Classification automatique

GHAZI BEL MUFTI

ghazi.belmufti@gmail.com

ESSAI-3 / DATA MINING

Plan I

Plan II

Introduction

- La classification automatique est utilisée pour former des groupes qui sont déterminés selon des critères d'homogénéité et de distance entre les membres : les objets d'une même classe doivent être "similaires" et les objets de deux classes différentes doivent être "distincts".
- Les méthodes de classification non supervisée se diversifient mais tombent essentiellement en deux catégories qui sont :
  - Méthodes à approche géométrique,
  - Méthodes à approche probabiliste.
- C'est un instrument privilègié de l'un des fondements des stratégies du marketing : la segmentation des consommateurs.

Données I

On note :

- $\Omega = \{\omega_1, \ldots, \omega_n\}$ l'ensemble des n objets ou individus à classer.

- X le tableau de données :

$$X = (x_i^j)_{i=1,\ldots,n\ ;\ j=1,\ldots,p}$$

où x_i^j désigne la valeur prise par la j ème variable pour le i ème individu.

- $D = (d_{ii'})$ le tableau $n \times n$ de proximité entre deux objets ω_i et $\omega_{i'}$.

Similarité, dissimilarité et distance I

- Un indice de similarité s de Ω^2 dans $\mathbb{R}^+$ vérifie les deux propriétés suivantes :
  - (1) s est symétrique : $\forall(\omega,\omega') \in \Omega^2,\ s(\omega,\omega') = s(\omega',\omega)$,
  - (2) $\forall(\omega,\omega') \in \Omega^2$, avec $\omega \neq \omega',\ s(\omega,\omega) = s(\omega',\omega') > s(\omega',\omega)$.

  La similarité d'un point à lui-même est une constante supérieure à toutes autre similarité ; on la note s_{max}.
- Un indice de dissimilarité d vérifie la condition (1) et la condition (3) suivante :
  - (3) $\forall\omega \in \Omega,\ d(\omega,\omega) = 0$
- Une distance est un indice de dissimilarité qui vérifie les propriétés suivantes :
  - (4) $d(\omega,\omega') = 0 \Rightarrow \omega = \omega'$
  - (5) $d(\omega,\omega') \leq d(\omega,\omega'') + d(\omega'',\omega')$, pour tout $\omega,\omega',\omega'' \in \Omega$

Similarité, dissimilarité et distance II

Exemples :

- La distance de Minkowski entre deux objets ω_i et $\omega_{i'}$ est donnée par

$$d_q(\omega_i, \omega_{i'}) = \sqrt[1/q]{\sum_{j=1}^{p} |\omega_{i'}^j - \omega_{i'}^j|^q}$$

  cette distance se réduit à la distance euclidienne pour $q = 2$ et city-block pour $q = 1$.
- La distance de Chebychev :

$$d(\omega_i, \omega_{i'}) = \max_{j \in \{1,\ldots,p\}} |\omega_i^j - \omega_{i'}^j|$$

- Le choix d'une distance a une influence sur la forme des classes obtenues : la distance euclidienne contribue à la formation de classes hypersphériques, alors qu'avec la distance de Chebychev, les classes obtenues sont hypercubiques.

Similarité, dissimilarité et distance III

- Distance entre séries temporelles : la distance de Fréchet, Dynamic Time Warping (DTW)...

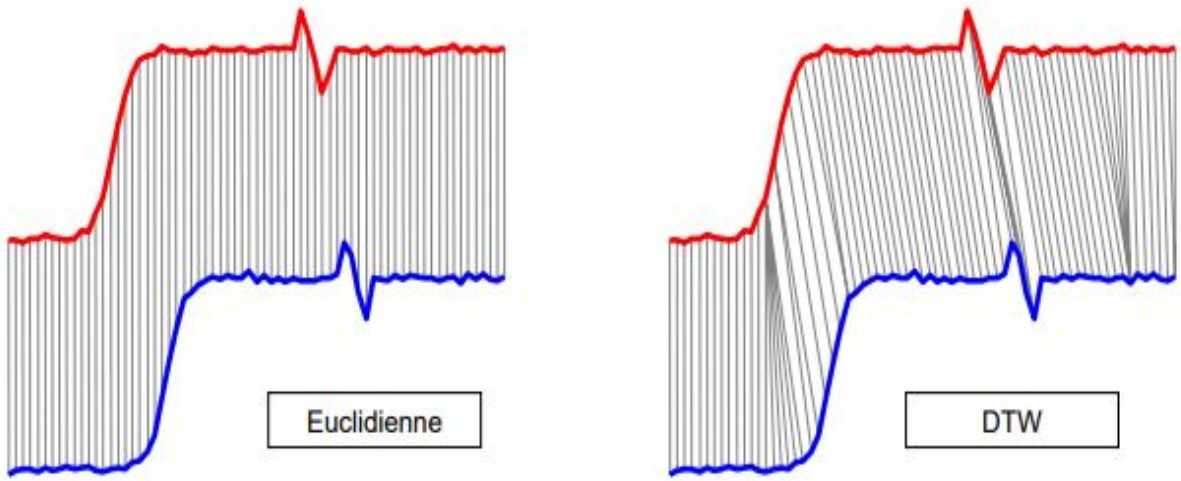


Figure – 1. Dynamic Time Warping

Procédures de constitution des classes

- Les méthodes hiérachiques : produisent des suites de partitions en classes de plus en plus vastes.

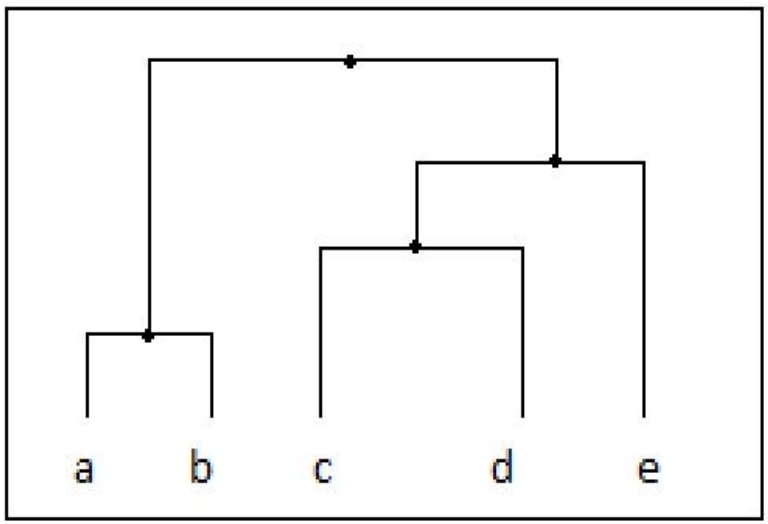


Figure – 2. Représentation hiérarchique

- Les méthodes de partitionnement : produisent directement une partition en un nombre de classes fixé au préalable.

Hiérarchie de parties d'un ensemble Ω

- Une famille H de parties de Ω est une hiérarchie si :
  - a) Ω et les parties à un élément appartiennent à H,
  - b) $\forall A, B \in H \quad A \cap B \in \{A, B, \emptyset\}$. En d'autre termes, deux classes sont soit disjointes soit contenues l'une dans l'autre,
  - c) Toute classe est la réunion des classes qui sont incluses en elle.

- A toute hiérarchie correspond un arbre de classification.
  Exemple : La hiérarchie de la figure 2 est donnée par :

$$H = \{\emptyset, a, b, c, d, e, ab, cd, cde, abcde\}$$

- Une hiérarchie est dite **indicée** s'il existe une application i de H dans $\mathbb{R}^+$ croissante c'est-à-dire telle que si $A \subset B$: $i(A) \leq i(B)$.

Principe des méthodes hiérachiques ascendantes

Il consiste à construire une suite de partitions en n classes, $n-1$ classes, ...emboîtées les une dans les autres de la manière suivante :

- on recherche à chaque étape les deux classes les plus proches, on les fusionne et on continue jusuqu'à ce qu'il n'y ait plus qu'une seule classe.
- la partition en k classes est obtenue en regroupant deux des classes de la partition en $k+1$ classes.
- la partition en n classes est celle où chaque individu est isolé et la partition en une classe n'est autre que la réunion de tous les individus (entre les deux il y'a $n-2$ partitions).

Distances ultramétriques

- A toute hiérarchie indicée H correspond un indice de distance entre éléments de H : $\forall A, B \in H$, $d(A, B)$ est le niveau d'agrégation de A et de B, c'est-à-dire l'indice de la plus petite partie de H contenant à la fois A et B.
- Cette distance possède la propriété suivante, plus forte que l'inégalité triangulaire, dite **ultramétrique** :
  - (6) $d(\omega, \omega') \leq sup\{d(\omega, \omega''); d(\omega', \omega'')\}$, pour tout $\omega, \omega', \omega'' \in \Omega$

Critères de regroupement (ou d'agrégation) de 2 classes. I

- Choisir un critère d'agrégation de deux classes revient à définir une distance entre classes.
- Considérons deux classes C_1 et C_2, la distance entre ces deux classes est souvent déterminée suivant l'un des critères d'agrégation suivants :
  - **Critère du lien minimum :**

$$d(C_1, C_2) = \min_{\omega_i \in C_1,\ \omega_{i'} \in C_2} d(\omega_i, \omega_{i'})$$

Critères de regroupement (ou d'agrégation) de 2 classes. II

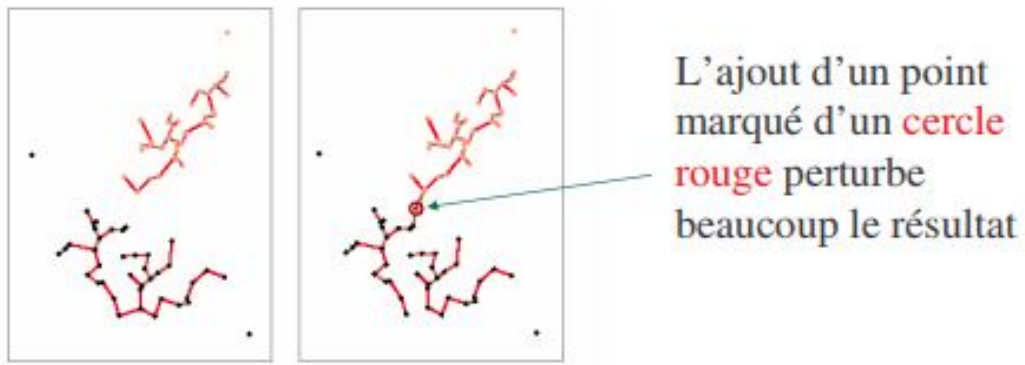


Figure – 3. Critère du lien minimum

- **Critère du lien maximum :**

$$d(C_1, C_2) = \max_{\omega_i \in C_1,\ \omega_{i'} \in C_2} d(\omega_i, \omega_{i'})$$

Critères de regroupement (ou d'agrégation) de 2 classes. III

- **Critère de la moyenne :**

$$d(C_1, C_2) = \frac{1}{n_1 n_2} \sum_{\omega_i \in C_1} \sum_{\omega_{i'} \in C_2} d(\omega_i, \omega_{i'})$$

où n_k désigne l'effectif de la classe k.

- **Critère de Ward :** L'inertie d'un nuage de points peut se décomposer comme suit :

$$\begin{aligned} I_{totale} &= I_{interclasse} + I_{intraclasse} \\ I_{totale} &= \sum_k n_k d^2(g_k, g) + \sum_k \sum_{i \in C_k} d^2(x_i, g_k) \end{aligned}$$

Lorsque l'on passe d'une partition en $k+1$ classes à une partition en k classes en regroupant 2 classes en une seule l'inertie

Critères de regroupement (ou d'agrégation) de 2 classes. IV

interclasse diminue (l'inertie intraclasse augmente). Le critère de regroupement consiste à regrouper les deux classes pour lesquelles la perte d'inertie est la plus faible ; ceci revient à réunir les deux classes les plus proches en prenant comme distance entre deux classes la perte d'inertie inter que l'on encourt en les regroupant :

$$d(C_1, C_2) = \frac{n_1 n_2}{n_1 + n_2} d^2(g_1, g_2)$$

où n_k et g_k sont, respectivement, le cardinal et le centre de gravité de la classe C_k.

En effet :

Soient g_1 et g_2 les centres de gravité des deux classes et g_{12} le centre de gravité de leur réunion.

Critères de regroupement (ou d'agrégation) de 2 classes. V

On peut décomposer l'inertie $I_{1,2}$ de g_1 et g_2 par rapport au centre de gravité du nuage de point g suivant la relation de Huygens :

$$\begin{aligned} I_{1,2} &= n_1 d^2(g_1, g) + n_2 d^2(g_2, g) \\ &= n_1 d^2(g_1, g_{12}) + n_2 d^2(g_2, g_{12}) + (n_1 + n_2) d^2(g_{12}, g) \end{aligned}$$

Seul le dernier terme subsiste si g_1 et g_2 sont remplacés par leur centre de gravité.

La perte d'inertie inter-classes ΔI_{12} due au passage de la partition à k classes à la partition à $k-1$ classes équivaut à :

$$\Delta I_{12} = n_1 d^2(g_1, g_{12}) + n_2 d^2(g_2, g_{12})$$

En remplaçant g_{12} par sa valeur (i.e. $\frac{n_1 g_1 + n_2 g_2}{n_1 + n_2}$), il vient :

$$\Delta I_{12} = \frac{n_1 n_2}{n_1 + n_2} d^2(g_1, g_2)$$

Critères de regroupement (ou d'agrégation) de 2 classes. VI

Remarque : Sachant que la somme des niveaux d'agrégation des différents noeuds de l'arbre est égal à l'inertie totale, cette méthode est considérée comme complémentaire de l'ACP et repose sur un critère d'optimisation assez naturel.

Formule de Lance et Williams

D'une manière générale, le calcul de l'indice d'agrégation entre la nouvelle classe (union de deux classes) et les autres classes de la partition peut se faire par la formule de Lance et Williams suivante :

$$D(C_l, C_i \cup C_j) = \alpha_i D(C_l, C_i) + \alpha_j D(C_l, C_j) + \beta D(C_i, C_j) + \gamma |D(C_l, C_i) - D(C_l, C_j)|$$

Algorithme de classification	α_i	α_j	β	γ
Single Linkage	$1/2$	$1/2$	0	$-1/2$
Complete Linkage	$1/2$	$1/2$	0	$1/2$
Centroïd Linkage	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$-\frac{n_i n_j}{n_i+n_j}$	0
Méthode de Ward	$\frac{n_i+n_l}{n_i+n_j+n_l}$	$\frac{n_j+n_l}{n_i+n_j+n_l}$	$-\frac{n_l}{n_i+n_j+n_l}$	0

3. Partition issue d'une hiérarchie

- Pour déterminer une partition à partir d'une hiérarchie, il suffit de couper la hiérarchie à un niveau donné et d'identifier les branches (classes) qui en découlent.

- Pour déterminer la meilleure partition issue de la hiérachie et donc le meilleure nombre de classes, il faut
  a. Critère du plus haut saut :
     - Identifier le plus haut saut entre deux paliers successifs
     - Couper la hiéarchie entre ces deux paliers : la partition obtenue est celle ayant le meilleur nombre de classes.
  b. Critères basés sur la mesure de l'adéquation de la structure en classes aux données initiales
  c. Critères basés sur la stabilité de la structure obtenue (i.e. des classes)

Les méthodes non hiérachiques

Principe

- Les méthodes non hiérarchiques permettent de traiter des populations importantes (même 1000 et plus) à des coûts raisonnables. Ces méthodes visent à constituer directement k types à partir de n objets en essayant d'optimiser un indice global mesurant la qualité de la classification.
- Le choix du nombre de groupes se pose ici *ex-ante* ; on peut parfois avoir des hypothèses *a priori* provenant d'une phase exploratoire qualitative.

  Exemples. centres mobiles, k-means

Algorithme des centres mobiles I

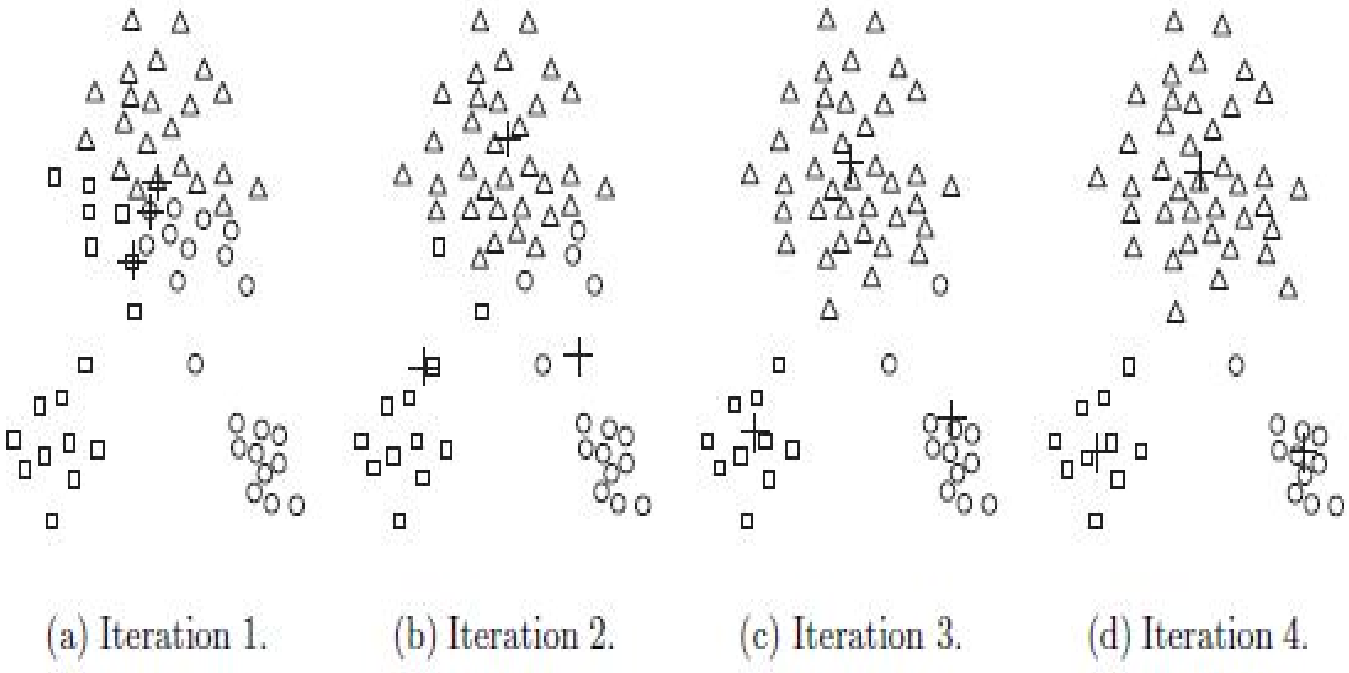


Figure – 4. Etapes de l'algorithme des centres mobiles

Algorithme des centres mobiles II

- Etape 0 :
  - on tire au hasard K centres arbitraires $c_1^{(0)}, c_2^{(0)}, ..., c_K^{(0)}$;
  - on regroupe les individus autour de ces K centres : la classe associée au centre $c_k^{(0)}$, $k = 1, ..., K$, est constituée de l'ensemble des individus les plus proches de $c_k^{(0)}$ que tout autre centre ; on obtient la partition : $\{C_1^{(0)}, ..., C_K^{(0)}\}$.
- Etape 1 :
  - on calcule les centres de gravité $g_1^{(1)}, g_2^{(1)}, ..., g_K^{(1)}$ des classes que l'on vient de former ;
  - on effectue une deuxième partition en regroupant les individus autour de ces centres $g_k^{(1)}$, $k = 1, ..., K$, on obtient la partition : $\{C_1^{(1)}, ..., C_K^{(1)}\}$

Algorithme des centres mobiles III

- Etape m :
  - on calcule K nouveaux centres de gravité $g_1^{(m)}$, $g_2^{(m)}$,..., $g_K^{(m)}$,
  - on regroupe les individus autour d'eux pour obtenir la partition formée des classes $\{C_1^{(m)}, ..., C_K^{(m)}\}$.

L'algorithme continue ainsi jusqu'à ce que la qualité de la partition mesurée par un critère convenablement choisi (i.e. l'inertie intra-classes) ne s'améliore plus.

Algorithme des centres mobiles IV

Méthode des $k-$means : A la différence avec la méthode des centres mobiles, cette méthode n'attend pas d'avoir procédé à la réaffectation de tous les individus pour modifier la postion des centres : chaque réaffectation d'individus entraîne une modification de la position du centre correspondant.

Remarque. L'inconvénient de ces méthodes est de fournir une partition finale qui dépend de la partition de départ : on n'atteint pas l'optimum global mais seulement la meilleure partition possible à partir de celle de départ, d'où la nécessité d'effectuer plusieurs **itérations** pour atteindre une partition de bonne qualité.

Algorithme des centres mobiles V

Justification de l'algorithme

La variance intra-classes ne peut que décroître (ou rester stationnaire) entre l'étape m et l'étape $m+1$. Des règles d'affectations permettent de faire en sorte que cette décroissance soit stricte et donc de conclure à la convergence de l'algorithme.

En effet :

- Supposons que les n individus de l'ensemble à classer soient munis de masses realtives p_i dont la somme vaut 1 et soit $d^2(i, g_k^{(m)})$ le carré de la distance entre l'individu i et le centre de la classe k à l'étape m. Nous nous intéressons à la quantité critère :

$$\nu(m) = \sum_{k=1}^{K} \sum_{i \in C_k^{(m)}} p_i d^2(i, g_k^{(m)})$$

Algorithme des centres mobiles VI

- Rappelons qu'à l'étape m, la classe $C_k^{(m)}$ est formée des individus plus proches de $g_k^{(m)}$ que de tous les autres centres (ces centres étant des centres de gravité des classes $C_k^{(m-1)}$ de l'étape précédente).
  La variance intra-classes à l'étape m est la quantité :

$$V(m) = \sum_{k=1}^{K} \left\{ \sum_{i \in C_k^{(m)}}^{n} p_i d^2(i, g_k^{(m+1)}) \right\}$$

où $g_k^{(m+1)}$ est le centre de gravité de la classe $C_k^{(m)}$.

Algorithme des centres mobiles VII

- A l'étape $m+1$, la quantité s'écrit :

$$\nu(m+1) = \sum_{k=1}^{K} \left\{ \sum_{i \in C_k^{(m+1)}}^{n} p_i d^2(i, g_k^{(m+1)}) \right\}$$

On vérifie alors que :

$$\nu(m) \geq V(m) \geq \nu(m+1)$$

ce qui établira la décroissance simultanée du critère et de la variance intra-classes : en notant p_k la somme des p_i pour $i \in C_k^{(m)}$, remarquons tout d'abord d'après le théorème de Huygens que

$$\nu(m) = V(m) + \sum_{k=1}^{K} p_k d^2(g_k^{(m)}, g_k^{(m+1)})$$

Algorithme des centres mobiles VIII

ce qui établit la première partie de l'inégalité.
La seconde partie découle du fait qu'entre les accolades qui apparaissent dans les définitions de $V(m)$ et $\nu(m+1)$, seules changent les affectations des points aux centres.
Puisque $C_k^{(m+1)}$ est l'ensemble des points plus proches de $g_k^{(m+1)}$ que de tous les autres centres, les distances n'ont pu que décroître (ou rester inchangées) au cours de cette réaffectation.

La classification mixte

1. Partition préliminaire : centre mobiles...
2. Classification ascendantes hiérarchique sur les centres

3.a Partition finale en k classes par coupure de l'arbre

3.b "Consolidation" par réaffectation : pour améliorer la partition obtenue, on utilise de nouveau une procédure d'agrégation autour des centres mobiles :

- Au départ, les centres sont ceux obtenus par coupure de l'arbre
- A la première itération, on affecte les individus à leur centre le plus proche
- Ceci crée de nouvelles classes dont on calcule les centres, puis on réaffecte...

Interprétation des classes d'une I

Cette étape consiste à trouver la signification pratique des classes. Pour cela il faut revenir au variables initiales décrivant les individus :

- V. quantitatives. on décrira les classes en se basant sur les coordonnées de leur centre de gravité. Ces coordonnées ne sont autres que les moyennes des variables pour les individus constituant une classe donnée
- V. qualitatives. l'interprétation peut être assez complexe dans la mesure où les classes ne sont pas toujours "pures" mais représentant des "tendances dominantes" au sein de la population étudiée
  Exemple : En croisant la variable sexe avec la variable classe d'appartenance d'un consommateur, on aura des types plutôt masculin formés de 70% d'hommes et 30% de femmes, d'autres plutôt féminin.

Interprétation des classes d'une partition II

- La fonction catdes du `package FactoMineR` permet de trier les varibles quantitatives
  - de la plus caractéristique à la moins caractéristique en positif (i.e. variables pour lesquelles les individus prennent des valeurs significativement sup. à la moyenne de l'ensemble des individus)
  - de la moins caratéristique à la plus caractéristique en négatif
- Pour les variables qualitatives, ce sont les modalités des variables qui sont triées selon le même principe que celui des variables quantitatives.
- Une valeur test supérieure à 2 en valeur absolue signifie que la moyenne de la classe est sig. différente de la moyenne générale
- Un signe positif (resp. négatif) de la valeur-test indique que la moyenne de la classe est sup. (resp. inf.) à la moyenne

Classification de données qualitatives I

Lorsque les n individus à calsser sont décrits par des variables qualitatives, divers cas se présentent :

- Pour les données du type présence-absence de nombreux indices de similarité existent et qui combinent les quatres nombres suivants associés à un couple d'individus i et i' :

  a = nombre de caractéristiques communes ;
  b = nombre de caractéristiques possédées par i et pas par i' ;
  c = nombre de caractéristiques possédées par i' et pas par i ;
  d = nombre de caractéristiques que ne possèdent ni i ni i'.

  Les indices suivants compris entre 0 et 1 sont alors aisément transformables en dissimilarité par complémentarité à 1 :

  - Jaccard = $\dfrac{a}{a+b+c}$
  - Russel et Rao = $\dfrac{a}{a+b+c+d}$

Classification de données qualitatives II

- Pour les données du type *m* variables qualitatives :

  Approche 1 : utiliser

  1. la représentation disjonctive complète des données,
  2. la distance du χ^2 entre les lignes du tableau :

$$d^2(i,i') = \sum_{j=1}^{p} \frac{nm}{n_{.j}} \left(\frac{x_i^j - x_{i'}^j}{m}\right)^2$$

     Cette similarité possède des propriétés intéressantes puisqu'elle dépend non seulement du nombre de modalités possédées en commun par i et i' mais de leur fréquence (deux individus ayant en commun une modalité rare sont plus proches que deux individus ayant en commun une modalité fréquente).
  3. la méthode de Ward (puisque la distance de χ^2 est euclidienne) sur le tableau de distance.

Classification de données qualitatives III

Approche 2 : Effectuer une classification hiérarchique sur le tableau des coordonnées des n individus après une ACM du tableau de données.

Approche 3 : Algorithme des $k-$modes [Huang 1997]

`R : klaR package`

C'est une adaptation de l'algorithme des k-means en considérant les modes des classes plutôt que les centres de gravité, une dissimilarité adaptée aux données qualitatives et une fonction qui met à jour les modes des classes à chaque itération.

1. parmi les dissimilarités utilisées, le "simple matching" ($d(x_i, x_{i'}) = \sum_j \delta(x_i^j, x_{i'}^j)$) ou celle du χ^2 ;
2. le mode $Q = (q_1, \ldots, q_m)$ d'un ensemble Ω est le vecteur Q de Ω minimisant $D(Q, \Omega) = \sum_{i=1}^{n} d(x_i, Q)$.

Classification sur variables mixtes I

Soit Ω l'ensemble de données dont les objets sont décrits par p variables dont p_1 quantitatives et p_2 qualitatives (i.e. catégorielles) :

$$X = (X^{N,1}, \ldots, X^{N,p_1}, X^{C,p_1+1}, \ldots, X^{C,p_1+p_2})$$

On retient alors les deux approches de classification suivantes :

Approche 1 :

1. Séparer les données en deux de manière à avoir une partie quantitative et une autre qualitative ;
2. L'ensemble des données est classifié en utilisant les variables quantitatives uniquement ce qui permet d'avoir la variable classe de chaque objet. Cette nouvelle variable "classe" est rajoutée aux p_2 variables qualitatives pour constituer une nouvelle matrice de données ;
3. Le nouveau jeu qualitatif est classifié en utilisant un algorithme de classification sur données qualitatives

Classification sur variables mixtes II

Approche 2 : Algorithme des *k*-prototypes [Huang 1997]

`R : clustMixType package`

C'est une adaptation de l'algorithme des *k*-means en considérant des "prototypes" (i.e. objets de type mixte) pour représenter les classes plutôt que les centres de gravité, une dissimilarité adaptée aux données mixte :

$$d(x_i, x_{i'}) = \sum_{j=1}^{p_1} (x_i^j - x_{i'}^j)^2 + \gamma \sum_{j=p_1+1}^{p=p_1+p_2} \delta(x_i^j, x_{i'}^j)$$

La valeur de γ est choisie de sorte à ne pas favoriser l'un ou l'autre des types de variables (cf. [Huang 1997] et [Huang 1998] pour une discussion sur le choix de la valeur de γ).

Huang, Z. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Variables, Data Mining and Knowledge Discovery 2, 283-304, 1998.

Huang, Z. A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining. in KDD : Techniques and Applications,

21-34, 1997.

Mesure de la validité d'une classification

Deux types de mesures :

- Indices de validité interne : à partir d'un résultat de classification et de l'information intrinsèque dans la base de données on mesure la qualité du résultat ;
- Indice de validité externe : comparer la classification obtenue avec une autre classification (soit une partition déjà connue ou une autre classification obtenue par une autre méthode).

Deux approches de validation interne d'un partionnement

1. *Adéquation de la structure en classes avec l'ensemble des données*

   La distance entre deux objets *classés ensemble* est inférieure à la distance entre deux objets *classés dans des classes différentes* : indice Silhouette, indice de Calinski and Harabasz, [Milligan et Cooper 1985]...

2. *Stabilité des résulats d'une méthode de partionnement*

   Des petits changements subis par l'ensemble de données n'ont pas d'effet significatif sur l'appartenance des individus aux classes : indice de Rand ou Rand corrigé [Hubert et Arabie 1985]...

Versions perturbées de l'ensemble des données

- Version perturbée de l'ensemble des données obtenue par échantillonnage aléatoire : Méthodes de rééchantillonnage, Validation croisée ;
- Version perturbée de l'ensemble des données obtenue par ajout de bruit.

Indices basé sur l'adéquation aux données I

Indice Silhouette [Rousseeuw 1987]

On définit la Silhouette d'un objet i par

$$S(i) = \frac{b_i - a_i}{max(b_i, a_i)}$$

où

- a_i est la distance moyenne entre x_i et tous les autres objets de la classe $C(i)$ (i.e. la classe à laquelle appartient l'objet x_i) :

$$a_i = \frac{1}{|C(i)|} \sum_{x \in C(i)} d(x_i, x).$$

- b_i est la distance moyenne entre x_i et les objets de la classe la plus proche de x_i :

$$b_i = \min_{C \neq C(i)} \frac{1}{|C|} \sum_{x \in C} d(x_i, x).$$

Indices basé sur l'adéquation aux données II

- **L'indice Silhouette** de validation d'une partition (i.e. du bon nombre de classes d'une partition) est la moyenne des silhouettes de tous les objets.
- La meilleure partition est celle dont le nombre de classes correspond à

$$\arg\max_{k} (\mathit{Silhouette}),\ k \in \{2, \ldots, K\}.$$

Indices basé sur l'adéquation aux données III

Indice de Calinski and Harabasz [1974]

- Pour un nombre k, $k \in \{2, \ldots, K\}$ de classes donné, cet indice est défini par

$$CH_k = \frac{B_k}{k-1} / \frac{W_k}{n-k}$$

.

- La meilleure partition est celle dont le nombre de classes correspond à

$$\arg\max_k CH_k, \ k \in \{2, \ldots, K\}.$$

Comparaison de deux partitions basée sur l'indice de Rand I

Considérons deux partitions P et Q de X.

Soit N_{11} (resp. N_{00}) le nombre de paires d'objets classés ensemble (resp. séparément) à la fois par P et par Q.

L'indice de Rand R est alors défini par :

$$\mathrm{R}(P,Q) = \frac{N_{11}+N_{00}}{\binom{n}{2}},$$

où $n = |X|$.

Comparaison de deux partitions basée sur l'indice de Rand II

A problem with the Rand index is that the expected value of the Rand index of two random partitions does not take a constant value (say zero). The adjusted Rand index proposed by [Hubert and Arabie, 1985] assumes the generalized hypergeometric distribution as the model of randomness, *i.e.*, the U and V partitions are picked at random such that the number of objects in the classes and clusters are fixed. Let n_{ij} be the number of objects that are in both class u_i and cluster v_j. Let $n_{i.}$ and $n_{.j}$ be the number of objects in class u_i and cluster v_j respectively. The notations are illustrated in Table 1.

Class \ Cluster	v_1	v_2	...	v_C	*Sums*
u_1	n_{11}	n_{12}	...	n_{1C}	$n_{1.}$
u_2	n_{21}	n_{22}	...	n_{2C}	$n_{2.}$
⋮	⋮	⋮		⋮	⋮
u_R	n_{R1}	n_{R2}	...	n_{RC}	$n_{R.}$
Sums	$n_{.1}$	$n_{.2}$	...	$n_{.C}$	$n_{..} = n$

Table 1: Notation for the contingency table for comparing two partitions.

The general form of an index with a constant expected value is $\frac{index - expected\ index}{maximum\ index - expected\ index}$, which is bounded above by 1, and takes the value 0 when the index equals its expected value.

Under the generalized hypergeometric model, it can be shown [Hubert and Arabie, 1985] that:

Comparaison de deux partitions basée sur l'indice de Rand III

$$E\left[\sum_{i,j}\binom{n_{ij}}{2}\right] = \left[\sum_{i}\binom{n_{i.}}{2}\sum_{j}\binom{n_{.j}}{2}\right] / \binom{n}{2} \tag{1}$$

The expression $a + d$ can be simplified to a linear transformation of $\sum_{i,j}\binom{n_{ij}}{2}$. With simple algebra, the adjusted Rand index [Hubert and Arabie, 1985] can be simplified to:

$$\frac{\sum_{i,j}\binom{n_{ij}}{2} - \left[\sum_{i}\binom{n_{i.}}{2}\sum_{j}\binom{n_{.j}}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_{i}\binom{n_{i.}}{2} + \sum_{j}\binom{n_{.j}}{2}\right] - \left[\sum_{i}\binom{n_{i.}}{2}\sum_{j}\binom{n_{.j}}{2}\right] / \binom{n}{2}} \tag{2}$$

Mesure de la stabilité par l'indice de Rand

1. Les données sont partitionnées pour obtenir une partition de référence P_k en k classes ;
2. Les données sont perturbées soit par rééchantillonnage soit par rajout de bruit au coordonnées des individus ;
3. Les données perturbées sont partionnées avec la méthode de classification utilisée dans 1. et avec le même nombre de classes pour obtenir une nouvlle partition Q_k ;
4. La partition Q_k est comparée à la partition P_k à l'aide de l'indice de Rand (ou Rand corrigé...). Plus cet indice est proche de 1 plus la partition est stable.

La meilleure partition est celle correspondant au nombre de classes qui maximise l'indice de Rand.

Cartes auto-organisatrices pour la classification (SOM) I

- Elles sont souvent désignées par le terme anglais self organizing maps (SOM), ou encore cartes de Kohonen du nom du statisticien ayant développé le concept en 1984.
- Les SOM forment une classe de réseau de neurones artificiels fondée sur des méthodes d'apprentissage non-supervisées.
- Un réseau de neurones artificiels, ou réseau neuronal artificiel, est un ensemble d'algorithmes dont la conception est à l'origine très schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques.

Cartes auto-organisatrices pour la classification (SOM) II

- Elles sont utilisées pour cartographier un espace réel, c'est-à-dire pour étudier la répartition de données dans un espace à grande dimension. En pratique, cette cartographie peut servir à réaliser des tâches de discrétisation, quantification vectorielle ou classification.
- La carte réalise une "discrétisation de l'espace", c'est-à-dire qu'elle le divise en zones, et affecte à chaque zone un point significatif dit "vecteur référent".
- Les SOM sont souvent utilisées pour la classification non supervisée de données :
  - chaque neurone représente une classe,
  - chaque observation est affectée au neurone dont le vecteur référent est le plus proche.
  - deux neurones voisins sur la carte représenteront des observations qui sont proches dans l'espace d'entrée.

Architecture I

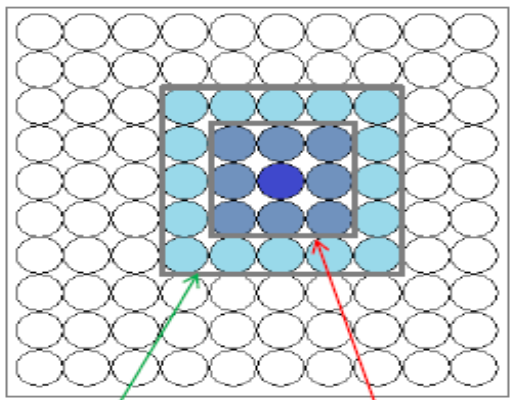
- Une carte auto-organisatrice est composée d'une grille de neurones de faible dimension.
  - Quand la grille est unidimensionnelle, chaque neurone a deux voisins.
  - Quand la grille est bidimensionnelle, l'arrangement des neurones se fait d'une façon rectangulaire où chaque neurone possède 4 voisins (topologie rectangulaire) ou d'une façon hexagonale où chaque neurone possède 6 voisins (topologie hexagonale).

Architecture II

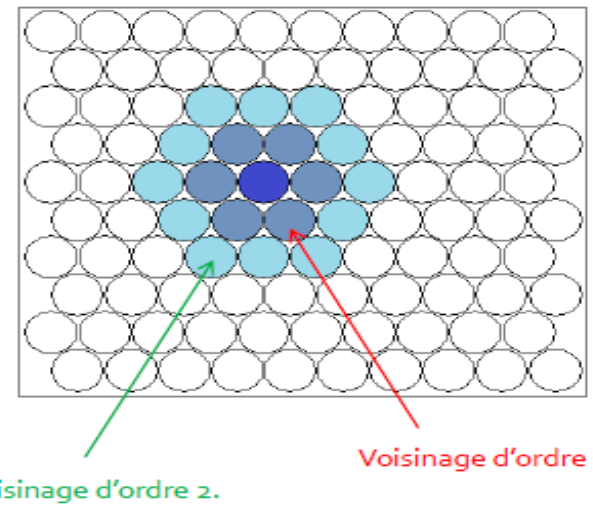
SOM
Architecture et notion de voisinage

La notion de voisinage est primordiale dans SOM, notamment pour la mise à jour des poids et leurs propagations durant le processus d'apprentissage.

Carte rectangulaire – Voisinage rectangulaire

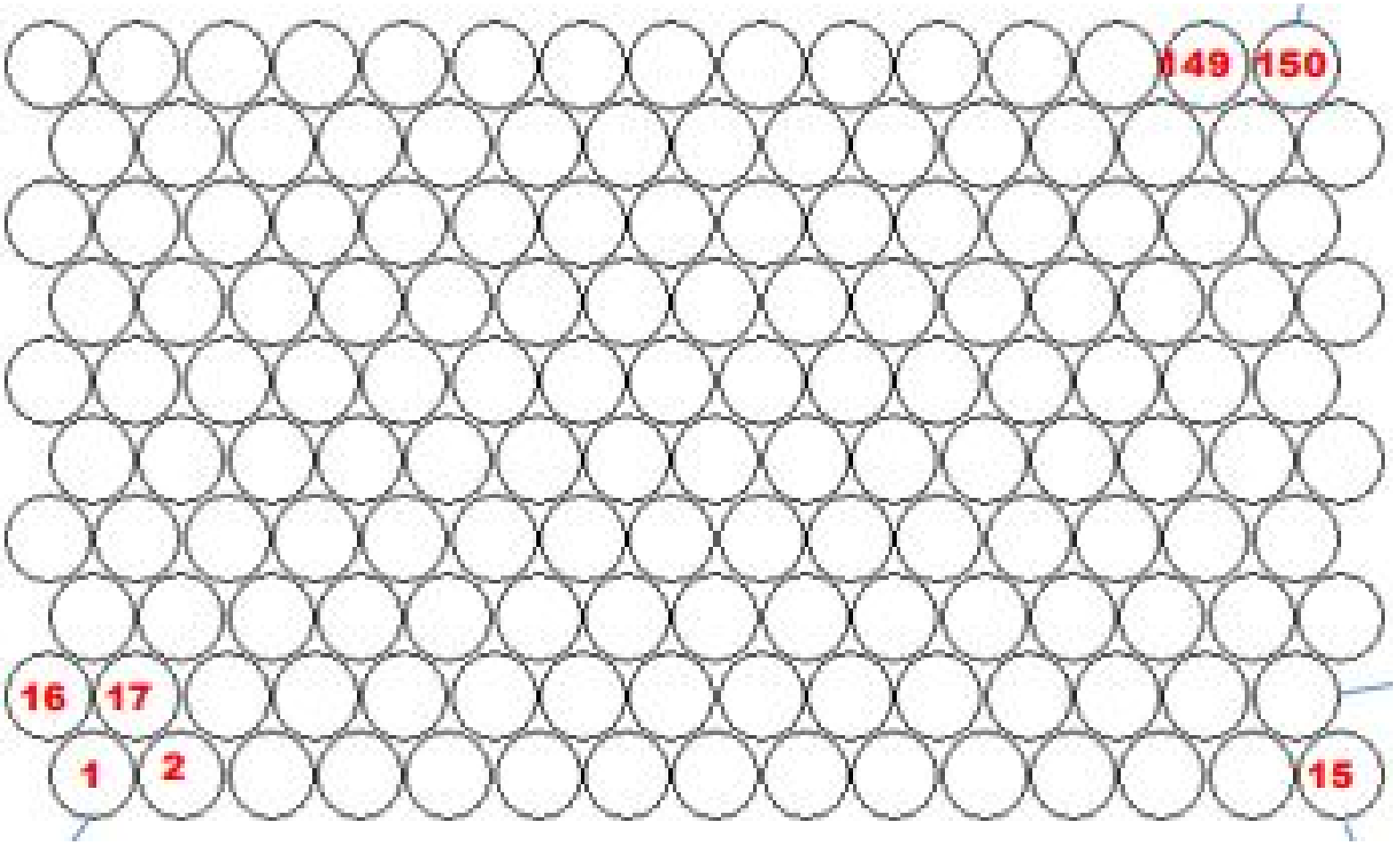


Carte hexagonale – Voisinage circulaire



Remarque : une carte unidimensionnelle (vecteur) est possible

Architecture III



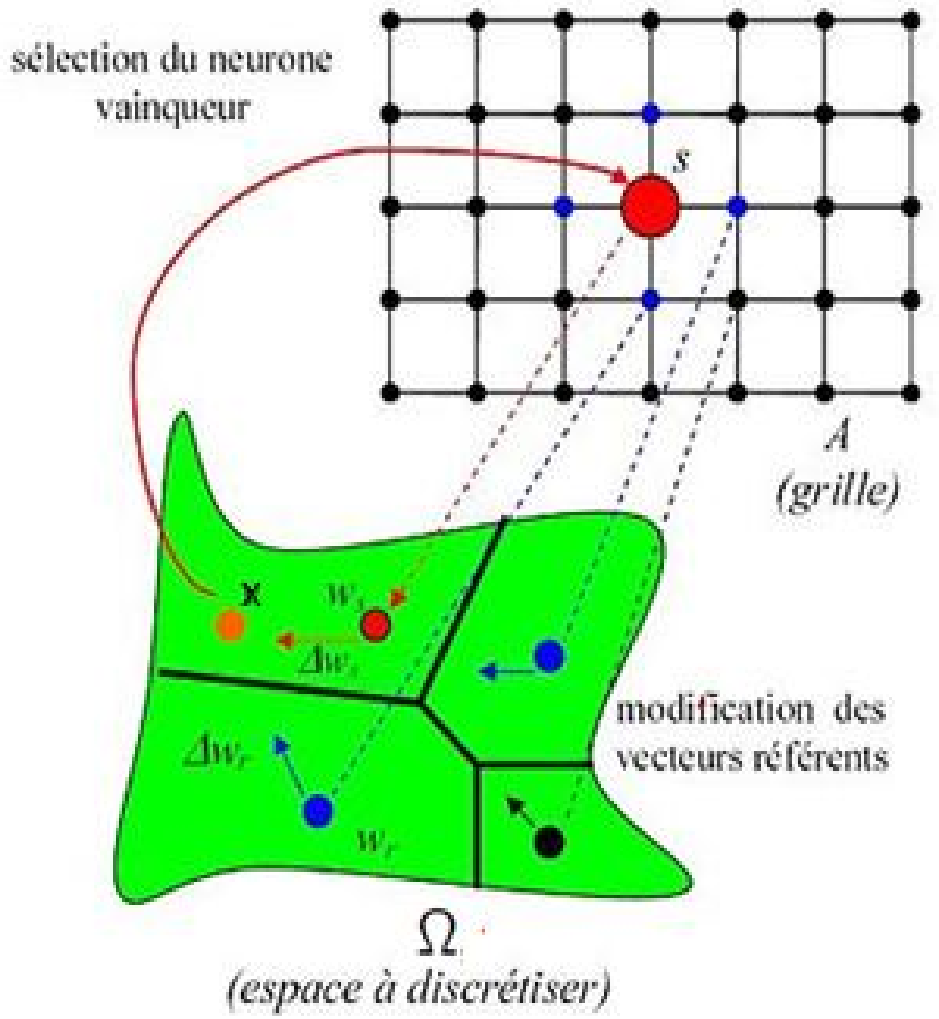
Architecture IV

- Les neurones sont reconnus par leur numéro et leur emplacement sur la grille.
- Par exemple, la figure précédente représente une carte rectangulaire à 150 noeuds.

Principe I

- Les données sont projetées de leur espace initial, ou espace d'entrée, vers la carte ou espace de sortie.
- A chaque neurone de la carte est associé un vecteur référent, appelé aussi vecteur prototype ou prototype, appartenant à l'espace d'entrée.
- En désignant par K le nombre total des neurones de la carte, le vecteur référent du neurone k est reconnu par w_k avec $k \in \{1, \ldots, K\}$ et $w_k \in \mathbb{R}^p$.
- L'objectif de l'apprentissage de la carte consiste à mettre à jour les vecteurs référents de façon à approximer au mieux la distribution des vecteurs d'entrée tout en reproduisant l'auto-organisation des neurones de la carte.

Principe II


sélection du neurone
vainqueur
s
A
(grille)
x
w_s
Δw_s
Δw_r
w_r
modification des
vecteurs référents
Ω
(espace à discrétiser)

Algorithme I

Chaque itération t de l'apprentissage comprend deux étapes :

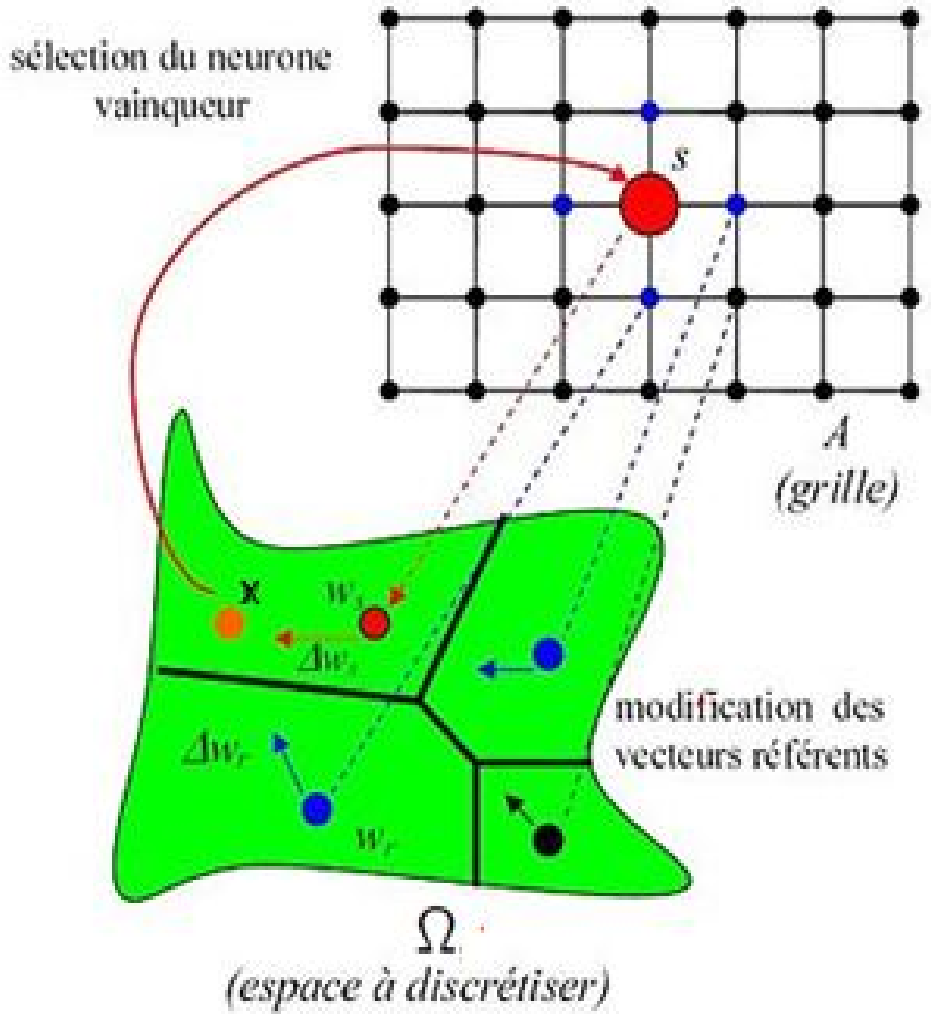
- Etape 1 :
  - Elle consiste à choisir au hasard une observation $x(t)$ de l'ensemble Ω, et à la présenter au réseau dans le but de déterminer son neurone vainqueur ;
  - Le neurone vainqueur (Best Matching Unit), d'une observation est le neurone dont le vecteur référent en est le plus proche au sens d'une distance donnée (ex : distance euclidienne) ;
  - Si s est le neurone vainqueur du vecteur $x(t)$, s est déterminé comme suit :

$$d(w_s(t), x(t)) = \min_{k \in \{1,\ldots,K\}} d(w_k(t), x(t))$$

Algorithme II

- Etape 2 :
  - Le neurone vainqueur est activé ;
  - Son vecteur référent est mis à jour pour se rapprocher du vecteur d'entrée présenté au réseau ;
  - Cette mise à jour ne concerne pas seulement le neurone vainqueur mais aussi les neurones qui lui sont voisins et qui voient alors leurs vecteurs référents s'ajuster vers ce vecteur d'entrée (cf. figure suivante) :

Algorithme III



Ajustement I

- L'amplitude de cet ajustement est déterminée par la valeur d'un pas d'apprentissage $\alpha(t)$ et la valeur d'une fonction de voisinage $h(t)$.
- Le paramètre $\alpha(t)$ règle la vitesse de l'apprentissage. Il est initialisé avec une grande valeur au début puis décroît avec les itérations en vue de ralentir au fur et à mesure le processus d'apprentissage.
- La fonction $h(t)$ définit l'appartenance au voisinage. Elle dépend à la fois de l'emplacement des neurones sur la carte et d'un certain rayon de voisinage.
  - Dans les premières itérations, le rayon de voisinage est assez large pour mettre à jour un grand nombre de neurones voisins du neurone vainqueur,
  - Ce rayon se rétrécit progressivement pour ne contenir que le neurone vainqueur avec ses voisins immédiats, ou bien même le neurone vainqueur seulement.

Ajustement II

La règle de mise à jour des vecteurs référents est la suivante :

$$w_k(t+1) = w_k(t) + \alpha(t) h_{sk}(t)[x(t) - w_k(t)], k \in \{1, \ldots, K\}$$

où s est le neurone vainqueur du vecteur d'entrée $x(t)$ présenté au réseau à l'itération t et $h_{sk}(t)$ est la fonction de voisinage qui définit la proximité entre les neurones s et k.

- Une fonction de voisinage entre le neurone vainqueur s et un neurone k de la carte vaut 1 si le neurone k se trouve à l'intérieur du carré centré sur le neurone s et 0 dans les autres cas.
- Le rayon de ce carré est appelé rayon de voisinage. Il est large au début, puis se rétrécit avec les itérations pour contenir seulement le neurone s avec ses voisins immédiats à la fin de l'apprentissage ou même seulement le neurone s.

Ajustement III

- Une fonction de voisinage plus flexible et plus commune est la fonction gaussienne suivante :

$$h_{sk}(t) = exp(-\frac{||r_s - r_k||^2}{2\sigma^2(t)})$$

où r_s et r_k sont respectivement l'emplacement du neurone s et du neurone k sur la carte, et $\sigma(t)$ est le rayon du voisinage à l'itération t du processus d'apprentissage.

- La fonction de voisinage h force les neurones qui se trouvent dans le voisinage de s à rapprocher leurs vecteurs référents du vecteur d'entrée $x(t)$. Moins un neurone est proche du vainqueur dans la grille, moins son déplacement est important.

L'algorithme EM I

- L'algorithme EM (pour Expectation-Maximisation) est un algorithme itératif du à Dempster, Laird et Rubin (1977). Il s'agit d'une méthode d'estimation paramétrique s'inscrivant dans le cadre général du maximum de vraisemblance.
- Lorsque les seules données dont on dispose ne permettent pas l'estimation des paramètres, et/ou que l'expression de la vraisemblance est analytiquement impossible à maximiser, l'algorithme EM peut être une solution.
- De manière grossière et vague, il vise à fournir un estimateur lorsque cette impossibilité provient de la présence de données cachées ou manquantes ou plutôt, lorsque la connaissance de ces données rendrait possible l'estimation des paramètres.
- L'algorithme EM tire son nom du fait qu'à chaque itération il opère deux étapes distinctes :

L'algorithme EM II

- la phase Expectation, souvent désignée comme "l'étape E", procède comme son nom le laisse supposer à l'estimation des données inconnues, sachant les données observées et la valeur des paramètres déterminée à l'itération précédente ;
- la phase Maximisation, ou "étape M", procède donc à la maximisation de la vraisemblance, rendue désormais possible en utilisant l'estimation des données inconnues effectuée à l'étape précédente, et met à jour la valeur du ou des paramètre(s) pour la prochaine itération.

Principe de fonctionnement I

- En considérant un échantillon $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ d'individus suivant une loi $f(\boldsymbol{x}_i, \theta)$ paramétrée par $\boldsymbol{\theta}$, on cherche à déterminer le paramètre $\boldsymbol{\theta}$ maximisant la log-vraisemblance donnée par

$$L(\mathbf{x}; \boldsymbol{\theta}) = \sum_{i=1}^{n} \log f(\boldsymbol{x}_i, \boldsymbol{\theta}).$$

- Cet algorithme est particulièrement utile lorsque la maximisation de L est très complexe mais que, sous réserve de connaître certaines données judicieusement choisies, on peut très simplement déterminer $\boldsymbol{\theta}$.

Principe de fonctionnement II

- Dans ce cas, on s'appuie sur des données complétées par un vecteur $\mathbf{z} = (z_1, \ldots, z_n)$ inconnu. En notant $f(z_i|\boldsymbol{x}_i; \theta)$ la probabilité de z_i sachant $\boldsymbol{x}_i$ et le paramètre $\boldsymbol{\theta}$, on peut définir la log-vraisemblance complétée comme la quantité

$$L((\mathbf{x}, \mathbf{z}); \boldsymbol{\theta}) = \sum_{i=1}^{n} \left(\log f(z_i|\boldsymbol{x}_i, \boldsymbol{\theta}) + \log f(\boldsymbol{x}_i; \boldsymbol{\theta})\right),$$

et donc,

$$L(\mathbf{x}; \boldsymbol{\theta}) = L((\mathbf{x}, \mathbf{z}); \boldsymbol{\theta}) - \sum_{i=1}^{n} \log f(z_i|\boldsymbol{x}_i, \boldsymbol{\theta}).$$

Principe de fonctionnement III

- L'algorithme EM est une procédure itérative basée sur l'espérance des données complétées conditionnellement au paramètre courant. En notant $\theta^{(m)}$ ce paramètre, on peut écrire

$$E\left[L(\mathbf{x};\theta)|\theta^{(m)}\right] = E\left[L\left((\mathbf{x},\mathbf{z});\theta\right)|\theta^{(m)}\right] - E\left[\sum_{i=1}^{n}\log f(z_i|\boldsymbol{x}_i,\theta)|\theta^{(m)}\right],$$

où l'espérance est prise sur $\mathbf{z}$.

On a alors

$$L(\mathbf{x};\theta) = Q\left(\theta;\theta^{(m)}\right) - H\left(\theta;\theta^{(m)}\right),$$

car $L(\mathbf{x};\theta)$ ne dépend pas de $\mathbf{z}$, avec

$$Q\left(\theta;\theta^{(m)}\right) = E\left[L\left((\mathbf{x},\mathbf{z});\theta\right)|\theta^{(m)}\right]$$

Principe de fonctionnement IV

et

$$H\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(m)}\right) = E\left[\sum_{i=1}^{n} \log f(z_i|\boldsymbol{x}_i,\boldsymbol{\theta})|\boldsymbol{\theta}^{(m)}\right].$$

On montre que la suite définie par

$$\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}}\left(Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(m)}\right)\right)$$

fait tendre $L\left(\mathbf{x};\boldsymbol{\theta}^{(m+1)}\right)$ vers un maximum local.

L'algorithme EM

L'algorithme EM peut être défini par :

- Initialisation au hasard de $\boldsymbol{\theta}^{(0)}$
- m=0
- Tant que l'algorithme n'a pas convergé, faire
  - Evaluation de l'espérance (étape E) :

$$Q\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(m)}\right) = E\left[L\left((\mathbf{x},\mathbf{z});\boldsymbol{\theta}\right))|\boldsymbol{\theta}^{(m)}\right]$$

  - Maximisation (étape M) :

$$\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}}\left(Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(\boldsymbol{m})}\right)\right)$$

  - $m = m + 1$
- Fin

Remarque : En pratique, pour s'affranchir du caractère local du maximum atteint, on fait tourner l'algorithme EM un grand nombre de fois à partir de valeurs initiales différentes de manière à avoir de plus grandes chances d'atteindre le maximum global de vraisemblance.

Application en classification automatique I

- Une des applications phares d'EM est l'estimation des paramètres d'une densité mélange en classification automatique dans le cadre des modèles de mélanges gaussiens. Dans ce problème, on considère qu'un échantillon $(x_1, \ldots, x_n)$ de $\mathbb{R}^p$, *i.e.* caractérisé par p variables continues, est en réalité issu de K différents groupes.
- En considérant que chacun de ces groupes G_k suit une loi f de paramètre θ_k, et dont les proportions sont données par un vecteur $(\pi_1, \ldots, \pi_K)$.

Application en classification automatique II

- En notant $\Phi = (\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K)$ le paramètre du mélange, la fonction de densité que suit l'échantillon est donnée par

$$g(x, \Phi) = \sum_{k=1}^{K} \pi_k f(x, \theta_k),$$

et donc, la log-vraisemblance du paramètre Φ est donnée par

$$L(x, \Phi) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k f(x_i, \theta_k) \right).$$

- La maximisation de cette fonction selon Φ est très complexe. Par exemple, si on souhaite déterminer les paramètres correspondant à 2 groupes suivant une loi normale dans un espace de dimension 3 (ce qui est peu), on doit optimiser une fonction non linéaire de $\mathbb{R}^{19}$

Application en classification automatique III

- Parallèlement, si on connaissait les groupes auxquels appartient chacun des individus, alors le problème serait un problème d'estimation tout à fait simple et très classique.
- La force de l'algorithme EM est justement de s'appuyer sur ces données pour réaliser l'estimation. En notant z_{ik} la grandeur qui vaut 1 si l'individu x_i appartient au groupe G_k et 0 sinon, la log-vraisemblance des données complétée s'écrit

$$L(x, z, \Phi) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \left(\pi_k f(x_i, \theta_k) \right).$$

On obtient alors

$$Q\left(\theta, \theta^{(m)}\right) = \sum_{i=1}^{n} \sum_{k=1}^{K} E\left(z_{ik} | x, \theta^{(m)}\right) \log \left(\pi_k f(x_i, \theta_k) \right)$$

Application en classification automatique IV

- On peut séparer l'algorithme EM en deux étapes, qu'on appelle classiquement, dans le cas des modèles de mélanges, l'étape Estimation et l'étape Maximisation.
- Ces deux étapes sont itérées jusqu'à la convergence.

Etape E

- En notant T_{ik} la quantité donnée par $T_{ik} = E\left(z_{ik}|x, \theta^{(m)}\right)$,
- Calcul de T_{ik} par la règle d'inversion de Bayes :

$$T_{ik} = \frac{\pi_k^{(m)} f(x_i, \theta_k^{(m)})}{\sum_{\ell=1}^{K} \pi_\ell^{(m)} f(x_i, \theta_\ell^{(m)})}$$

Application en classification automatique V

Etape M

- Détermination de Φ maximisant

$$Q\left(\theta, \theta^{(m)}\right) = \sum_{i=1}^{n} \sum_{k=1}^{K} T_{ik} \log\left(\pi_k f(x_i, \theta_k)\right)$$

- L'avantage de cette méthode est qu'on peut séparer le problème en K problèmes élémentaires qui sont, en général relativement simples.
- Dans tous les cas, les proportions optimales sont données par

$$\pi_k = \frac{1}{n} \sum_{i=1}^{n} T_{ik}$$

L'estimation des θ dépend par ailleurs de la fonction de probabilité f choisie.

Application en classification automatique VI

- Cas gaussien, il s'agit des moyennes μ_k et des matrices de variance-covariance Σ_k ainsi que les probabilités des classes. Les estimateurs optimaux sont alors donnés par

$$\mu_k = \frac{\sum_{i=1}^{n} T_{ik} x_i}{\sum_{i=1}^{n} T_{ik}}$$

et

$$\Sigma_k = \frac{\sum_{i=1}^{n} T_{ik}(x_i - \mu_k)(x_i - \mu_k)'}{\sum_{i=1}^{n} T_{ik}}$$

Application en classification automatique VII

En effet, dans le cas gaussien unidimensionnel, on a :

$$f(y;\theta) = \sum_{k=1}^{K} p_k \Phi(y, \mu_k, \sigma_k^2)$$

$$\log f(y,z;\theta) = \sum_{k=1}^{K} 1_{z=k}(\log p_k + \log(\Phi(y, \mu_k, \sigma_k^2))$$

Application en classification automatique VIII

Ainsi, la log-vraisemblance complète est donnée par :

$$\begin{aligned}
\log L(y_{1:n}, z_{1:n}\,;\,\theta) &= \sum_{i=1}^{n} \log f(y_i, z_i) \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K} 1_{z_i=k}(\log p_k + \log(\Phi(y, \mu_k, \sigma_k^2)) \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K} 1_{z_i=k}[\log p_k - \frac{1}{2}\log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2}(x_i - \mu_k)^2]
\end{aligned}$$

Application en classification automatique IX

Etape E : $Q(\theta|\theta^{(m)})$

$$
\begin{aligned}
Q(\theta|\theta^{(m)}) &= E_{Z|y,\theta^{(m)}}\left[\log \ell(y_1 \dots y_n, Z_1, \dots Z_n; \theta)\right] \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K} E[\mathbb{I}_{Z_i=k}|y_1 \dots y_n, \theta^{(m)}]\left[\log p_k - \frac{1}{2}\log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2}(x_i-\mu_k)^2\right] \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{P}(Z_i=k|y_1 \dots y_n, \theta^{(m)})\left[\log p_k - \frac{1}{2}\log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2}(x_i-\mu_k)^2\right]
\end{aligned}
$$

Or

$$
\begin{aligned}
\mathbb{P}(Z_i=k|y_{1:n}, \theta^{(}m)) &= \frac{f(y_i|Z_i=k;\theta^{(m)})\mathbb{P}(Z_i=k|y_i,\theta^{(m)})}{f(y_i|\theta^{(m)})} \\
&= \frac{p_k^{(m)}\phi(y_i|\mu_k^{(m)},(\sigma_k^{(m)})^2)}{\sum_{k=1}^{K} p_k^{(m)}\phi(y_i|\mu_k^{(m)},(\sigma_k^{(m)})^2)} \\
&:= T_{ik}^{(m)}
\end{aligned}
$$

D'où

$$
Q(\theta|\theta^{(m)}) = \sum_{i=1}^{n}\sum_{k=1}^{K} T_{ik}^{(m)}\left[\log p_k - \frac{1}{2}\log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2}(x_i-\mu_k)^2\right]
$$

Application en classification automatique X

Etape M : maximisation en θ de $Q(\theta|\theta^{(m)})$

Maximisation en $(p_1, p_2, \ldots, p_K)$ **avec** $\sum_{k=1}^{K} p_k = 1$

On utlise le Lagrangien : $Q(\theta|\theta^{(m)}) + \lambda(\sum_{k=1}^{K} p_k - 1)$

- $$\frac{\partial}{\partial p_k}\left[\sum_{i=1}^{n}\sum_{k=1}^{K} T_{ik}^{(m)}\left[\log p_k - \frac{1}{2}\log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2}(x_i-\mu_k)^2\right]\right] + \lambda = 0$$
  $$\frac{1}{p_k}\sum_{i=1}^{n} T_{ik}^{(m)} + \lambda = 0$$
- $\sum_{k=1}^{K} p_k = 1$.
- On obtient alors $-\lambda \underbrace{\sum_{k=1}^{K} p_k}_{=1} = \sum_{i=1}^{n} \underbrace{\sum_{k=1}^{K} T_{ik}^{(m)}}_{=1} = n$
- Par conséquent :

$$p_k^{(m+1)} = \frac{\sum_{i=1}^{n} T_{ik}^{(m)}}{n}$$

Application en classification automatique XI

Etape M : maximisation en θ de $Q(\theta|\theta^{(m)})$

Maximisation en $(\mu_1, \ldots, \mu_K)$

$$\frac{\partial}{\partial \mu_k} Q(\theta|\theta^{(m)}) = c\left[\sum_{i=1}^{n} T_{ik}^{(m)} y_i - \sum_{i=1}^{n} T_{ik}^{(m)} \mu_k\right]$$

$$\mu_k^{(m+1)} = \frac{\sum_{i=1}^{n} T_{ik}^{(m)} y_i}{\sum_{i=1}^{n} T_{ik}^{(m)}}$$

Application en classification automatique XII

Initialisation de l'algorithme EM

- pour les μ_k on prend les valeurs de K observations x_i tirées au hasard dans l'échantillon,
- pour les σ_k on prend l'écart-type empirique $s_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$, i.e. $\sigma_k = s_x, k = 1, \ldots, K$,
- pour les π_k on choisit des proportions équilibrées, i.e. $\pi_k = 1/K, k = 1, \ldots, K$.

Application en classification automatique XIII

Mise en œuvre sous R : le package mclust

- Description détaillé du package dans
  - C. Fraley and A. E. Raftery (2006). MCLUST Version 3 for R : Normal Mixture Modeling and Model-Based Clustering, Technical Report no. 504, Department of Statistics, University of Washington
  - C. Fraley and A. E. Raftery (2002). Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association 97 :611-631
- Utiliser la commande `Mclust`,
- `Mclust` applique l'EM-algorithme (avec des différentes paramétrisations), et utilise le BIC comme critère de sélection du modèle : chercher le modèle M_k correspondant au BIC maximal.

$$\text{BIC} = 2 \log \text{Vraisemblance} - \text{Nbre paramètres} \times \log(n)$$