 'A'

Classifier
Detection

- **Detector**: a model able to decide if an unknown pattern belongs or not to the category of interest

**Question**: does this object belong to the “apple” category?

**Note**: It can be seen as a binary classification problem, where the possible categories are: “apples”, “not apples”
Detection

Also in this case the training of the model is performed via **Supervised learning**: the training set contains only objects from the category of interest.

Training set: only apples

A priori information: we know that there are only apples.
Detection

Detector
Clustering

- **Clustering system**: a model able to partition a set of patterns into groups of “similar” objects

Question: which are the groups in the set of patterns?
Clustering

- In this case the training of the model is performed via **Unsupervised learning**: the training set contains only objects, we don't have any a priori information

- We can have the information about the number of groups (clusters) which are present in the data

- In this case, at the end of the training we already have the result (i.e. the clusters)
Clustering system
Other learning paradigms

- There are other learning paradigms, depending on the available a priori knowledge
  - Active learning, multiple instance learning, transfer learning, semi-supervised learning (more clear in the following)

- **Example.** Semi-supervised learning:
  - in a classification scenario some points of the training set are labelled, some others not
  - Possible reason: labelling is very expensive, getting data is not
Other learning paradigms

- How to deal with this situation
  - A trivial solution is to learn a model by using only labelled points (we want to derive a classifier, so let's only use labelled objects)

- Semi-supervised learning solution:
  
  *Can we exploit also unlabelled points to define a better classifier?*
Other learning paradigms
Approaches for classification

Classification: huge amount of classification approaches have appeared in the past

- Based on different theories, assumptions, algorithms, representations
  (Not reviewed here, see “further readings”)

- Main paradigm: Bayes decision theory
Bayes decision Theory

- Probabilistic perspective: all relevant entities are modelled via probability distributions
- Main ingredients:
  - $x$, object of the problem
  - $\omega_1, \omega_2, \ldots \omega_C$ are the classes of the problem
  - $p(\omega_i) \rightarrow$ Prior probability of the class $\omega_i$
  - $p(x|\omega_i) \rightarrow$ Conditional probability
  - $p(\omega_i|x) \rightarrow$ Posterior probability
Bayes decision Theory

- The three probabilities are linked together via the Bayes rule

\[ P(\omega_j | x) = \frac{p(x | \omega_j) P(\omega_j)}{p(x)} \]

where \( p(x) \) is called “evidence”
Bayes decision Theory

Decision rule.

*Given an object $x$ to classify, assign $x$ to the class which posterior is maximum*

$$\text{class}(x) = \arg \max_j P(\omega_j | x)$$

Or, in the same way (the evidence is a constant)

$$\text{class}(x) = \arg \max_j P(x | \omega_j) P(\omega_j)$$
Bayes decision Theory

- The Bayes rule guarantees to achieve the minimum probability of error

but...

- Probabilities are not known, and should be estimated from the training set

- Two general ways of implementing this rule: Generative approaches vs Discriminative approaches
Bayes decision theory

- Generative approaches: estimation of the posterior probabilities of each class via the estimation of the \textit{conditional} probabilities and the \textit{prior} probabilities
  - More explicitly: estimation of a “\textit{model}” for every class

- Discriminative approaches: direct estimation of the posterior probabilities
  - More explicitly: direct estimation of the \textit{boundary} of the classifier
Generative approaches

- Generative approaches: the estimation of conditional probabilities can be performed via parametric or not parametric approaches
  - **Parametric**: the shape of the pdf is known, we should estimate the parameters (e.g. Maximum Likelihood estimation)
    - Normal linear classifiers, normal quadratic classifier
  - **Non Parametric**: no assumptions on the shape of the pdf, which is estimated using only the data
    - Parzen windows classifier, K-nearest neighbors
Discriminative approaches

- Direct estimation of the posterior probabilities via the direct estimation of the boundary

**Boundary**: Points $x$ for which $p(\omega_1|x) = p(\omega_2|x)$

The boundary can also derive not from the posteriors but from geometric intuitions (e.g. Support Vector Machines)
Approaches for clustering

Clustering: also in this case many approaches have been designed.

- Please note that clustering is more challenging than classification: no a priori information (unsupervised), no ground truth (how can the effectiveness of a clustering system be assessed?)

- Main subdivision: partitional vs hierarchical approaches
  - **Partitional**: the goal is to find a partition of the data
  - **Hierarchical**: the goal is to find a dendrogram, i.e. a set of nested partitions
Original Points

A Partitional Clustering

Hierarchical Clustering

Dendrogram
The Pattern Recognition Pipeline

- **Data acquisition**: How to collect measurements from the entities of the problem
- **Representation**: How to represent the objects to build the PR system
- **Modeling**: How to create a model from the training set
- **Evaluation**: How to extract information
Evaluation

- This represents the step when the Pattern Recognition system designed in the previous phases is used
  - A classifier is fed with a set of unknown objects
  - A clustering result is analysed

- Typically, more than extracting information, here we are interested in “judging” the pattern recognition system
- To analyse “how well the system behaves”
Evaluation of classifiers

- Evaluation: see if a classifier correctly predicts the labels of the objects fed to it

  ![Diagram]

  - Classifier → "apple"  CORRECT!
  - Classifier → "pear"   ERROR!

Problem: how to get objects for testing?
Best solution: sample novel objects!
Evaluation of classifiers

- Re-sampling is not always possible
- *(very bad)* Idea: we can use again the training set to test the classifier
- but... the main goal of a classifier is the capability to classify also objects not present in the training set (generalization capability)

We need a separate testing set!
Evaluation of classifiers

- Cross validation solution: to divide the available objects into training and testing sets

  - Same set for training and testing may give a good classifier, but will definitely yield an optimistically biased error estimate.
  - A large, independent test set yields an unbiased and reliable (small variance) error estimate for a badly trained classifier.
  - A small, independent test set yields an unbiased, but unreliable (large variance) error estimate for a well trained classifier.
  - 50-50 is an often used standard solution for the sizes for training and testing. Is it really good? There are better alternatives.

*Holdout, Averaged Holdout, Leave One Out, Leave K out*
Evaluation of classifiers

Goodness of the approximation depends on the size of the training set

- $\varepsilon_{n-1}$: Leave-one-out (n-1) error estimate
- $\varepsilon_{\pi}$: Crossvalidation error estimate (rotation method)
- True error $\varepsilon$
- Apparent error $\varepsilon_A$ of training set

Classification error vs. Size training set
Evaluation of clustering

- Evaluation of clustering is more difficult: the ground truth is not available
- There are some criteria (internal or external)
  - **External**: Use a labelled dataset, remove labels, perform clustering and check (e.g. purity, NMI)
  - **Internal**: check separation and compactness of obtained clusters (Davies Bouldin)

*In general this remains an unsolved issue: the best option is to analyse the results from the point of view of the application context, to see if confirmations can be found*
A summarizing example: analysis of expression data
Expression data

- Expression data: measure of the “differential gene expression” mechanism

(from Gilbert, “Developmental Biology”)

The three postulates of **differential gene expression** are as follows:

1. *Every cell nucleus contains the complete genome established in the fertilized egg. In molecular terms, the DNAs of all differentiated cells are identical.*

2. *The unused genes in differentiated cells are not destroyed or mutated, and they retain the potential for being expressed.*

3. *Only a small percentage of the genome is expressed in each cell, and a portion of the RNA synthesized in the cell is specific for that cell type.*
Expression data

- There are different technologies usable to measure the differential gene expression:
  - Microarray, RNA/DNA seq, etc
- In general, all these technologies are able to provide, for a given sample, the **level of expression** of all its genes
  - Main idea: measuring the **quantity of RNA** (the expressed DNA)
  - A crucial issue is represented by the post-processing needed to obtain “numbers”
Expression data

• Typically we investigate the expression of different samples

• The goal is to analyse differences in expression among different experimental conditions
  • Diseased / healthy people
  • Different tissues
  • Different time steps
  • ...

Expression data

Result: the *gene expression matrix*
Pattern Recognition analyses

Pattern Recognition systems in the analysis of this matrix can be very useful!

**Example 1.** The clustering of genes / samples

**Example 2.** The classification of samples

**Example 3.** The curse of dimensionality and the feature (gene) selection process

**Example 4.** The evaluation: gene-ontology
The clustering of genes / samples

Hierarchical clustering of samples

Hierarchical clustering of genes
The clustering of genes / samples

- **Clustering of samples**: to detect groups of samples showing similar activation levels among all genes

- Sample labels are typically given, but it can be useful to discover sub types of diseases
- Typically less investigated
The clustering of genes / samples

- **Clustering of genes**: to detect groups of genes showing similar activation levels among all experiments

- Similar activation patterns probably correspond to similar functions (co-regulated genes)
The clustering of genes / samples

- Another option: **simultaneous** clustering of genes and samples → BICLUSTERING
  - A bicluster represents a **group of genes** showing similar activation levels among a **group of samples**
The classification of samples

- Goal: to build a Pattern Recognition system able to **discriminate** between different categories of samples by analysing the expression levels among all genes

- Example: discriminate between healthy and diseased people
  - Final Goal: given the expression level of all genes of an unknown individual, predict the presence (of the absence) of a given disease
The classification of samples

- The classification problem is very challenging!
- Motivations:
  - Noisy data (deriving from cutting-edge technologies)
  - Complex data
  - Biological variability between samples
  - Many genes with respect to the number of samples → Curse of dimensionality!
The curse of dimensionality and the feature (gene) selection process

- Experiment with real data: expression matrix of 12533 genes relative to 174 individuals, divided in 5 classes (4 different kinds of tumors plus 1 class of healthy)
- The dataset is divided in training set and testing set
- The classifier is learnt on the training set and tested on the testing set
- Curve of errors by varying the number of genes employed
Feature (gene) selection process

- It is essential to select a small set of “relevant” features (feature selection – here called gene selection)

Example: filtering out genes with low variance (genes which do not vary among the different experiments)
The evaluation: gene-ontology

- In order to evaluate clustering of genes (or even biclustering) we can exploit external information
- Gene Ontology can provide such information
  * Each gene is annotated using **GO terms** which indicate functions/concepts/biological processes
- Examples
  * GO:0009063: cellular amino acid catabolic process
  * GO:0008202: steroid metabolic process
The evaluation: gene-ontology

- A cluster of genes is “reasonable” if genes have many GO-terms in common
- This can be evaluated via an enrichment analysis

From http://geneontology.org/page/go-enrichment-analysis

One of the main uses of the GO is to perform enrichment analysis on gene sets. For example, given a set of genes that are up-regulated under certain conditions, an enrichment analysis will find which GO terms are over-represented (or under-represented) using annotations for that gene set.

It also returns a p-value
Further Readings


Introduction to structured data
Structured data

- Structured data: data which can be **naturally** described with a **complex representation** (strings, sequences, graphs, 3D data, images...)
Structured data

A possible (Pattern Recognition) definition for structured data:

A set of features (measures) + A relation between them

This definition holds also for the simplest example: the vectorial representation
The simplest example: vectors

- Vectorial representation: the measures (features) are arranged in a vector

(Pattern)

(Set of features)

Length, height

(Structured data)
The simplest example: vectors

• **Relation**: the first feature (length) is always in the first position of the vector, the second feature (height) is always in the second position of the vector

• To get this representation we have to:
  • Measure length and height
  • Place them in the correct position inside the vector
The simplest example: vectors

Objects are represented as points in a vectorial space (called **feature space**).
Important observations

Observation 1. The same object can be represented in different ways (changing the measures)

Observation 2. The same set of measures can be represented in different ways (changing the relation)
Beyond vectors...

- Vectors have been largely studied in Pattern Recognition
  - Vectors represent the simplest representation
  - Many approaches are based on vectorial representation (not seen here)
  - But: sometimes this representation is too simplistic to capture the real nature of the object (especially in bioinformatics)

*Focus of this course: complex representations* (Sequences, strings, images, 3D data, sets, graphs, profiles, networks...)

*Let's see them...*
Sequences

- The measures are arranged following a natural “sequential ordering”
- The relation encodes the fact that the feature in position \(i\) follows the feature in position \((i-1)\)

- The ordering is fundamental: if we change the order we would destroy the information
  - Or we would drastically change it

```
AAT
Asparagine
Moving 'T' to the beginning
```
```
TAA
stop codon
```
Sequences

Example: the EEG signal.

- Each signal encodes the electrical activity recorded by an electrode placed on the head

Relation: temporal
Strings

• Same as sequences but the features/measures are **categorical** (i.e. from a discrete alphabet)

Example: biological sequences

```plaintext
atgcgatcgatcgatcgatcgatcaggcgctacgagcgccggcaggacct
catcatcgatcag
```

_Nucleotides: alphabet is made by 4 symbols_

```plaintext
MRPQAPGSLVDNEDELRMAPWYWGRISREEAKSILHGTKPDGSFLVRDALSMKGEYTLTLMKDGCEKL
LIKICHMDRKYGFIETDLFSVEMINYKENSLSMYNKTLDITLSNPPIVRAREDEESQPHGDLCLLSNEFIRTCQLLOQLQIQNLQLENKRNSFNLAIREELQEKKLHQSVFGNTEKIFRNQIKLNESFMKAPADA...
```

_Aminoacids: alphabet is made by 20 symbols_

```plaintext
```
Sequences/strings: PRO and CONS

- PRO: this representation is able to capture sequential evolution

- CON: the PR model should describe this ordering

- CON: sequences can have different length: there is not a vector space anymore
  - Even the simplest concepts (like the mean) are difficult to realize
Images

- The measures are arranged in a matrix

The measures depend on the sensor type

X-ray image: each pixel measures the level of absorption to X-ray
Image from a camera: each pixel measures the level of light intensity
Images

- The relation reflects the **spatial proximity** between measures
Images: PROS and CONS

- PRO: this representation captures spatial proximity (fundamental in many problems)
- PRO: interpretable representation (very easy to reason on images)

- CON: this spatial proximity should be modelled by PR systems
Images: PROS and CONS

- CON: an image cannot be considered as a point in a vector space (by transforming the matrix in a vector we lose spatial information)

- CON: high dimensional data (images can contain millions of measures)

- CON: it is not always easy to exclude background (the image often contains also not relevant information)
3D data

- The measures are **placed** in a three dimensional space (i.e. they are equipped with 3D coordinates)

*Proteins: aminoacids are arranged in a three-dimensional structure*

*Every aminoacid has 3-d coordinates*
Proteins represent a clear example of measures (amino acids) which can be arranged in two different ways (strings or three dimensional data)
3D data: PROS and CONS

- PRO: very expressive (real world is 3D!)
- CON: in some cases having the 3D structure can be very expensive/difficult with respect to simplest representations

*aminoacid sequence vs structure of a protein*

*visual image vs 3D data*
3D data: PROS and CONS

- CON: high dimensional data

- CON: Definitely a not vectorial representation, difficult to deal with
  
  - (please note that in vector spaces a point represents the whole set of measures, here a point in the 3D space represents a single measure)
Sets

- The measures are simply grouped together in an unordered collection

- Very flexible representation, usable when:
  - Natural (true) representations are too complex (or does not permit to derive effective PR models)
  - Features are heterogeneous
Sets

The genome is difficult to manage as a whole, we can simplify the modelling by seeing it as a **collection of genes**

The brain can be seen as a **collection of regions** (too difficult to put together so heterogeneous descriptors)
Sets: PROS and CONS

- **PRO:** very general representation which can also capture partial (local) properties of the object

- **CON:** difficult to create PR models, since this representation makes only the assumption that the measures belong to the object
  - No information about ordering or about cardinality (i.e. number of measures)
Graphs

- The measures are arranged in a graph
  - Nodes represent features
  - Edges represent relations between features

Powerful representation!

Brain

The regions of the brain are the nodes of the graph, the edges represent connectivity (related activity)
Graphs: PROS and CONS

- PRO: very interpretable representation, it immediately provides relation between components
- PRO: there exist many mathematical tools and results available
- CON: difficult to create models for these representations
  - Graphs can have different number of nodes or different number of edges
  - Cycles can make intractable even very simple computations
Networks

• Very similar to graphs (nodes and edges which represent relations between nodes)
• Main difference: in a network a node represents a single object, not a single measure of the object

In this sense a network represents the **whole problem**

*E.g. Protein-protein interaction networks*
Networks

- Powerful representation of a problem, it can be inspected to derive crucial information about relations between objects
  - Main issue: how to build the network
- It can also be used to derive **features for objects**
  - For example mathematical properties of the node (degree, centrality, etc etc) can characterize the object
- Recently also used as a representation for individual (protein interaction networks built from protein expressed in a single individual)
Conclusions

Devising Pattern Recognition techniques able to deal with these structured data can be very challenging

- Even trivial concepts (like the mean) are difficult to be instantiated within these representations

\[ \text{mean} \left( \begin{array}{c}
\text{atgcgatc} \\
\text{ct} \\
\text{agggtc}
\end{array} \right) = ? \]
Conclusions

- But, they are worth to be investigated: structured representations are more expressive than simple vectorial representations!!

- They can capture relations among measures which can not be captured with vectors (or even they can provide representations where it is impossible to derive a vectorial one)
This course

- We will face different aspects linked to the design of Pattern Recognition systems for structured data:
  - Representation of structured data
  - Models for structured data
  - *Kernels for structured data (for kernel machines such as Support Vector Machines) and alternative learning schemes*
  - Deep learning