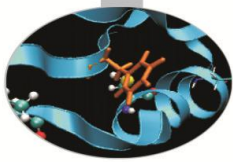
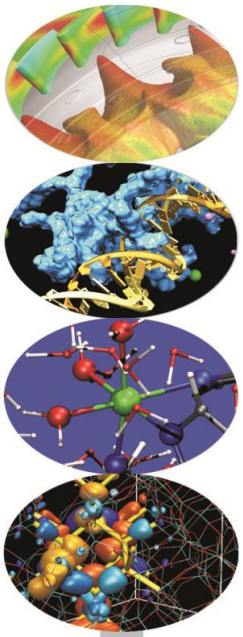
CINECA SCAI
SuperComputing Applications and Innovation



Machine Learning with Spark

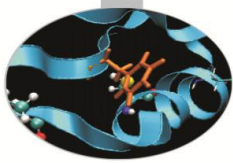


Giorgio Pedrazzi, *CINECA-SCAI*

Roma, 16/12/2015

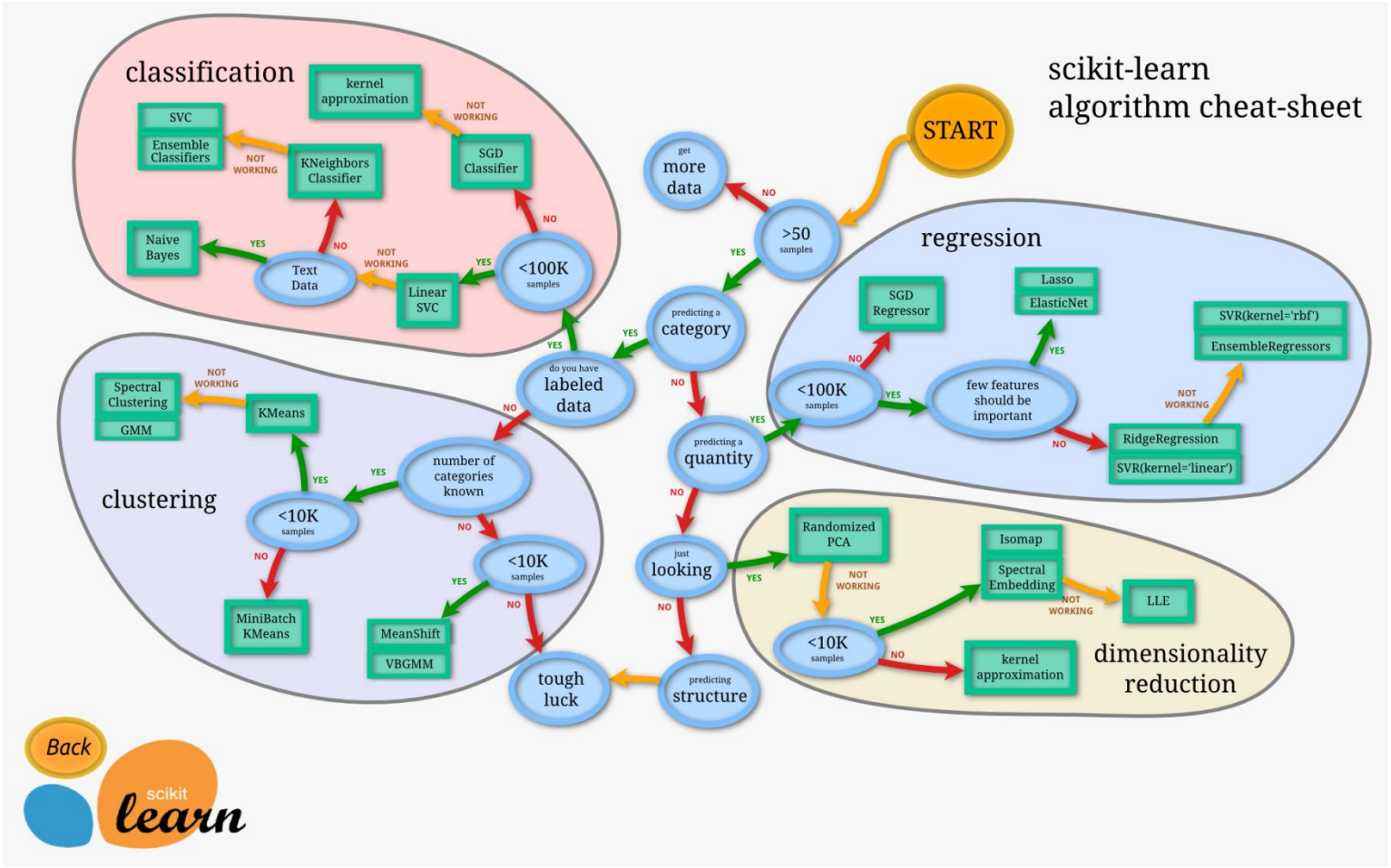
CINECA

Agenda

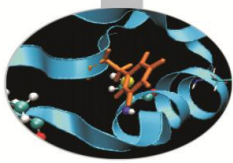


- Unsupervised learning: Clustering
  - Distance measures
  - K-means
  - Clustering validation
- Supervised learning: Classification
  - Training and test
  - Evaluation metrics
  - Decision tree
  - Naïve Bayes
- Examples with Spark MLlib in Scala and Python

Algorithm cheat-sheet



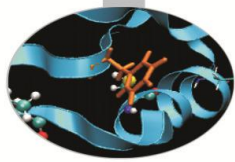
http://scikit-learn.org/stable/tutorial/machine_learning_map/



Clustering

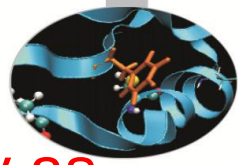
- Cluster: A collection/group of data objects/points
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis
  - find *similarities* between data according to characteristics underlying the data and grouping similar data objects into clusters
- Clustering Analysis: Unsupervised learning
  - no predefined classes for a training data set
  - Two general tasks: identify the “natural” clustering number and properly grouping objects into “sensible” clusters
- Typical applications
  - as a stand-alone tool to gain an insight into data distribution
  - as a preprocessing step of other algorithms in intelligent systems

Typical applications



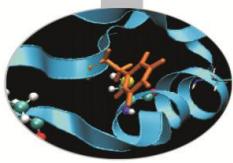
- Scientific applications
  - **Gene expression data**: Discover genes with similar functions in DNA microarray data.
  - …
- Business applications
  - **Customer segmentation:** Discover distinct groups in customer bases (insurance, bank, retailers) to develop targeted marketing programs.
  - …
- Internet applications
  - **Social network analysis:** in the study of social networks, clustering may be used to recognize communities within large groups of people.
  - **Search result grouping:** in the process of intelligent grouping of the files and websites, clustering may be used to create a more relevant set of search results.
  - …

Data Matrix



The problem must be formulated in a mathematical way as a matrix of data containing information on N objects (cases or observations ; rows of the matrix) specified by the values assigned to V variables (columns of the matrix)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

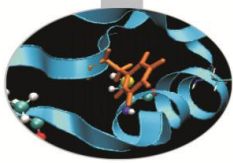


Cluster Analysis steps

- Pre processing
- Select a clustering algorithm
- Select a distance or a similarity measure (*)
- Determine the number of clusters (*)
- Validate the analysis

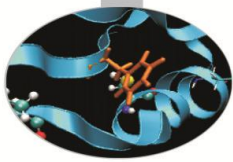
(*) if needed by the method used

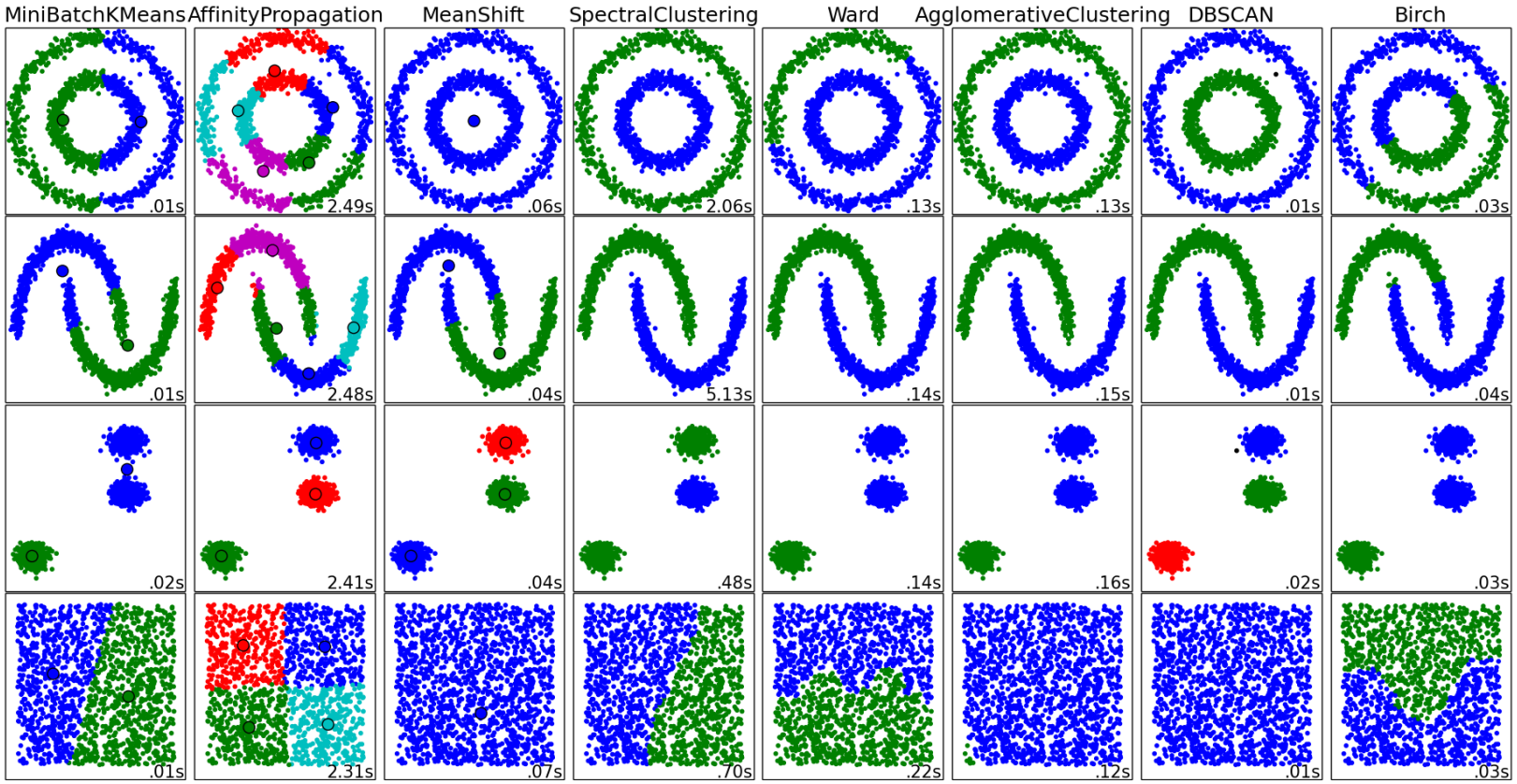
Classification of methods



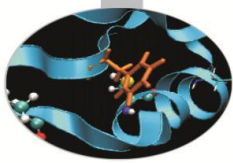
- Distance-based
  - partitioning algorithms
  - hierarchical algorithms
- Density based (DBSCAN)
- Model based
- Spectral clustering
- Combination of methods

Comparison of algorithms





Distance measure



Minkowski distance (L_p Norm)

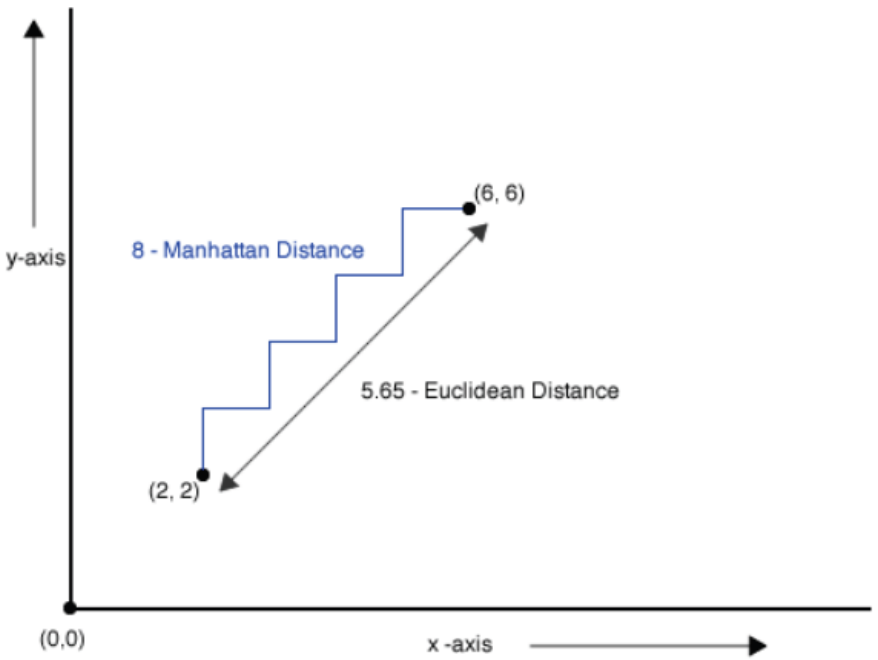
$$d(i,k) = \left[\sum_{j=1}^{d} | x_{ij} - x_{kj} |^{p} \right]^{1/p}$$

Euclidean distance (L_2 Norm)

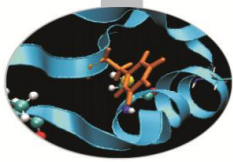
$$d(i,k) = \left[\sum_{j=1}^{d} (x_{ij} - x_{kj})^{2} \right]^{1/2}$$

Manhattan distance (city block distance)

$$d(i,k) = \sum_{j=1}^{d} | x_{ij} - x_{kj} |$$



Distance Measures



- Cosine Measure (Similarity vs. Distance)

  For $\mathbf{x} = (x_1\, x_2 \cdots x_n)$ and $\mathbf{y} = (y_1\, y_2 \cdots y_n)$

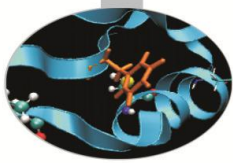
$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \bullet \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{x_1 y_1 + \cdots + x_n y_n}{\sqrt{x_1^2 + \cdots + x_n^2}\sqrt{y_1^2 + \cdots + y_n^2}}$$

$$d(\mathbf{x}, \mathbf{y}) = 1 - \cos(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \bullet \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

$$0 \le d(\mathbf{x}, \mathbf{y}) \le 2$$

  - Nonmetric vector objects: keywords in documents, gene features in micro-arrays, …
  - Applications: information retrieval, biologic taxonomy, ...

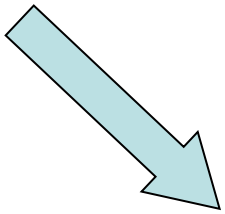
Similarity measures

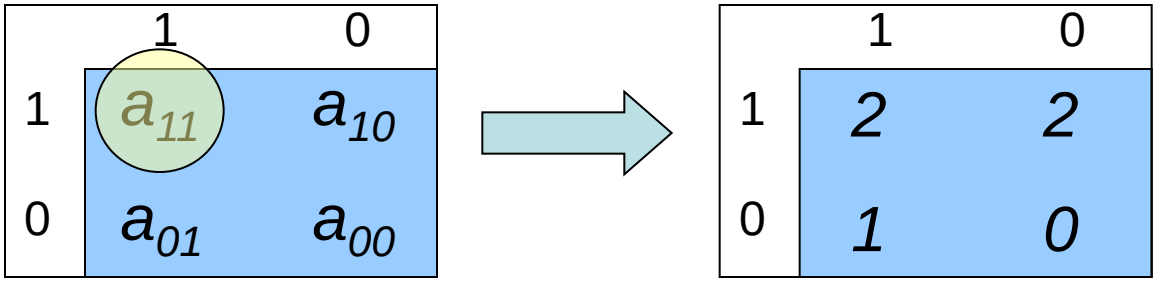


Correspondent 1's

x_k: 0 1 1 0 1

x_j: 1 1 0 1 1



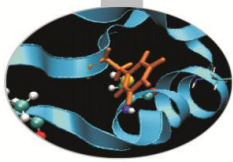


Jaccard: $d(i,k) = (a_{11}) / (a_{11} + a_{10} + a_{01})$

Condorcet: $d(i,k) = a_{11} / [a_{11} + 0.5(a_{10} + a_{01})]$

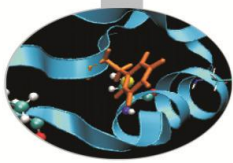
Dice bis: $d(i,k) = a_{11} / [a_{11} + 0.25(a_{10} + a_{01})]$

Partitioning Approach

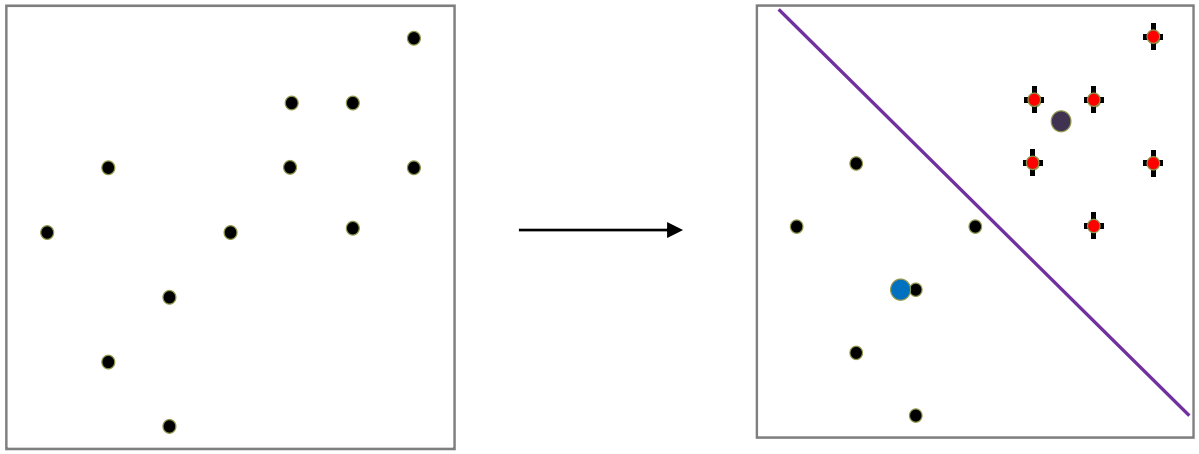


- Partitioning method: Discovering the groupings in the data by optimizing a specific objective function and iteratively improving the quality of partitions
- K-partitioning method: Partitioning a dataset D of n objects into a set of K clusters so that an objective function is optimized (e.g., the sum of squared distances is minimized, where ck is the centroid or medoid of cluster Ck)
  - A typical objective function: Sum of Squared Errors (SSE)
- Problem definition: Given K, find a partition of K clusters that optimizes the chosen partitioning criterion
  - Global optimal: Needs to exhaustively enumerate all partitions
  - Heuristic methods (i.e., greedy algorithms): K-Means, K-Medians, K-Medoids, etc.

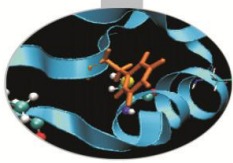
Partitioning Approach



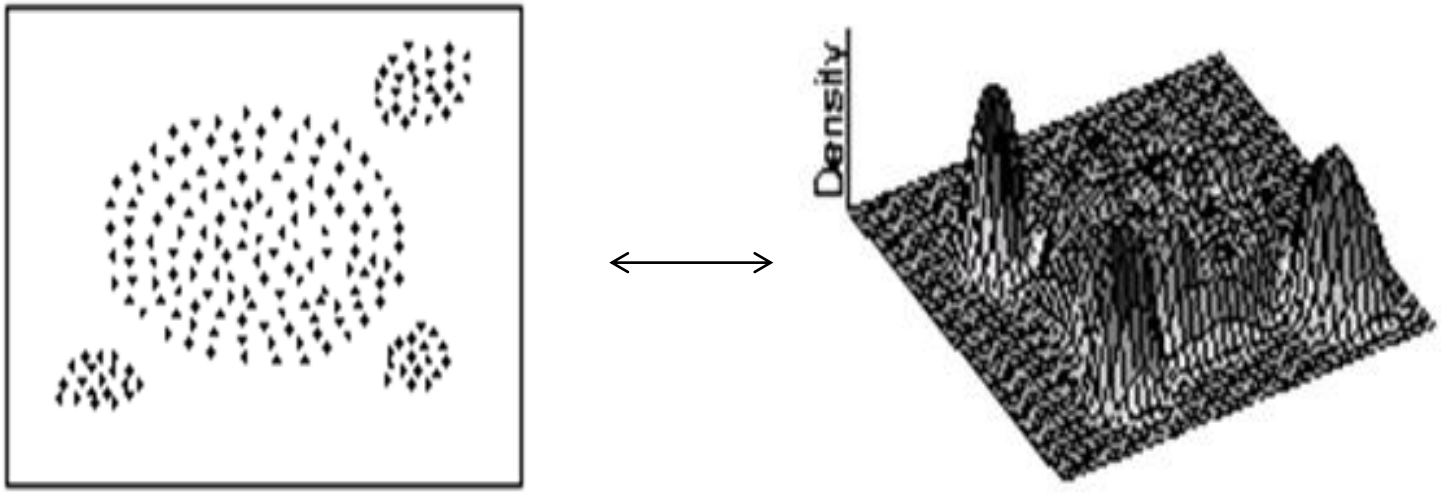
- Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square distance cost
- Typical methods: K-Means, K-Medoids, K-Medians, ……



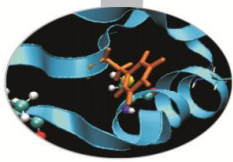
Density based Approach



- Based on connectivity and density functions
- Typical methods: DBSCAN, OPTICS, DenClue, ……

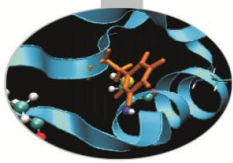


Density based approach

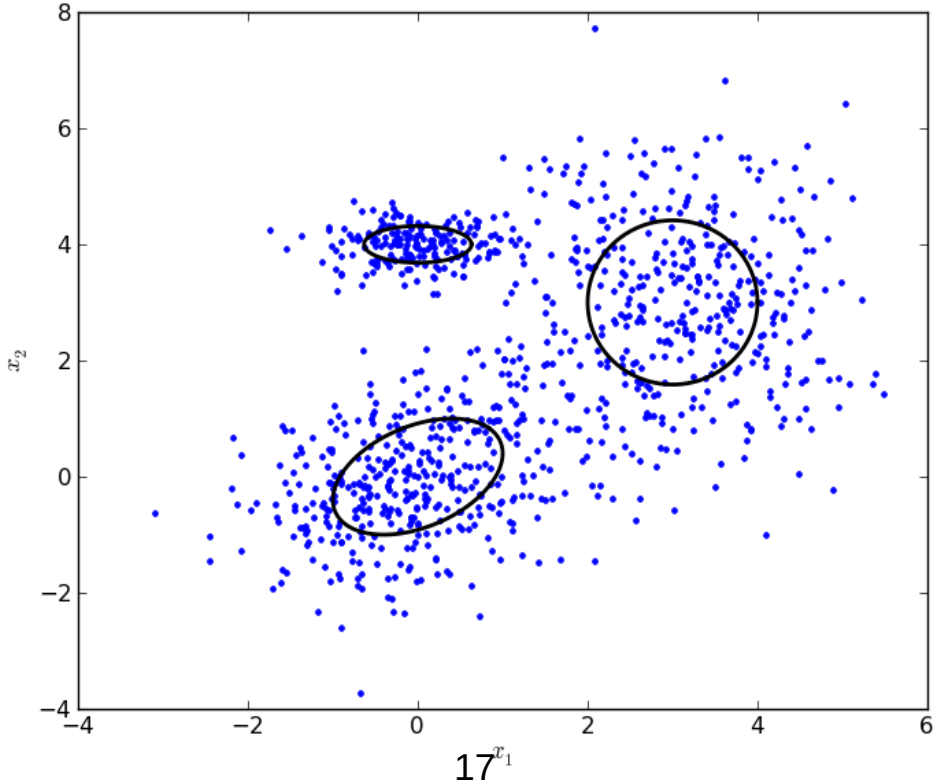


Density-based spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996. It is a density-based clustering algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature

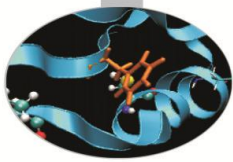
Model-based Approach



- For each cluster, a theoretical model is hypothesized in order to find the best fit.
- Typical methods: Gaussian Mixture Model (GMM), COBWEB, ……

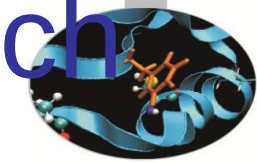


Model-based Approach

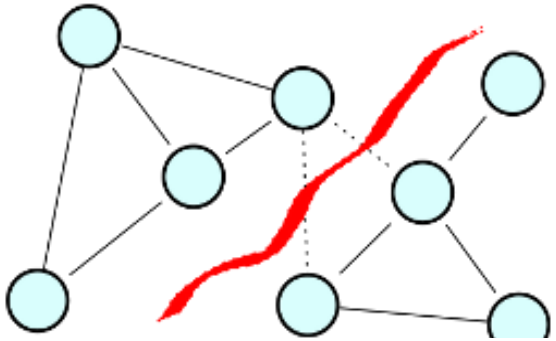


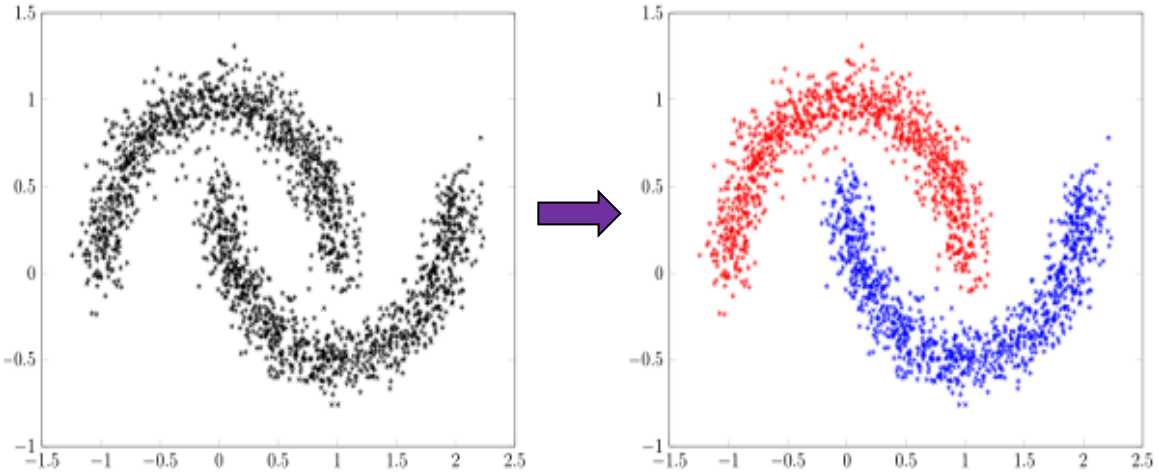
- Probabilistic model-based clustering
  - In model-based clustering, it is assumed that the data are generated by a mixture of underlying probability distributions in which each component represents a different group or cluster.
  - Cluster: Data points (or objects) that most likely belong to the same distribution
  - Clusters are created so that they will have a maximum likelihood fit to the model by a mixture of K component distributions (i.e., K clusters)

Spectral Clustering Approach

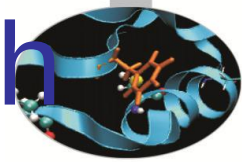


- Convert data set into weighted graph (vertex, edge), then cut the graph into sub-graphs corresponding to clusters via spectral analysis
- Typical methods: Normalised-Cuts ……





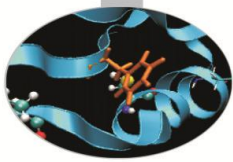
Spectral Clustering Approach



- In multivariate statistics, spectral clustering techniques make use of eigenvalue decomposition (spectrum) of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions. The similarity matrix is provided as an input and consists of a quantitative assessment of the relative similarity of each pair of points in the dataset.

- In application to image segmentation, spectral clustering is known as segmentation-based object categorization.

Combination of methods

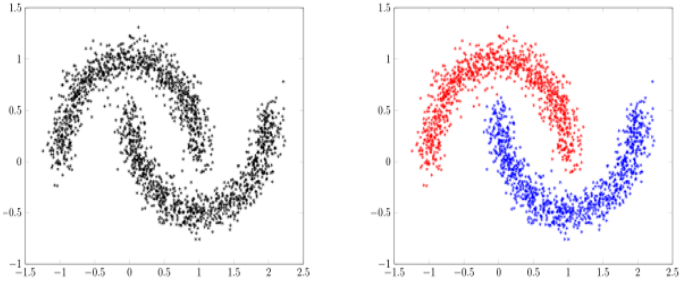


Using different methods can be useful for overcome the drawbacks of a single methods.

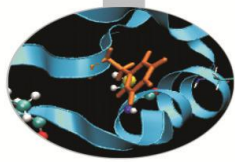
For example it is possible to generate a large number of clusers with K-means and then cluster them together using a hierarchical method.

It is important using the “single-link” method, in which the distance between two clusters is defined by the distance between the two closest data points we can find, one from each cluster.

This method has been applied to find cluster in non-convex set.



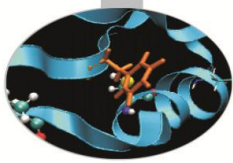
Clustering validation



Clustering validation is a technique to find a set of clusters that best fits natural partitions (number of clusters K). Generally speaking, there are two types of validation techniques, which are based on internal criteria and external criteria.

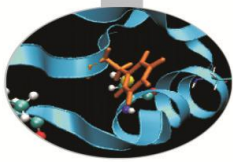
- Internal validation: based on the information intrinsic to the data alone
- External validation: based on previous knowledge about data

Supervised learning: classification



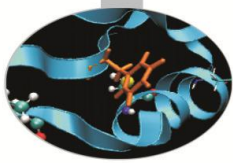
- Human learning from past experiences.
- A computer does not have “experiences”.
- A computer system learns from data, which represent some “past experiences” of an application domain.
- Learn a target function that can be used to predict the values of a discrete class attribute,
- The task is commonly called: Supervised learning, classification, or inductive learning.

Classification: Definition



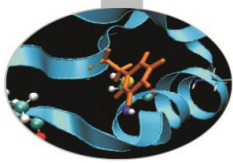
- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Supervised learning: classification



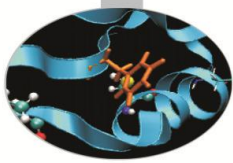
- Data: A set of data records (also called examples, instances or cases) described by
  - *k* attributes: A_1, A_2, … A_k.
  - a class: Each example is labelled with a pre-defined class.
- Goal: To learn a classification model from the data that can be used to predict the classes of new (future, or test) cases/instances.

Typical applications



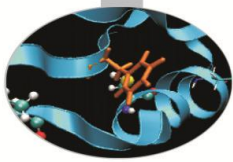
- Scientific applications
  - **Medical Diagnosis**: Given the symptoms exhibited in a patient and a database of anonymized patient records, predict whether the patient is likely to have an illness.
  - …
- Business applications
  - **Credit Card Fraud Detection**: Given credit card transactions for a customer in a month, identify those transactions that were made by the customer and those that were not.
  - **Stock Trading**: Given the current and past price movements for a stock, determine whether the stock should be bought, held or sold. A model of this decision problem could provide decision support to financial analysts.
  - …
- Internet applications
  - **Spam Detection**: Given email in an inbox, identify those email messages that are spam and those that are not.
  - …

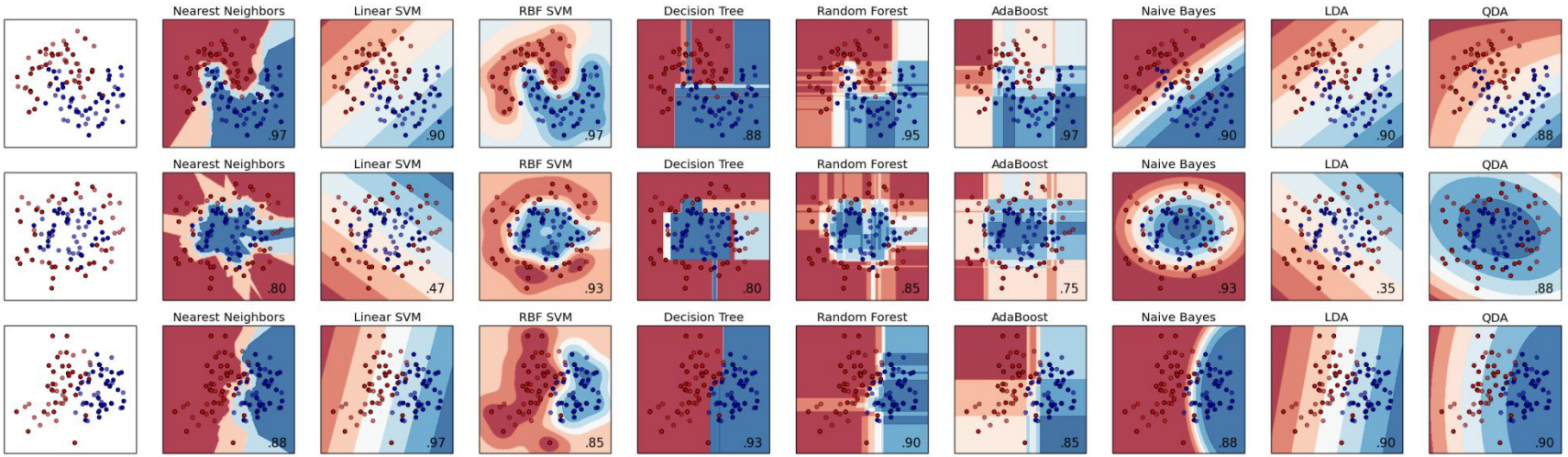
Classification Techniques



- Decision Tree based Methods
- Naïve Bayes and Bayesian Belief Networks
- Rule-based Methods
- Memory based reasoning
- Neural Networks
- Support Vector Machines

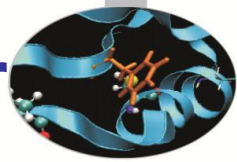
Comparison of algorithms





CINECA

Training and test a classifier

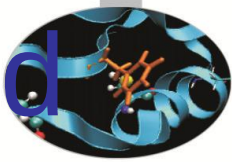


Is the model able to generalize? Can it deal with unseen data, or does it overfit the data? Test on hold-out data:

- **split** data to be modeled in training and test set
- **train** the model on training set
- evaluate the model on the training set
- **evaluate** the model on the test set
- difference between the fit on training data and test

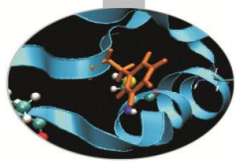
data measures the model's ability to *generalize*

Methods to create training and test data



- Fixed
  - Leave out random N% of the data
- K-fold Cross-Validation
  - Select K folds without replace
- Leave-One-Out Cross Validation
  - Special case of CV
- Bootstrap
  - Generate new training sets by sampling with replacement

Evaluation metrics



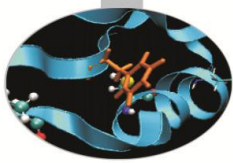
Confusion matrix

The known class of test samples is matched against the class predicted by the model		Predicted labels (model)		
		False	**True**	
True labels (target)	**False**	TN	FP	Specificity TN / (FP+TN)
	True	FN	TP	Sensitivity TP / (TP+FN)
		Negative Predictive Value TN / (TN + FN)	Positive Predictive Value TP / (TP + FP)	Accuracy (TP+TN) / (TP+FP+TN+FN)

⇨ Recall

⇩ Precision

F-score = 2*Precision*Recall / (Precision + Recall)
Error rate = 1 – Precision
FP rate = 1 – Specificity

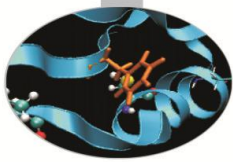


Evaluation metrics

Accuracy baselines

- Base Rate
  - Accuracy of trivially predicting the most-frequent class
- Random Rate
  - Accuracy of making a random class assignment
- Naive Rate
  - Accuracy of some simple default or pre-existing model

Building a Decision Tree

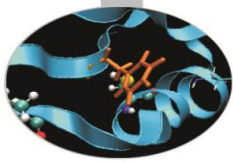


- Choose the attribute with the highest Information Gain
- Create branches for each value of attribute
- Partition examples on the basis of selected attributes
- Repeat with remaining attributes
- Stopping conditions
  - All examples assigned the same label
  - No examples left

Problems

- Expensive to train
- Prone to overfitting
  - perform well on training data, bad on test data
  - pruning can help: remove or aggregate subtrees that provide little discriminatory power

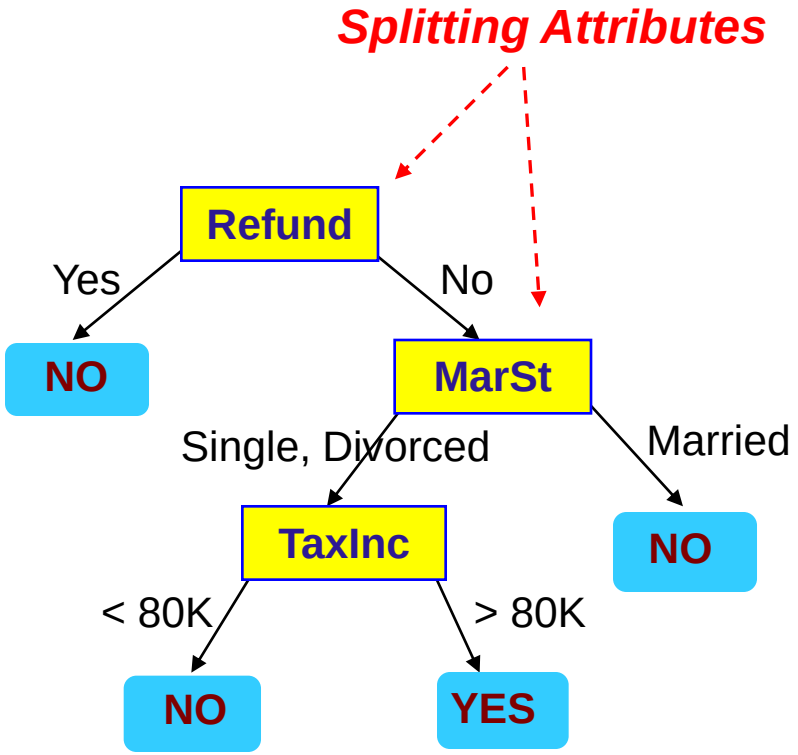
Example of a Decision Tree



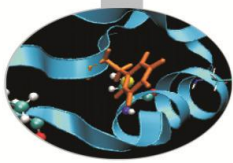
categorical | categorical | continuous | class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Model: Decision Tree



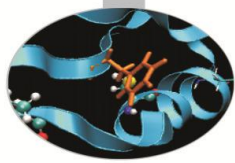
Bayesian classification

The classification problem may be formalized using **a-posteriori probabilities**:

P(C|X) = prob. that the sample tuple $X=<x_1,\ldots,x_k>$ is of class C

Idea: assign to sample X the class label C such that **P(C|X) is maximal**

Estimating a-posteriori probabilities



Bayes theorem:

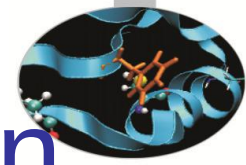
$$P(C|X) = P(X|C) \cdot P(C) / P(X)$$

P(X) is constant for all classes

P(C) = relative freq of class C samples

C such that **P(C|X)** is maximum =
C such that **P(X|C)·P(C)** is maximum

Problem: computing P(X|C) is unfeasible!



Naïve Bayesian Classification

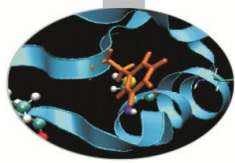
Naïve assumption: **attribute independence**

$$P(x_1,\ldots,x_k|C) = P(x_1|C)\cdot\ldots\cdot P(x_k|C)$$

- If i-th attribute is categorical:
  $P(x_i|C)$ is estimated as the relative freq of samples having value x_i as i-th attribute in class C
- If i-th attribute is continuous:
  $P(x_i|C)$ is estimated thru a Gaussian density function

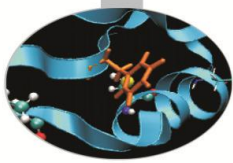
Computationally easy in both cases

Spark MLlib



- MLlib is a Spark subproject providing machine learning primitives:
- MLlib's goal is to make practical machine learning (ML) scalable and easy. Besides new algorithms and performance improvements that we have seen in each release, a great deal of time and effort has been spent on making MLlib *easy*.

Spark MLlib



- MLlib algorithms
  - classification: logistic regression, naive Bayes, decision tree, ensemble of trees (random forests)
  - regression: generalized linear regression (GLM)
  - collaborative filtering: alternating least squares (ALS)
  - clustering: k-means, gaussian mixture, power iteration clustering, latent Dirichelet allocation
  - decomposition: singular value decomposition (SVD), principal component analysis, singular value decompostion
- Spark packages availables for machine learning at http://spark-packages.org