Nonhierarchical clustering methods

Mohamed Nadif

Université Paris Descartes, France

Outline

1. **Introduction**
   - Cluster Analysis
2. *k*-means algorithm
   - *k*-means and its variants
   - Principal points to be retained
   - Principal points to be retained
3. Extensions of *k*-means
   - Sequential methods
   - Self-Organizing-Mapping
   - Dynamic clusters method

Clustering

- Aim: It seeks to obtained a reduced representation of the initial data
- Organization of data into homogeneous subsets "clusters" or "classes"
- Terminology can depend on the field:

Structure of clustering

- It can take different forms: partitions, sequence of encased partitions or hierarchical, overlapping clusters, clusters with high density, fuzzy clusters.
- In this chapter we focus on the hierarchical methods

Characteristics of these methods

- This section is devoted to the partitioning methods or nonhierarchical clustering

Outline

1. Introduction
   - Cluster Analysis

2. *k*-means algorithm
   - *k*-means and its variants
   - Principal points to be retained
   - Principal points to be retained

3. Extensions of *k*-means
   - Sequential methods
   - Self-Organizing-Mapping
   - Dynamic clusters method

Description

- This section is devoted to the partitioning methods or nonhierarchical clustering. We keep the previous notation and we begin by describing the well-know k-means when the set to classify Ω is measured by p continuous variables
- To look for the optimal partition $\mathbf{z}$ it suffices to minimize the within-cluster variance $W(\mathbf{z})$

$$W(\mathbf{z}) = \sum_{k=1}^{K} \sum_{i \in \mathbf{z_k}} ||\boldsymbol{x}_i - \overline{x}_{\mathbf{z_k}}||^2.$$

which is equivalent to maximize the between-cluster variance

$$B(\mathbf{z}) = \sum_{k=1}^{K} \pi_k ||\overline{x}_{\mathbf{z_k}} - \overline{x}||^2,$$

where π_k is the weight of the cluster z_k and $\overline{x}$ is the vector center of all data. This equivalence is due to the decomposition of the total variance I of data

$$I = W(\mathbf{z}) + B(\mathbf{z})$$

Description of k-means

- The one-parameter optimization $W(\mathbf{z})$ is equivalent to the optimization of the two-parameter optimization $W(\mathbf{z},\boldsymbol{\mu})$

$$W(\mathbf{z},\boldsymbol{\mu}) = \sum_{k=1}^{K} \sum_{i \in z_{\mathbf{k}}} ||\boldsymbol{x}_i - \boldsymbol{\mu}_k||^2, \tag{1}$$

  where $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$ with $\boldsymbol{\mu}_k$ from $\mathbb{R}^p$ represents the center or prototype of the cluster z_k.

- This optimization can be carried out by the k-means algorithm and the principal steps of the k-means are the following:

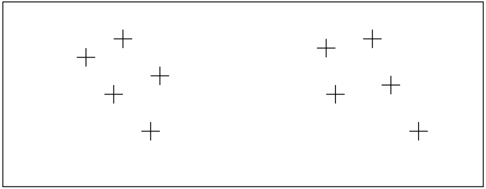
  1. Randomly select K objects of Ω which form the K first cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$.
  2. While not convergence
     1. assign each object of Ω to the cluster with the nearest cluster mean. If this one is not unique the object is assigned to the cluster with the smallest subscript.
     2. The cluster means computed become the new cluster means.

In the iteration process k-means yields a sequence $\boldsymbol{\mu}^{(0)}, \mathbf{z}^{(1)}, \boldsymbol{\mu}^{(1)}, \mathbf{z}^{(2)}, \ldots$ of partitions and centers with decreasing the values of the criterion until the convergence at the minimum value

Description of *k*-means

- We illustrate the different steps of *k*-means by applying it with $K = 2$ on a simple set Ω of 10 objects located in plan as depicted in a rectangle

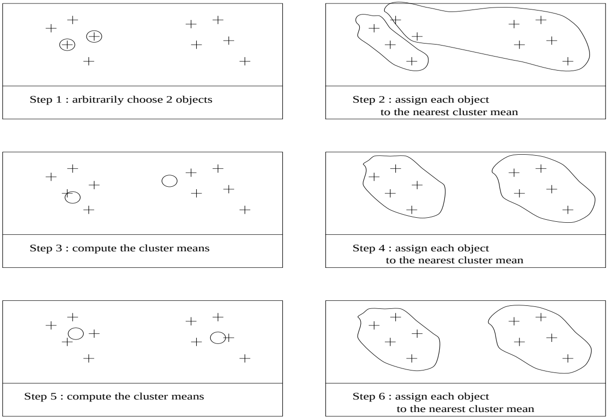
Example of 10 objects to classify



Description of *k*-means

- The *k*means algorithm can then be summarized in the following way **??**

Process of *k*-means.



Step 1 : arbitrarily choose 2 objects

Step 2 : assign each object to the nearest cluster mean

Step 3 : compute the cluster means

Step 4 : assign each object to the nearest cluster mean

Step 5 : compute the cluster means

Step 6 : assign each object to the nearest cluster mean

The process terminates and this algorithm will not change any more the results: The algorithm converges. Note that the obtained partition corresponds to the observable structure in two clusters

Within-cluster criterion

- It corresponds to the famous sum-of-squares criterion (SSQ)
- Different approaches in clustering are based on this criterion but under different forms due to different hypothesis

First Hypothesis

- $\mathbf{z}$ is a known partition and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ as a realization of a random vector $\mathbf{x}$ with f its density on $\mathbb{R}^p$.
  - The problem is to look for the partition $\mathbf{z}$ in $\mathbb{R}^p$ minimizing :

$$W(\mathbf{z}) = \sum_k \int_{\mathbf{z}_k} ||\boldsymbol{x} - \mathbb{E}_{\mathbf{z}_k}(X)||^2 dP(x)$$

As $P(x < X < x + dx) = f(x)dx$, $W(\mathbf{z})$ is equivalent to

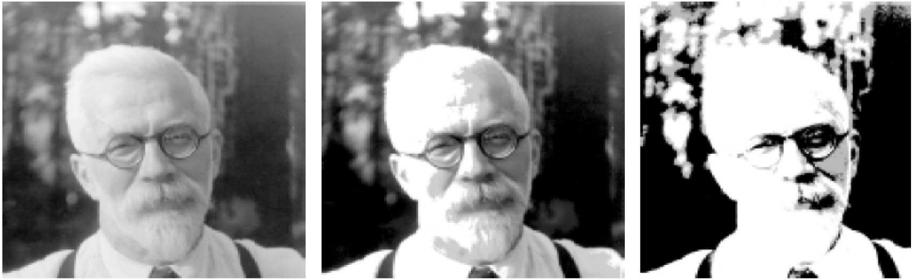
$$\begin{aligned} W(\mathbf{z}, \boldsymbol{\mu}) &= \sum_k \int_{\mathbf{z}_k} ||\boldsymbol{x} - \mathbb{E}_{\mathbf{z}_k}(X)||^2 f(\boldsymbol{x}) d\boldsymbol{x} \\ &= \sum_k \int_{\mathbf{z}_k} f(\boldsymbol{x}) ||\boldsymbol{x} - \boldsymbol{\mu}_k||^2 d\boldsymbol{x} \qquad (2) \end{aligned}$$

  - In $\mathbb{R}$, this formulation has been provided in the framework of optimum proportional stratified sampling (Dalenius, 1950). Even if the k-means algorithm was not used but another called the *shooting* algorithm, this later needs two principal steps of k-means

Within-cluster criterion

- Different extensions to this algorithm were proposed and successfully applied in image compression. The figure(photo-Fisher) illustrates an application of the LLoyd algorithm in the context of scalar quantization.

Example of scalar quantization.



Within-cluster criterion

- In addition, the optimization of SSQ2 was considered in multidimensional case $\mathbb{R}^p$, and the first to propose the k-means explicitly was Steinhauss (1956)
- Actually, the k-means version commonly used is due to Forgy (1965)
- After several works concerned on different variants of k-means and sometimes under different names algorithms were proposed, see for instance (Bock, 2007)
- We can cite the *nuées dynamiques* or dynamic clusters method (Diday, 1971), iterated minimum-distance partition method (Bock, 1974)
- Another approach which consists to consider the data as a sample is appeared with MacQueen (1967) and a stochastic version of k-means is performed and inspired the Kohonen maps

- Before performing a cluster analysis on coordinate data, it is necessary to consider scaling or transforming the variables since variables with large variances tend to have more effect on the resulting clusters than those with small variances. Other transformations can be used according the nature of data, this step is very important and can be present in the metric as we will see later on.
- If our aim is to find the couple $(\mathbf{z}, \boldsymbol{\mu})$ minimizing the criterion W, the k-means algorithm does not provide necessarily the best result, but just a sequence of couples whose the value of criterion is going to decrease and we obtain a local optimum. Then and as in practice the convergence is reached very quickly (often less than 10 iterations even with large data) in order to obtain an interesting solutions, the user needs to run k-means several times and choose the best result according the SSQ criterion.
- The k-means algorithm can use an L_m clustering criterion instead of the least-squares L_2 criterion. Note that values of m less than 2 reduce the effect of outliers on the cluster centers compared with least-squares criterion.
- In general, the criterion is not independent of the number of classes. For example, the partition into n classes, where each object forms a singleton cluster, has a null within-cluster criterion and therefore the optimal partition is without interest. It is then necessary to fix a priori the number of classes.

- If this number is not known, several solutions allowing to solve this very difficult problem are used. For example, the best partition is sought for several numbers of classes and we study the decrease of the criterion according the number of classes to select the number of classes by using the scree plot and choosing an elbow. Indeed, the quality of a partition can be evaluated by the *Rsquare* (RSQ)

  $$RSQ = 1 - \frac{W}{I} = \frac{B}{I}$$

  The *k*-means has a very low computational complexity which translates directly into a high speed, it suffices then to run *k*-means with different number of clusters and use the elbow method
- different methods (see course 3)
- Knowing that according to starting points chosen, the results will be different, it remains with to exploit these different results. Several solutions were proposed: we run the *k*-means several times by initiating with different random initializations. Several strategies are then possible.
  - We select a *good* initialization with supplementary informations or with an automatic procedure (points strongly distant, regions with high density, etc.)
  - We should however make a compromise between the necessary time to the research of the initial configuration and that necessary time for the algorithm itself

- Link between *k*-means and the Ward method: The two methods are similar in that they both attempt to the within-cluster variance (Wong, 1982).

  1. Apply *k*-means to cluster Ω into fifty clusters, for example. In practice, this number depending on the size of data can be taken equal to $n^{\frac{1}{3}}$
  2. Run the Ward method on these obtained cluster means
  3. From the dendrogram we propose a number of clusters by using the SPRSQ criterion
  4. Eventually, apply *k*-means on the obtained clusters to improve SPRSQ

- Fisher's method (1958): Note that there exist some situations for which we have effective algorithms allowing to find global optimum. It is the case where there is an order constraint on the partitions. This constraint can be implicit (for instance, when the data are in $\mathbb{R}$) or explicit (for instance,constraint imposed by the user). We can then use a dynamic algorithm of programming such as the Fisher's algorithm which provides the global optimum.

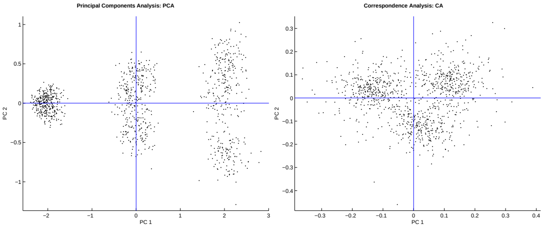
Importance of the metric

- Let be **x** a data matrix that consists of a set of objects described by 3 continuous variables x, y and z Naturally, we can use the Standardized Euclidean distance on standardized data but sometimes the clustering on the profiles (row percents) are more adapted in certain contexts and as the values of this data matrix are all positive we can use the metric χ^2

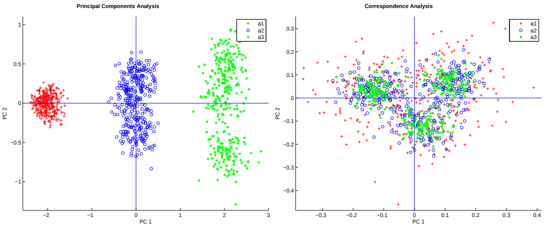
Extract of data

x	y	z	x	y	z	x	y	z
37	31	40	117	132	142	201	240	194
35	26	29	166	118	117	205	266	205
42	44	25	115	126	153	178	212	256
43	20	28	152	105	115	240	223	172
32	26	43	114	119	162	195	199	256
44	32	27	109	109	91	190	223	203
31	38	29	136	150	95	277	206	190
28	47	49	100	132	152	212	198	259
..	..	..	...	...	...	...	...	...

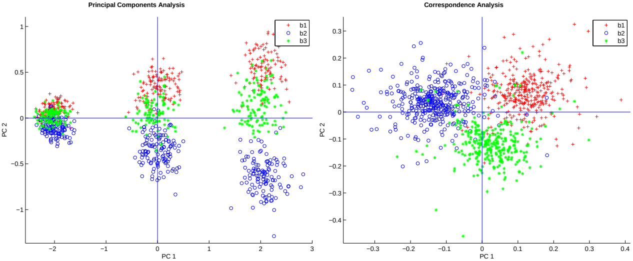
Projection of objects on the factorial planes spawned by the first and second axes by PCA and CA



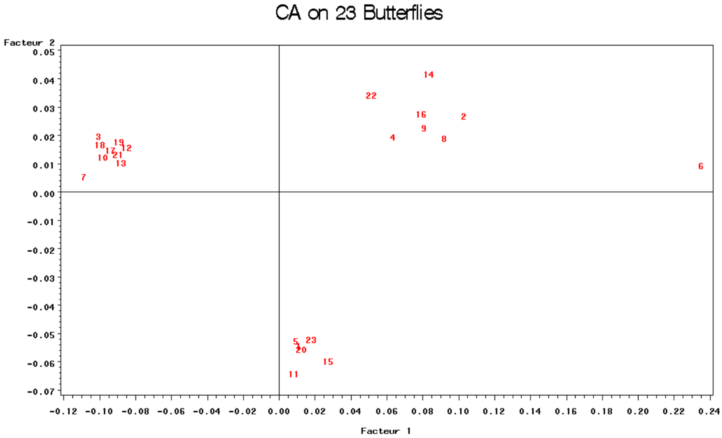
Projection of clusters on the factorial planes spawned by the first and second axes by PCA and CA.



Projection of clusters on the factorial planes spawned by the first and second axes by PCA and CA.



Example of Butterflies



k-means by using the χ^2 distance

- This table can be considered as the contingency table
- Why ?

Contingency table and χ^2

There are several measures of association and the most employed is the chi-square χ^2. This criterion, used for example in the correspondence analysis

$$\chi^2(I,J) \quad = \quad \sum_{i,j}\frac{(x_{ij}-\frac{x_{i.}x_{.j}}{N})^2}{\frac{x_{i.}x_{.j}}{N}} = N\sum_{i,j}\frac{(f_{ij}-f_{i.}f_{.j})^2}{f_{i.}f_{.j}}.$$

- χ^2 usually provides statistical evidence of a significant association, or dependence, between rows and columns of the table. It represents the deviation between the theoretical frequencies $f_{i.}f_{.j}$, that we would have if I and J were independent, and the observed frequencies f_{ij}
- If I and J are independent, the χ^2 will be zero and if there is a strong relationship between I and J, the χ^2 will be high
- A significant chi-square indicates a departure from row or column homogeneity and can be used as a measure of heterogeneity. Then, the chi-square can be used to evaluate the quality of a partitions of I or $\mathbf{w}$ of J
- Associated $\chi^2(\mathbf{z},J)$ of the contingency table with K rows in making the sum of rows of each cluster
- We have $\chi^2(I,J) \geq \chi^2(\mathbf{z},J)$ and the objective is to find the partitions $\mathbf{z}$ which minimizing this loss, i.e. which maximizes $\chi^2(\mathbf{z},J) = N\sum_{k,\ell}\frac{(f_{\mathbf{k}\mathbf{j}}-f_{\mathbf{k}.}f_{.\mathbf{j}})^2}{f_{\mathbf{k}.}f_{.\mathbf{j}}}$

Clustering of contingency table

- The Euclidean classical distance in not appropriated
- Clusters of objects is not informative than clustering of the profiles
- The χ^2 is more adapted

Transformation of data

- The data matrix $\mathbf{x}$ is a $n \times p$ matrix defined by $\mathbf{x} = \{(x_{ij}); i \in I, j \in J\}$ where I is a categorical variable with n categories and J a categorical variable with p categories
- We denote the row and columns total of $\mathbf{x}$ by $x_{i.} = \sum_{j=1}^{p} x_{ij}$ and $x_{.j} = \sum_{i=1}^{n} x_{ij}$ and the overall total simply by $N = \sum_{ij} x_{ij}$.
- We denote $\{(f_{ij} = x_{ij}/N); i \in I, j \in J\}$
- the marginal frequencies $f_{i.} = \sum_j f_{ij}$ and $f_{.j} = \sum_i f_{ij}$
- The row profiles $f_J^i = (f_{i1}/f_{i.}, \ldots, f_{ip}/f_{i.})$
- The average row profile $f_J = (f_{.1}, \ldots, f_{.p})$

Time-budget data matrix (Jambu 1976)

	prof	tran	home	child	shop	wash	meal	sleep	tv	leis
maus	610	140	60	10	120	95	115	760	175	315
waus	475	90	250	30	140	120	100	775	115	305
wnaus	10	0	495	110	170	110	130	785	160	430
mnsus	615	141	65	10	115	90	115	765	180	305
wnsus	179	29	421	87	161	112	119	776	143	373
msus	585	115	50	0	150	105	100	760	150	385
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
mnsea	652	133	134	22	68	94	102	762	122	310
wnsea	434	77	431	60	117	88	105	770	73	229
msea	627	148	68	0	88	92	86	770	58	463
wsea	433	86	296	21	128	102	94	758	58	379

I : types of population and J : variety of activities, x_{ij}: amount of time spent on a variety of activities j by i during a given time period j

Notation

- The choice of χ^2 metric is justified for several reasons, in particular because of the similar role played by each of the two dimensions in the analyzed table, and also because of the property of distributional equivalence, which implies stable results when agglomerating elements with similar profiles
- Each row i corresponds to a point vector $\mathbb{R}^p$ defined by the profile f_{iJ} weighted by the marginal frequency $f_{i.}$
- The maximization of $\chi^2(\mathbf{z}, J)$ can be viewed as the minimization of a criterion depending on the partition and the centers of clusters
- $\mathbf{z}$ is a partition of the rows, we can define the frequencies $f_{kj} = \sum_{i \in z_{\mathbf{k}}} f_{ij}$ and the average row profile of the cluster z_k is defined by the vector $f_{kJ} = (\frac{f_{\mathbf{k1}}}{f_{\mathbf{k}.}}, \ldots, \frac{f_{\mathbf{kp}}}{f_{\mathbf{k}.}})$ where $f_{k.} = \sum_{j=1}^{p} f_{kj}$

SSQ criterion

- With this representation, the total of squared distances T, the between-cluster sums of squares $B(\mathbf{z})$ and the within-cluster sums of squares take the forms

$$T = \sum_{i=1}^{n} f_{i;} d^2(f_{iJ}, f_J) \ , \ B(\mathbf{z}) = \sum_{i=1}^{n} f_{k.} d^2(f_{kJ}, f_J) = \frac{1}{N}\chi^2(\mathbf{z}, J),$$

and

$$W(\mathbf{z}) = \sum_{k=1}^{K} \sum_{i \in z_{\boldsymbol{k}}} f_{i.} d^2(f_{iJ}, f_{kJ}).$$

- The traditional relation $T = W(\mathbf{z}) + B(\mathbf{z})$ leads to the following relation:

$$\chi^2(I, J) = NW(\mathbf{z}) + \chi^2(\mathbf{z}, J).$$

- The term $NW(z)$ therefore represents the information lost when grouping the elements according to the partition $\mathbf{z}$, and $\chi^2(\mathbf{z}, J)$ corresponds to the information which is preserved
- Looking for the partition maximizing the criterion $\chi^2(\mathbf{z}, J)$ is equivalent to looking for the partition minimizing $W(\mathbf{z})$ or $W(\mathbf{z}, \mathbf{a})$
- To minimize this criterion it is possible to apply k-means to the set of profiles with the χ^2 metric. The iterative algorithm maximizing locally $\chi^2(\mathbf{z}, J)$

Clustering of Categorical data

- Generally we apply the clustering to a particular indicator matrix
- Let a variable with 3 categories $1, 2, 3 \Rightarrow (1, 0, 0), (0, 1, 0), (0, 0, 1)$, then the matrix has the number of rows equal to the total number of objects and the number of columns equal to the sum of all categories corresponding to all variables
- As before the χ^2 is the more appropriate metric
- We can apply the *k*means with the χ^2 metric

Example

	a	b
1	1	2
2	3	2
3	2	3
4	1	1
5	1	2
6	3	2
7	3	3
8	1	1
9	2	2
10	2	3

	a1	a2	a3	b1	b2	b3
1	1	0	0	0	1	0
2	0	0	1	0	1	0
3	0	1	0	0	0	1
4	1	0	0	1	0	0
5	1	0	0	0	1	0
6	0	0	1	0	1	0
7	0	0	1	0	0	1
8	1	0	0	1	0	0
9	0	1	0	0	1	0
10	0	1	0	0	0	1

The distance takes the following form:

$$d_{\chi^2}(i, i') = \sum_{j=1}^{p} \frac{1}{f_{.j}} \left(\frac{f_{ij}}{f_{i.}} - \frac{f_{i'j}}{f_{i'.}}\right)^2$$

Outline

1 Introduction

- Cluster Analysis

2 k-means algorithm

- k-means and its variants
- Principal points to be retained
- Principal points to be retained

3 Extensions of k-means

- Sequential methods
- Self-Organizing-Mapping
- Dynamic clusters method

Sequential methods

- The k-means algorithm has different extensions in order to apply it on different types of data such as the sequential data
- The online variants of k-means are particularly adequate when all the data to be classified are not available in the beginning
- The parameters defining the classes can then be adjusted when a new data comes as a continuous stream without too many calculations
- Unlike k-means the objects concerning by the step assignation are randomly selected and the update of cluster means is realized after each assignation of one object
- More precisely, at the (t)th iteration, the object $\boldsymbol{x}_i$ is randomly selected, then we determine the nearest prototype $\boldsymbol{\mu}_k^{(t)}$ which becomes after assignation of $\boldsymbol{x}_i$ equal to

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\boldsymbol{x}_i + n_k^{(t)} \cdot \boldsymbol{\mu}_k^{(t)}}{n_k^{(t+1)}},$$

where $n_k^{(t)}$ represents the cardinality of the cluster $z_k^{(t)}$ and $n_k^{(t+1)} = n_k^{(t)} + 1$.

$$\boldsymbol{\mu}_k^{(t+1)} = \boldsymbol{\mu}_k^{(t)} + \frac{1}{n_k^{(t+1)}}(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t)}),$$

- The mean can be generalized

$$\boldsymbol{\mu}_k^{(t+1)} = \boldsymbol{\mu}_k^{(t)} + \varepsilon(t)(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t)}),$$

  where $\varepsilon(t)$ is a decreasing learning coefficient

- The usual hypothesis on the adaptation parameter to get almost sure results is then (conditions of Robbins-Monro):

$$\sum_t \varepsilon(t) = +\infty \text{ and } \sum_t \varepsilon(t)^2 < +\infty$$

- This formulation of the cluster means can be extended and constitutes the version of other algorithms such as the well-known *Self-Organizing-Mapping*

Self-Organizing-Mapping

- *Self-Organizing-Mapping* or SOM a type of clustering, inspired by neuroscience, that has been introduced in Kohonen (1982).
- In the SOM literature, we refer to the clusters by the nodes or neurons and each of them has a weight in $\mathbb{R}^p$, these weights refer to the cluster means
- The principal advantage of SOM that is preserves the topology clustering. Generally, the neurons are arranged as one or two-dimensional rectangular grid preserving relations between the objects called also units
- SOM offers an good tool to visualize clusters and evaluate their proximity in a reduced space
- Unlike k-means and AHC, the previous expression of the cluster means or the weight of a neuron k becomes in the SOM context

$$\boldsymbol{\mu}_k^{(t+1)} = \boldsymbol{\mu}_k^{(t)} + \varepsilon(t) \times h(k,\ell)(\mathbf{x}_i - \boldsymbol{\mu}_k^{(t)}), \qquad (3)$$

  where $h(k,\ell)$ is the neighborhood function between the neuron k whose weight $\boldsymbol{\mu}_k^{(t)}$ is the most similar to $\mathbf{x}_i$ (the best matching unit or the winner) and the other neurons ℓ with weights $\boldsymbol{\mu}_\ell$ close enough to $\boldsymbol{\mu}_k^{(t)}$

The neighborhood function

- This function can take different forms, it evaluates the proximity between the winner k and the neuron ℓ located in a reduced space generally in $\mathbb{R}^2$ with the position r_k and r_ℓ
- In the early publications about SOMs, $h(k,\ell)$ was defined by: $h(k,\ell) = 1$ if $d(k,\ell) \leq \lambda$ and 0 otherwise
- Gaussian function is a common choice $h(k,\ell) = exp(-\frac{\alpha||r_k - r_\ell||^2}{2\sigma_h(t)})$ where $\sigma_h(t)$ controls the width of the neighborhood of h. This function h is declined because $\sigma_h(t)$ during the training process as well as the learning rate $\varepsilon(t)$
- With this grid moving during the iterations of SOM, we obtain a partition and a visualization of the clusters as in factorial plan from PCA, except this representation is not linear because it is not an orthogonal projection.
- The different steps of SOM are similar than the steps of k-means. In addition, two versions batch and online can be used. The first one performs the assignment and update steps for all data units at once and the second process as the MacQueen algorithm.
- As k-means, SOM requires to fix the number of clusters (nodes of the grid), and a choice of initialization. PCA appears an attractive and interesting approach
- As the numbers of nodes is higher, the number of clusters can be assess by applying AHC algorithm with appropriated agglomerative criterion on these nodes

Example: Iris data

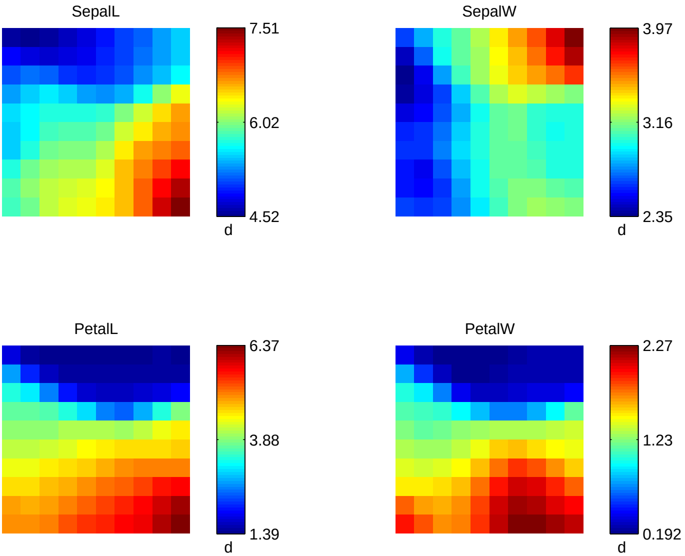
- From the well-know Iris flower data (Fisher, 1936) which consists of 50 samples of three species of Iris flowers (Setosa, Virginica and Versicolor). Four features were measured from each sample: length and width of sepal and petal, we apply SOM with 10 × 10 clusters

Iris on grid 10×10

Labels

	Setosa Setosa	Setosa Setosa Setosa	Setosa Setosa Setosa	Setosa Setosa	Setosa Setosa	Setosa Setosa Setosa Setosa	Setosa Setosa Setosa		Setosa Setosa Setosa Setosa Setosa
			Setosa Setosa Setosa Setosa Setosa	Setosa Setosa Setosa	Setosa Setosa Setosa	Setosa Setosa	Setosa Setosa	Setosa Setosa Setosa	Setosa Setosa
Versicolor Versicolor Versicolor	Versicolor			Setosa	Setosa	Setosa Setosa	Setosa Setosa		
	Versicolor Versicolor	Versicolor Versicolor							
Versicolor Versicolor	Versicolor Versicolor	Versicolor Versicolor Versicolor	Versicolor	Versicolor Versicolor Versicolor			Versicolor Versicolor	Versicolor	Versicolor Versicolor Versicolor
Versicolor	Versicolor		Versicolor	Versicolor	Versicolor	Versicolor	Versicolor	Versicolor	Versicolor Versicolor Versicolor
Versicolor Versicolor Virginica	Versicolor		Versicolor Versicolor Versicolor Versicolor		Versicolor Virginica			Versicolor	Versicolor
	Versicolor Versicolor		Virginica	Virginica Virginica		Virginica Virginica Virginica	Virginica Virginica Virginica	Virginica	Virginica
	Virginica	Versicolor Virginica	Virginica		Virginica Virginica		Virginica	Virginica Virginica	Virginica Virginica
Virginica Virginica Virginica Virginica Virginica		Versicolor Virginica	Virginica Virginica	Virginica Virginica Virginica	Virginica Virginica	Virginica Virginica Virginica Virginica	Virginica Virginica Virginica	Virginica	Virginica Virginica Virginica Virginica Virginica Virginica

Grid by variable



Dynamic clusters method

- The *nuées dynamiques* (Diday, 1971) is based on the quite powerful idea that the cluster centers are not necessarily cluster means elements of $\mathbb{R}^p$. Then he proposed to replace them by centers being able to take different forms adapted to the problem to be solved.
- Let $\mathbb{L}$ be the set of centers, and $D : \Omega \times L \rightarrow \mathbb{R}^+$, a measure of dissimilarity between objects of Ω and the centers of $\mathbb{L}$. The aim is to look for a partition of Ω into K clusters minimizing the following criterion

$$C(\mathbf{z}, L) = \sum_{k=1}^{K} \sum_{x \in z_{\boldsymbol{k}}} D(\boldsymbol{x}, \boldsymbol{\lambda}_k)$$

  where $\mathbf{z} = (z_1, \ldots, z_K)$ and $L = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_K)$ with $\boldsymbol{\lambda}_k \in \mathbb{L}$
- If $\Omega \subset \mathbb{R}^p$, $\mathbb{L} = \mathbb{R}^p$ and $D(\mathbf{x}, \boldsymbol{\lambda}) = d^2(\boldsymbol{x}, \boldsymbol{\lambda})$ then $C(\mathbf{z}, L) = W(\mathbf{z}, \boldsymbol{\mu})$.
- Like the k-means, to tackle the minimization of $C(\mathbf{z}, L)$ we can use an alternating optimization method based on
  1. Compute $\mathbf{z}^{(t+1)}$ minimizing $C(., L^{(t)})$
  2. Compute $L^{(t+1)}$ minimizing $C(\mathbf{z}^{(t+1)}, .)$.
- These steps yield the following sequence

$$L^{(0)} \rightarrow \mathbf{z}^{(1)} \rightarrow L^{(1)} \rightarrow \mathbf{z}^{(2)} \rightarrow L^{(2)} \rightarrow \ldots \rightarrow \mathbf{z}^{(t)} \rightarrow L^{(t)} \rightarrow \ldots$$

The dynamical clusters method became so classical that often it is referred wrongly by k-means. It allows the user to choose the nature of the cluster centers. Next, we sketch different situations

Examples of dynamic clusters method

- The k-medoids algorithm is a typical dynamical cluster methods minimizing the SSQ criterion, But in contrast to k-means the cluster centers are objects of Ω. The principal advantage of k-medoids is that it overcomes the problem of outliers (Kaufman, 1987). A version PAM (Partition Around Medoids) by Kaufman (1990). Other variants of PAM with less complexity were proposed such as CLARANS by Ng and Han (1994)

- The choice of distance is crucial in clustering. In the process of dynamical clusters we can try to learn a metric. Then instead of to use a fixed distance we can consider that $\Omega \subset \mathbb{R}^p$ and $\mathbb{L} = \mathbb{R}^p \times \Delta$ where Δ is set of distances defined on $\mathbb{R}^p$ and $D(\boldsymbol{x}, (\boldsymbol{\lambda}, d)) = d(\mathbf{x}, \boldsymbol{\lambda})$. Then the method performs clustering and distance metric learning simultaneously. This process allows to take into account the shapes of clusters (Diday, 1974, 1977)

Clustering of categorical data

- The dissimilarity between two vectors of categories can be expressed as

$$D(\boldsymbol{x}_i, \boldsymbol{\lambda}_k) = \sum_{j=1}^{p} \delta(\boldsymbol{x}_{ij}, \boldsymbol{\lambda}_{kj})$$

  where $\delta(\boldsymbol{x}_{ij}, \boldsymbol{\lambda}_{kj}) = 1$ if $\boldsymbol{x}_{ij} = \boldsymbol{\lambda}_{kj}$ and 0 otherwise
- D reflects the number of different categories between the vector $\boldsymbol{x}_i$ and the center $\boldsymbol{\lambda}_k$
- The centers of obtained clusters are summarized by the vectors of categories (Nadif and Marchetti, 1993) or under name k-modes (Huang, 97)

Binary data

- When the data are binary, the distance

$$D(\boldsymbol{x}, \boldsymbol{\lambda}_k) = \sum_{j=1}^{p} |\boldsymbol{x}_{ij} - \boldsymbol{\lambda}_{kj}|$$

  is the Manhattan distance and the vector centers belong to $\{0,1\}^p$

Nominal categorical data matrix and reorganized data matrix

	a	b	c	d	e
1	1	2	2	3	2
2	3	2	1	1	1
3	2	3	3	1	1
4	1	1	2	3	3
5	1	2	1	3	3
6	3	2	1	1	2
7	3	3	2	1	1
8	1	1	1	3	3
9	2	2	2	1	1
10	2	3	3	2	2

	a	b	c	d	e
3	2	3	3	1	1
7	3	3	2	1	1
9	2	2	2	1	1
10	2	3	3	2	2
1	1	2	2	3	2
4	1	1	2	3	3
5	1	2	1	3	3
8	1	1	1	3	3
2	3	2	1	1	1
6	3	2	1	1	2

Centers and Degree of homogeneity

	a	b	c	d	e
A	2	3	2	1	1
B	1	1	1	3	3
C	3	2	1	1	1

	a	b	c	d	e
A	75	75	50	75	75
B	100	50	50	100	75
C	100	100	100	100	50

Binary data matrix and reorganized data matrix

	a	b	c	d	e
1	1	0	1	0	1
2	0	1	0	1	0
3	1	0	0	0	0
4	1	0	1	0	0
5	0	1	0	1	1
6	0	1	0	0	1
7	0	1	0	0	0
8	1	0	1	0	1
9	1	0	0	1	0
10	0	1	0	1	0

	a	b	c	d	e
1	1	0	1	0	1
4	1	0	1	0	0
8	1	0	1	0	1
2	0	1	0	1	0
5	0	1	0	1	1
6	0	1	0	0	1
10	0	1	0	1	0
3	1	0	0	0	0
7	0	1	0	0	0
9	1	0	0	1	0

Centers and Degree of homogeneity

	a	b	c	d	e
A	1	0	1	0	1
B	0	1	0	1	0
C	1	0	0	0	0

	a	b	c	d	e
A	100	100	100	100	67
B	100	100	100	75	50
C	67	67	100	67	100

Simple solution for Ordinal data

- Let a variable with 3 categories $1, 2, 3 \Rightarrow (1, 0, 0), (1, 1, 0), (1, 1, 1)$
- Clustering of binary data

Conclusion

Advantages

- Simple and efficient method
- Give readable results
- Complementary to PCA, CA, MDS etc.
- Extension to contingency tables or categorical data from the principal components
- Fuzzy variants of k-means (see the finite mixture model)
- Methods available in Statistic and data mining Software (See FactoMIner of R)

Disadvantages

- Depend on the shape of clusters
- It requires the number of clusters

Course 2

- Mixture model

Exercise 1

Id	y	x
S1	5	5
S2	6	6
S3	15	14
S4	16	15
S5	25	20
S6	30	19

Without any calculation

- plot $y * x$
- Initialize the k-means algorithm with S1, S4 et S6 and looking for 3 clusters. Remark ?
- Initialize the k-means algorithm with S4, S5 et S6 and looking for 3 clusters. Remark ?