MLCC 2018 - Clustering

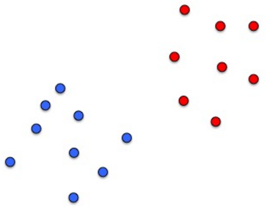
Lorenzo Rosasco
UNIGE-MIT-IIT

About this class

We will consider an *unsupervised setting*, and in particular the problem of clustering unlabeled data into "coherent" groups.

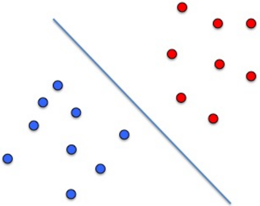
supervised learning

- "Learning with a teacher"
- Data set $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$
- $\hat{X} = (x_1, \dots, x_n)^\top \in \mathbb{R}^{n \times d}$ and $\hat{y} = (y_1, \dots, y_n)^\top$.



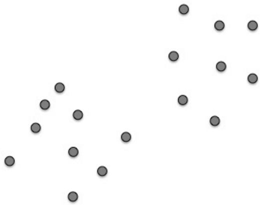
supervised learning

- "Learning with a teacher"
- Data set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$
- $\hat{X} = (x_1, \ldots, x_n)^\top \in \mathbb{R}^{n \times d}$ and $\hat{y} = (y_1, \ldots, y_n)^\top$.



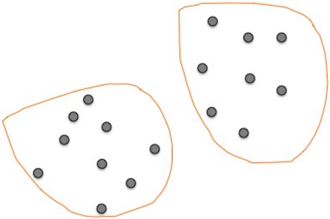
Unsupervised learning

- "Learning without a teacher"
- Data set $S = \{x_1, \ldots, x_n\}$ with $x_i \in \mathbb{R}^d$



Unsupervised learning

- "Learning without a teacher"
- Data set $S = \{x_1, \dots, x_n\}$ with $x_i \in \mathbb{R}^d$



Unsupervised learning problems

- Dimensionality reduction
- **Clustering**
- Density estimation
- Learning association rules
- Learning adaptive data representations
- ...

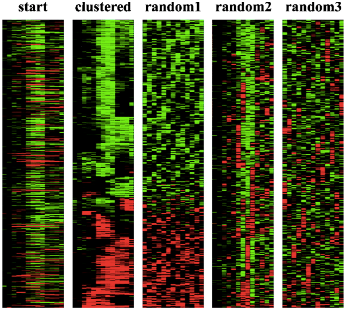
Supervised vs unsupervised methods

- In supervised learning we have a measure of success — based on a loss function and on a model selection procedure e.g., cross validation
- In unsupervised learning we don't !
  - hence many heuristics and the proliferation of many algorithms difficult to evaluate — lack of theoretical grounds

Clustering

- *Clustering* is a widely used technique for *data analysis*, with applications ranging from statistics, computer science, biology, social sciences....
- **Goal:**
  Grouping/segmenting a collection of objects into subsets or clusters.
  (Possibly also) arrange clusters into a natural hierarchy

Clustering examples



Michael B. Eisen et al. PNAS 1998;95:14863-14868

Clustering algorithms

- Combinatorial algorithms - directly from data $\{x_i\}_{i=1}^n$ + some notion of *similarity* or *dissimilarity*
- Mixture models - based on some assumption on the underlying probability distribution

Combinatorial clustering

- We assume some knowledge on the number of clusters $K \leq n$.
  Goal: associate a cluster label $k = \{1, \ldots, K\}$ with each datum, by defining an encoder $\mathcal{C}$ s.t.

$$k = \mathcal{C}(x_i)$$

- We look for an encoder $\mathcal{C}^*$ that achieves the goal of clustering data, according to some specific requirement of the algorithm and based on data pairs dissimilarities

Combinatorial clustering

- *Criterion:* assign to the same cluster similar/close data
- We may start from the following "loss" or energy function (within class):

$$W(\mathcal{C}) = \frac{1}{2} \sum_{k=1}^{K} \sum_{\mathcal{C}(i)=k} \sum_{\mathcal{C}(i')=k} d(x_i, x_{i'})$$

- $\mathcal{C}^* = \arg\min W(\mathcal{C})$
- Unfeasible in practice!

$$S(N,K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^n$$

and notice that $S(10,4) \sim 34K$ while $S(19,4) \sim 10^{10}$

K-means algorithm

It refers specifically to the Euclidean distance.

- initialize cluster centroids m_k $k = 1, \ldots, K$ at random
- repeat until convergence
    1. assign data to centroids $\mathcal{C}(x_i) = \arg\min_{1 \leq k \leq K} ||x_i - m_k||^2$
    2. update centroids

K-means functional

K-means corresponds to minimizing the following function

$$J(\mathcal{C}, m) = \sum_{k=1}^{K} \sum_{\mathcal{C}(i)=k} ||x_i - m_k||^2$$

The algorithm is an alternating optimization procedure, with convergence guarantees in practice (no rates).

The function J is not convex, thus K-means is not guaranteed to find a global minimum.

Computational cost

1. data assignment $O(Kn)$
2. cluster centers updates $O(n)$

K-means

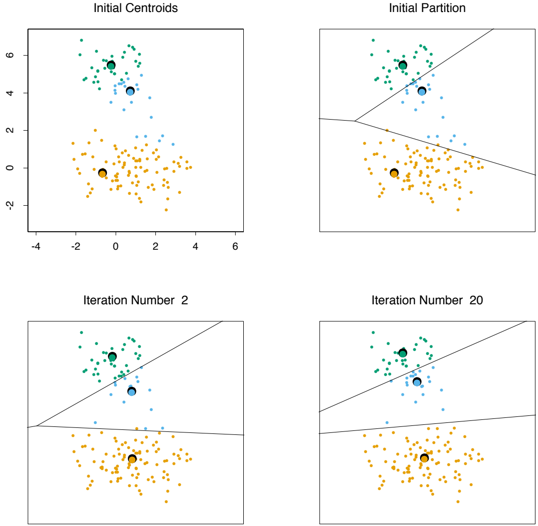


Figure from Hastie, Tibshirani, Friedman

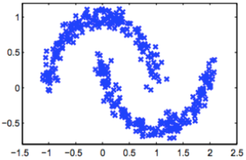
Example Vector Quantization

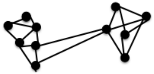
FIGURE 14.9. *Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a* 1024×1024 *grayscale image at 8 bits per pixel. The center image is the result of* 2×2 *block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel*

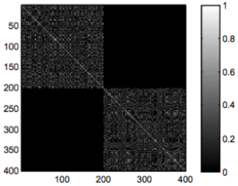
Figure from Hastie, Tibshirani, Friedman

Spectral clustering - similarity graph

- A set of unlabeled data $\{x_i\}_{i=1}^n$ and some notion of similarity between data pairs s_{ij}
  - We may represent them as a *similarity graph* $G = (V, E)$







  - Clustering can be seen as a graph partitioning problem

Spectral clustering - graph notation

$G = (V, E)$ undirected graph

- V : data correspond to the vertices
- E : **Weighted adjacency matrix** $W = (w_{ij})_{i,j=1}^n$ with $w_{ij} \geq 0$.
  W is symmetric $w_{ij} = w_{ji}$, as G is undirected.
- *Degree of a vertex:* $d_i = \sum_{j=1}^n w_{ij}$
  Degree matrix: $D = diag(d_i)$
- *Sub-graphs:*
  $A, B \subset V$ then $W(A, B) = \sum_{i \in A, j \in B} w_{ij}$
  Subgraph size:
  - $|A|$ number of vertices
  - $\mathrm{vol}(A) = \sum_{i \in A} d_i$

Spectral clustering - how to build the graph

We use the available pairwise similarities s_{ij}

- *ϵ-neighbourhood graph*: connect vertices whose similarity is larger than ϵ
- *KNN graph*: connect vertex v_i to its K neighbours. Not symmetric!
- *fully connected graph:* $s_{ij} = \exp(-d_{ij}^2/2\sigma^2)$
  d is the Euclidean distance, $\sigma \geq 0$ controls the width of a neighborhood

Spectral clustering - how to build the graph

- n can be very large, it would be preferable if W was sparse
- In general it is better some notion of locality

$$w_{ij} = \begin{cases} s_{ij} & \text{if } j \text{ is a KNN of } i \\ 0 & \text{otherwise} \end{cases}$$

Spectral clustering - graph Laplacians

Unnormalized graph Laplacian: $L = D - W$
Properties:

- For all $f \in \mathbb{R}^n$

$$f^\top L f = \frac{1}{2} \sum_{ij=1}^{n} w_{ij}(f_i - f_j)^2$$

$$\begin{aligned}
f^\top L f &= f^\top D f - f^\top W f \\
&= \sum_i d_i f_i^2 - \sum_{i,j} f_i f_j w_{ij} \\
&= \frac{1}{2}\left(\sum_i (\sum_j w_{ij}) f_i^2 - 2 \sum_{ij} f_i f_j w_{ij} + \sum_j (\sum_i w_{ij}) f_j^2 \right) = \\
&= \frac{1}{2} \sum_{ij} w_{ij}(f_i - f_j)^2
\end{aligned}$$

Spectral clustering - graph Laplacians

Unnormalized graph Laplacian: $L = D - W$

- For each vector $f \in \mathbb{R}^n$

$$f^\top L f = \frac{1}{2} \sum_{ij=1}^{n} w_{ij}(f_i - f_j)^2$$

  The graph Laplacian measures the variation of f on the graph ($f^\top L f$ small if close points have close function values f_i)
- L is symmetric and positive semi-definite
- The smallest eigenvalue of L is 0 and its corresponding eigenvector is a vector of ones
- L has N non negative real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$

Spectral clustering - graph Laplacians

Unnormalized graph Laplacian: $L = D - W$

- For each vector $f \in \mathbb{R}^n$

$$f^\top L f = \frac{1}{2} \sum_{ij=1}^{n} w_{ij}(f_i - f_j)^2$$

  The graph Laplacian measures the variation of f on the graph ($f^\top L f$ small if close points have close function values f_i)
- L is symmetric and positive semi-definite
- The smallest eigenvalue of L is 0 and its corresponding eigenvector is a vector of ones
- L has N non negative real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$

Laplacian and clustering: the multiplicity k of $\lambda_0 = 0$ equals the number of connected components in the graph

Spectral clustering - graph Laplacians

Unnormalized graph Laplacian:

$$L = D - W$$

Normalized graph Laplacians:

$$L_{n1} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

$$L_{n2} = D^{-1} L = I - D^{-1} W$$

A spectral clustering algorithm

- Graph Laplacian
  - compute the Unnormalized Graph Laplacian L (unnormalized algorithm)
  - compute a Normalized Graph Laplacian L_{n1} or L_{n2} (normalized algorithm)
- compute the first k eigenvectors of the Laplacian (k number of clusters to compute)
- let $U_k \in \mathbb{R}^{n \times k}$ be the matrix containing the k eigenvectors as columns
- $y_j \in \mathbb{R}^k$ be the vector obtained by the j-th row of U_k $j = 1 \ldots n$. Apply k-means to $\{y_j\}$

A spectral clustering algorithm

Computational cost

- Eigendecomposition $O(n^3)$
- It may be enough to compute the first k eigenvalues/eigenvectors. There are algorithms for this

Example

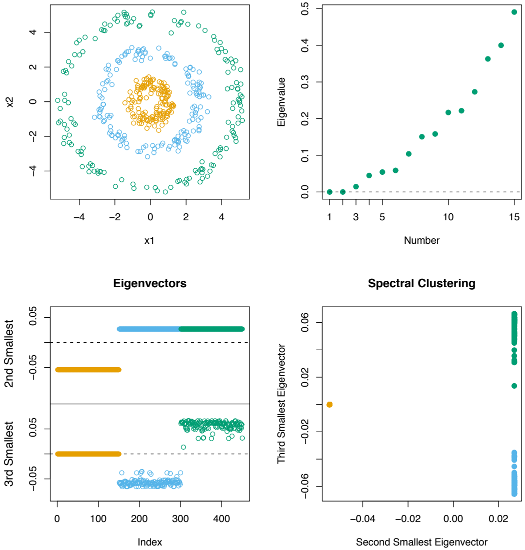


Figure from Hastie, Tibshirani, Friedman

The number of clusters

eigengap heuristic

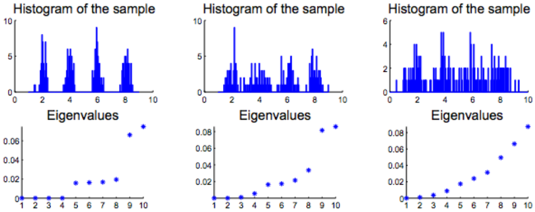


Figure from Von Luxburg tutorial

Semi-supervised learning

Laplacian-based regularization algorithms (Belkin et al. 04)
Set of labeled examples: $\{(x_i, y_i)\}_{i=1}^n$
Set of unlabeled examples: $\{(x_j)\}_{j=n+1}^{n+u}$

$$f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \lambda_A \|f\|^2 + \frac{\lambda_I}{u^2} f^T L f$$

Wrapping up

In this class we introduced the concept of data clustering and sketched some of the best known algorithms

Ulrike Von Luxburg - *A tutorial on Spectral Clustering*