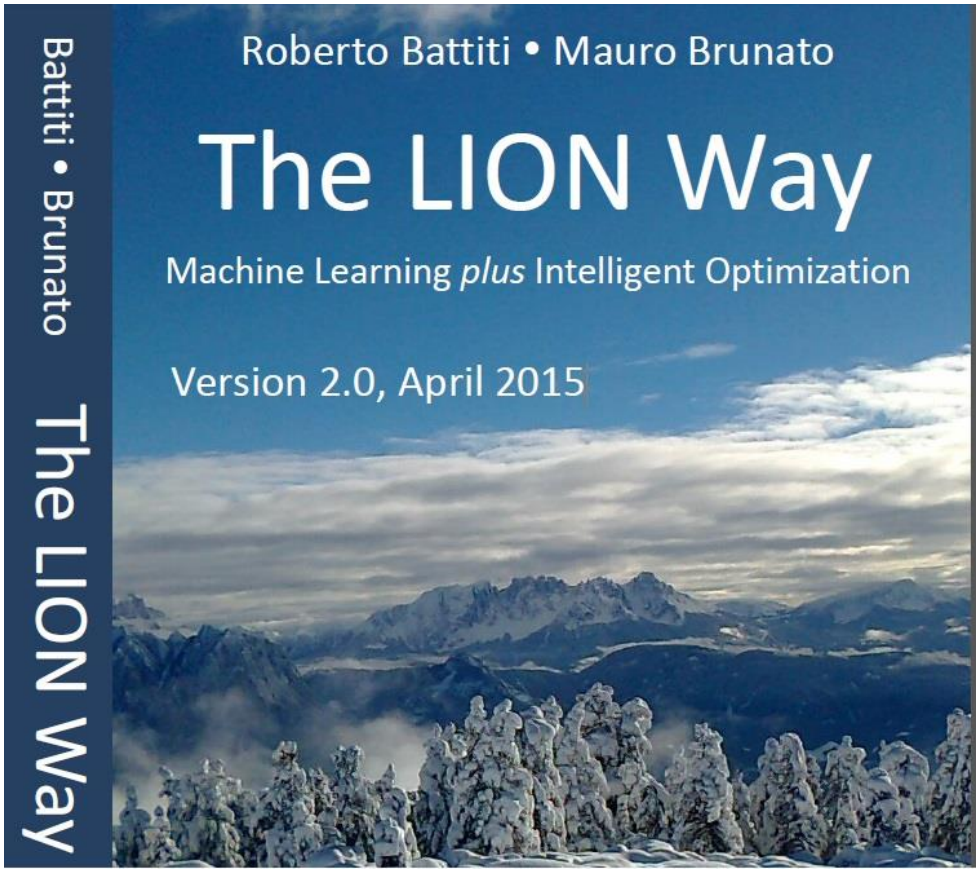


ROBERTO BATTITI, MAURO BRUNATO.
The LION Way: Machine Learning plus *Intelligent Optimization*.
LIONlab, University of Trento, Italy,
Apr 2015

http://intelligent-optimization.org/LIONbook

Top-down clustering: K-means

So out of the ground the Lord God formed every beast of the field and every bird of the air, and brought them to the man to see what he would call them; and whatever the man called every living creature, that was its name. **The man gave name**s to all cattle, and to the birds of the air, and to every beast of the field.

(Book of Genesis)

What can be learnt *without* teachers and labels?

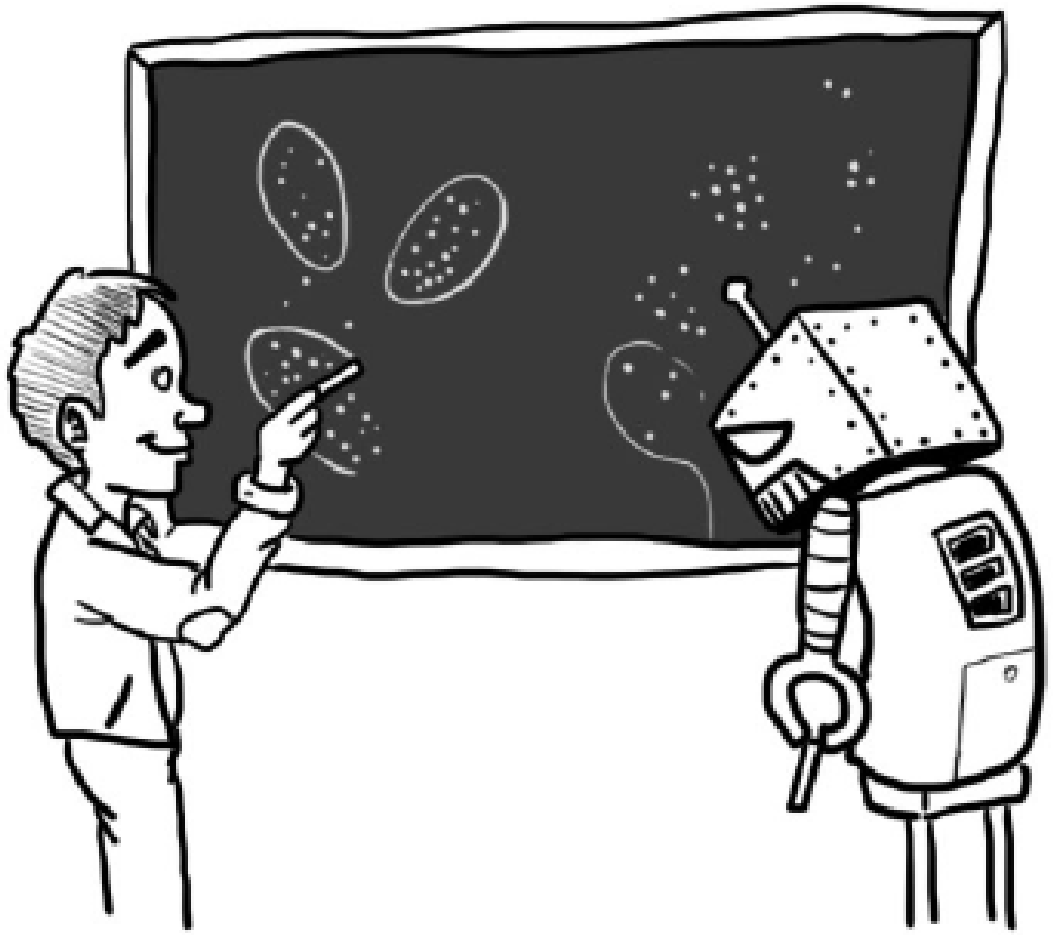
- Modeling and understanding structure is at the basis of our cognitive abilities.
- A name is a way to **group** different experiences so that we can start speaking and reasoning (think about animal species, or continent's names)

An example

- Clustering different flowers in a meadow without knowing names

Clustering

- Clustering: grouping similar things together, then one can label the groups with names.
- Compression of information (prototypes)
- The prototype summarizes the information contained in the subset of cases which it represents
- When similar cases are grouped together, one can reason about groups instead of individual entities.



Clustering is deeply rooted into the human activity of grouping and naming entities.

Practical applications

Clustering is used in a wide range of areas like:

- Marketing (Market **segmentation**)
- Finance (Risk-analysis of financial portfolios, diversification)
- Healthcare (grouping symptoms and defining illnesses, grouping genes in biological networks)
- Text mining (construction of semantic networks)

Approaches for unsupervised learning

- **Top-down**:

First the number of classes is decided, then the division is performed.

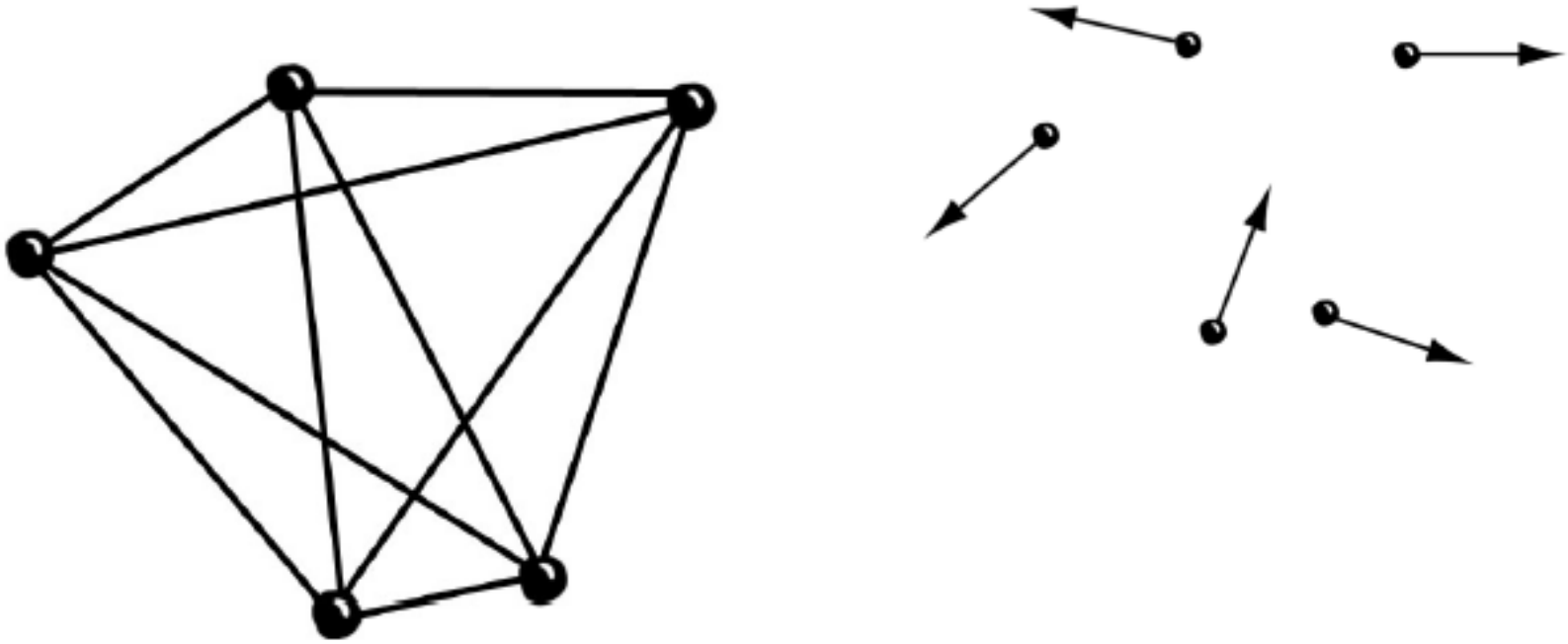
- **Bottom-up**:

One starts by merging the most similar items and stops when the achieved grouping “makes sense” for the application.

Approaches for unsupervised learning

- Dimensionality reduction (identify the "directions of variation" in the data)
- Generative models (cluster → a model for a probability distribution of the process producing the observed examples in the cluster)
- Semi-supervised learning (use *both* labeled *and* unlabeled examples in the learning process)

Clustering: Representation and metric



External representation by relationships (left) and **internal representation** with coordinates (right). In the first case mutual similarities between pairs are given, in the second case individual vectors.

Clustering: Representation and metric (2)

Two different contexts

1. An **internal representation** is available for each entity, and mutual similarities are derived from it

$$\delta_E(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \mathbf{x}\| = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2}.$$

2. Only **external representation of dissimilarities** is available.

Clustering: Representation and metric (3)

- The effectiveness of a clustering method *depends* on the similarity metric, which is strongly problem-dependent.

- Let's denote the dissimilarity between entities x and y as $\delta(x, y)$.

Dissimilarity metrics

- If an **internal representation** is present, a metric can be derived by the usual Euclidean distance:

$$\delta_E(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \mathbf{x}\| = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2}.$$

- Or the Manhattan norm:

$$d_{ij}^{\text{Manhattan}} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_1 = \sum_{k=1}^{n} |\boldsymbol{x}_{ik} - \boldsymbol{x}_{j_k}|.$$

- Or the cosine similarity

$$\cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = \frac{\sum_{i=1}^{M} x_i \times y_i}{\sqrt{\sum_{i=1}^{M} (x_i)^2} \times \sqrt{\sum_{i=1}^{M} (y_i)^2}},$$

Dissimilarity metrics (2)

- If different coordinates have **very different ranges**, the Euclidean distance may be dominated by a subset of coordinates
- This is the case if **units of measure** are picked in different ways
- It cab be useful to **normalize values** and make them dimensionless, as follows:

$$\delta_{norm}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{M} \left(\frac{x_i - y_i}{\text{maxval}_i - \text{minval}_i} \right)^2},$$

K-means for hard and soft clustering

- **Hard clustering** problem: partition the entities D into k disjoint subsets $C = (C_1, \ldots, C_k)$ to reach the following **two objectives**:

1. Minimization of the average **intra-cluster dissimilarities**

$$\min \sum_{d_1, d_2 \in C_i} \delta(\mathbf{x}_{d_1}, \mathbf{x}_{d_2}). \quad \text{or} \quad \min \sum_{d \in C_i} \delta(\mathbf{x}_d, \mathbf{p}_i).$$

2. Maximization of **inter-cluster distance**

Clustering is a multi-objective optimization task

K-means for hard and soft clustering(2)

- **Divisive algorithms** are very simple clustering algorithms: begin with the whole set and divide it into successively smaller clusters
- For each cluster, its **prototype** is calculated by minimizing the its quantization error:

$$\text{Quantization Error} = \sum_{d} \|\mathbf{x}_d - \mathbf{p}_{c(d)}\|^2,$$

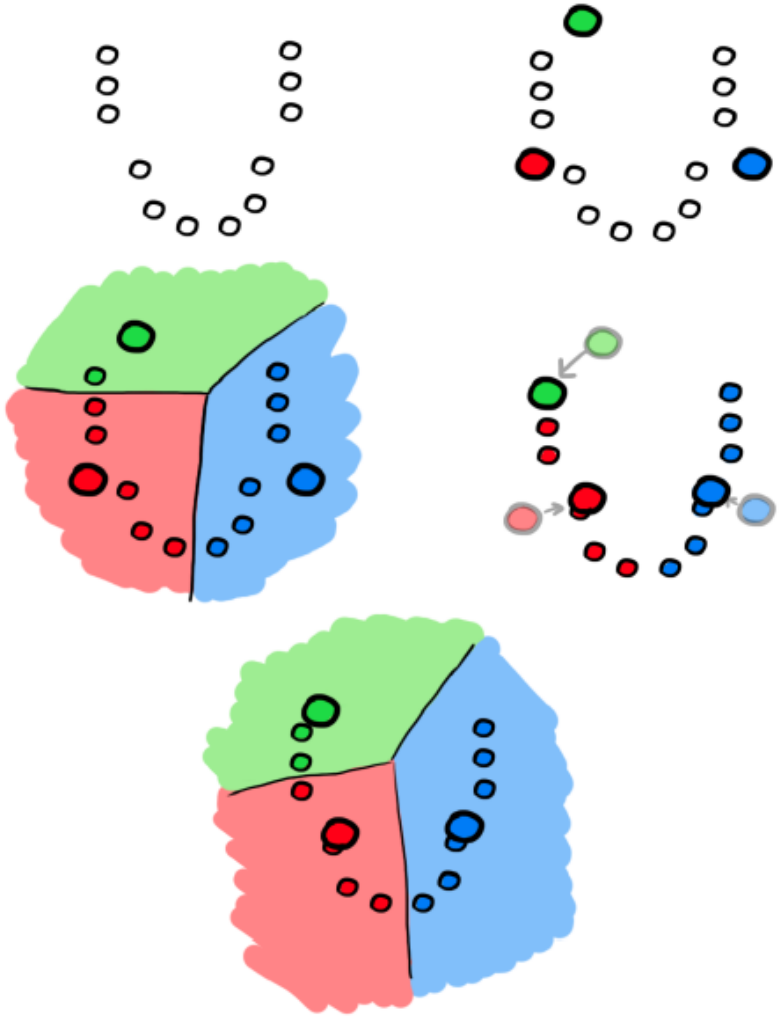
- **k-means clustering** partitions the observations into k clusters, so that each observation belongs to the cluster with the nearest **centroid**

K-means: the algorithm

1. Choose the number of clusters **k**.
2. Randomly generate k clusters and determine the **cluster centroids pc**
3. Repeat the following steps until some convergence criterion is met
   - **Assign** each point x to the nearest cluster centroid
   - **Recompute** the new cluster centroid

$$\mathbf{p}_c \leftarrow \frac{\sum_{\text{entities in cluster } c} \mathbf{x}}{\text{number of entities in cluster } c}.$$

SIMPLE AND FAST!



K-means algorithm in action (from top to bottom, left to right). Initial centroids are placed. Space is subdivided into portions close to the centroids (Voronoi diagram: each portion contains the points which have the given centroid as closest prototype).
New centroids are calculated. A new space subdivision is obtained.

Voronoi diagrams

- Each prototype p_c is assigned to a **Voronoi cell**
- A **cell** for the prototype p_c is made of all points closer to pc than to any other prototype
- Segments are all the points that are equidistant to the two nearest sites
- Voronoi nodes are points equidistant to three (or more) sites

Soft clustering, fuzzy membership

- In some cases assignments of entities to clusters are not crisp, but probabilistic or fuzzy
- The assignment of each entity is defined in terms of a probability associated with its membership in different clusters, so that values sum up to one (or fuzzy membership if a probabilistic interpretation is not valid)
- Cluster membership can be defined as a decreasing function of the dissimilarities:

$$\text{membership}(\mathbf{x}, c) = \frac{e^{-\delta(\mathbf{x},\mathbf{p}_c)}}{\sum_c e^{-\delta(\mathbf{x},\mathbf{p}_c)}}.$$

Soft clustering, centroids update

Online update:

- repeatedly consider an **entity x**, derive its current fuzzy cluster memberships and updates all prototypes so that **the closer prototypes tend to become even closer** to the entity x:

$$\begin{aligned} \Delta \mathbf{p}_c &= \eta \cdot \text{membership}(\mathbf{x}, c) \cdot (\mathbf{x} - \mathbf{p}_c); \\ \mathbf{p}_c &\leftarrow \mathbf{p}_c + \Delta \mathbf{p}_c. \end{aligned}$$

- the prototype is **pulled** by each entity to move along the vector (**x** - $\mathbf{p}_c$)

Soft clustering, centroids update(2)

Batch update:

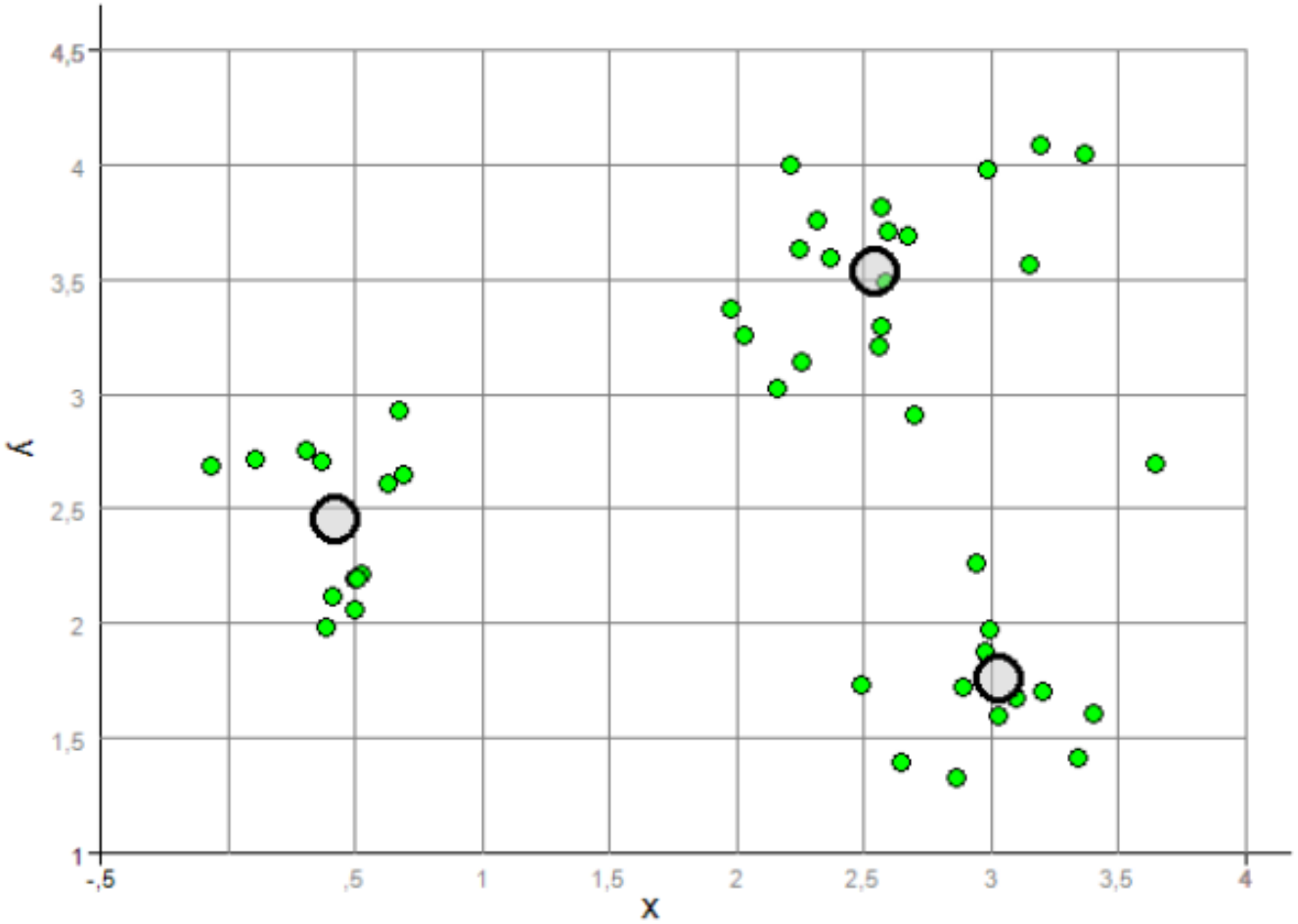
- *first* sum updated contributions over all entities and *then* proceed to update, as follows:

$$\mathbf{p}_c \leftarrow \mathbf{p}_c + \Delta_{total}\mathbf{p}_c.$$

Batch vs. online update

- When η increases, increasingly different results can be obtained

- The online update avoids summing all contributions before moving the prototype, and it is therefore suggested when the number of data items becomes very large



K-means clustering. Individual points and cluster prototypes are shown.

Gist

- Unsupervised learning deals with building models by using only input data, without resorting to classification labels.
- clustering aims at grouping similar cases in the same group
- The information to start the clustering can be given as **relationships** between couples of points (external representation, prototypes can be computed) or as **vectors** describing individual points (internal representation)

Gist(2)

- Objectives of clustering:

1. Information compression
2. Identification of global structure
3. Reducing cognitive overload by using prototypes

- The choice of the **similarity metric** is key
- It is a **multi-objective** task: similarity within clusters vs. dissimilarity between clusters

Gist(3)

- In **top-down** clustering one proceeds by selecting the desired number of classes and subdividing the cases
- K-means starts by positioning **K prototypes**, assigning cases to their closets prototypes, re-computing the prototypes as averages of the cases assigned to them, ...