Classification, Apprentissage, Décision

Classification non-supervisée : Regroupement (clustering)

C’est quoi ?

- Regroupement (Clustering): construire une collection d’objets
  - Similaires au sein d’un même groupe
  - Dissimilaires quand ils appartiennent à des groupes différents
- Le Clustering est de la classification non supervisée: pas de classes prédéfinies

Qu'est ce qu'un bon regroupement ?

- Une bonne méthode de regroupement permet de garantir
  - Une grande similarité intra-groupe
  - Une faible similarité inter-groupe
- La qualité d'un regroupement dépend donc de la mesure de similarité utilisée par la méthode et de son implémentation

Structures de données

- Matrice de données

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

- Matrice de similarité

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Mesurer la qualité d'un clustering

- Métrique pour la similarité: La similarité est exprimée par le biais d'une mesure de distance
- Une autre fonction est utilisée pour la mesure de la qualité
- Les définitions de distance sont très différentes selon que les domaines d'attributs sont des intervalles (continues), catégories, booléens.
- En pratique, on utilise souvent une pondération des attributs

Intervalle (discrètes) : pré-traitement

- Standardiser les données
  - Calculer l'écart absolu moyen:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + ... + |x_{nf} - m_f|)$$

ou

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + ... + x_{nf})$$

  - Calculer la mesure standardisée (*z-score*)

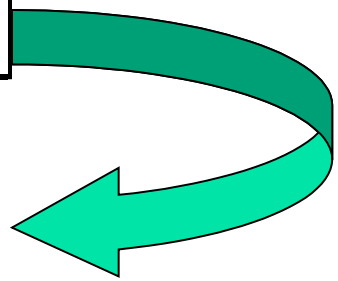
$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

Exemple

	05	
	07	
	06	
	06	

$$M_{Age} = 60 \qquad S_{Age} = 5$$

$$M_{salaire} = 11074 \qquad S_{salaire} = 148$$



	Age	Salaire
Personne 1	-2	-0,5
Personne 2	2	0.175
Personne 3	0	0,324
Personne 4	0	0

Similarité entre objets

- Les distances expriment une similarité
- Ex: la *distance de Minkowski* :

$$d(i,j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \ldots + |x_{ip} - x_{jp}|^q)}$$

où $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ et $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ sont deux objets p-dimensionnels et q un entier positif

- Si $q = 1$, d est la distance de Manhattan :

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ip} - x_{jp}|$$

- Si q = 2, d est la distance euclidienne :

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \ldots + |x_{ip} - x_{jp}|^2}$$

Exemple: distance de Manhattan

→ d(p1,p2)=120

d(p1,p3)=132

Conclusion: p1 ressemble plus à p2 qu'à p3 ☹

→ d(p1,p2)=4,675

d(p1,p3)=2,324

Conclusion: p1 ressemble plus à p3 qu'à p2 ☺

Attributs binaires

- Une table de contingence pour données binaires

Objet *i* \ Objet *j*	1	0	*sum*
1	*a*	*b*	*a+b*
0	*c*	*d*	*c+d*
sum	*a+c*	*b+d*	*p*

a : nombre de positions où i a 1 et j a 1

b : nombre de positions où i a 1 et j a 0

c : nombre de positions où i a 0 et j a 1

d : nombre de positions où i a 0 et j a 0

- Exemple o_i=(1,1,0,1,0) et o_j=(1,0,0,0,1) :

  a=1, b=2, c=1, d=2

Mesures de distances

- Coefficient d'appariement (matching) simple (invariant pour variables symétriques):

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

Ex. pour o_i=(1,1,0,1,0) et o_j=(1,0,0,0,1) d(o_i, o_j)=3/5

- Coefficient de Jaccard

$$d(i,j) = \frac{b+c}{a+b+c}$$

Ex. : d(o_i, o_j)=3/4

Attribut binaires (I)

- Attribut symétrique: Ex. le sexe d'une personne, i.e coder masculin par 1 et féminin par 0 c'est pareil que le codage inverse

- Attribut asymétrique: Ex. Test HIV. Le test peut être positif ou négatif (0 ou 1) mais il y a une valeur qui sera plus présente que l'autre.

  Généralement, on code par 1 la modalité la moins fréquente

  - 2 personnes ayant la valeur 1 pour le test sont *plus similaires* que 2 personnes ayant 0 pour le test

Attribut binaires (II)

- Exemple

	Genre	F	Test-1	Test-2	Test-3	Test-4	Test-5
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- Sexe est un attribut symétrique
- Les autres attributs sont asymétriques
- Y et P ≡ 1, N ≡ 0, la distance n'est mesurée que sur les asymétriques

$$d(jack, mary) = \frac{0 + 1}{2 + 0 + 1} = 0.33$$

$$d(jack, jim) = \frac{1 + 1}{1 + 1 + 1} = 0.67$$

$$d(jim, mary) = \frac{1 + 2}{1 + 1 + 2} = 0.75$$

Les plus similaires sont Jack et Mary ⇒ atteints du même mal

Attributs Nominaux

- Une généralisation des attributs binaires, ex: rouge, vert et bleu
- Méthode 1: Matching simple
  - *m* : # d'appariements, *p* : # total de variables

$$d(i,j) = \frac{p - m}{p}$$

- Méthode 2: utiliser un grand nombre d'attributs binaires
  - Créer un attribut binaire pour chaque modalité (ex: attribut rouge qui prend les valeurs vrai ou faux)

Attribut Ordinaux

- Un attribut ordinal peut être discret ou continu
- L'ordre peut être important, ex: classement
- Peuvent être traitées comme les variables intervalles
  - remplacer x_{if} par son rang $r_{if} \in \{1,\ldots,M_f\}$
  - Remplacer le rang de chaque variable par une valeur dans [0, 1] en remplaçant la variable f dans l'objet I par

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

  - Utiliser une distance pour calculer la similarité

En Présence d'attributs de différents Types

- Pour chaque type d'attributs, utiliser une mesure adéquate. Problèmes: les clusters obtenus peuvent être différents
- On utilise une formule pondérée pour faire la combinaison

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

  - f est binaire ou nominale:

    $d_{ij}^{(f)} = 0$ si $x_{if} = x_{jf}$, sinon $d_{ij}^{(f)} = 1$
  - f est de type intervalle: utiliser une distance normalisée
  - f est ordinale
    - calculer les rangs r_{if} et $z_{if} = \frac{r_{if} - 1}{M_f - 1}$
    - Ensuite traiter z_{if} comme un attribut de type intervalle

Clustering

Approches de Clustering

- Algorithmes de Partitionnement: Construire plusieurs partitions puis les évaluer selon certains critères
- Algorithmes hiérarchiques: Créer une décomposition hiérarchique des objets selon certains critères
- Algorithmes basés sur la densité: basés sur des notions de connectivité et de densité
- Algorithmes de grille: basés sur un structure à multi-niveaux de granularité
- Algorithmes à modèles: Un modèle est supposé pour chaque cluster. Puis vérifier chaque modèle sur chaque groupe pour choisir le meilleur

Algorithmes à partionnement

- Construire une partition à ***k*** clusters d'une base ***D*** de ***n*** objets
- Les *k clusters doivent* optimiser le critère choisi
  - Global optimal: Considérer toutes les k-partitions
  - Heuristic methods: Algorithmes *k-means* et *k-medoids*
    - *k-means* (MacQueen'67):

      Chaque cluster est représenté par son centre
    - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87):

      Chaque cluster est représenté par un de ses objets

La méthode des k-moyennes (*K-Means)*

- L'algorithme *k-means* est en 4 étapes :

1) Choisir k objets M_i formant ainsi k clusters C_i

2) (Ré)affecter chaque objet *o* au cluster C_i de centre M_i tel que dist(*o*,M_i) est minimal

3) Recalculer M_i de chaque cluster (le barycentre)

4) Aller à l'étape 2 si on vient de faire une affectation

K-Means : Exemple

- A={1,2,3,6,7,8,13,15,17}. Créer 3 clusters à partir de A

- On prend 3 objets au hasard. Supposons que c'est 1, 2 et 3. Ca donne C_1={1}, M_1=1, C_2={2}, M_2=2, C_3={3} et M_3=3

- Chaque objet *o* est affecté au cluster au milieu duquel *o* est le plus proche. 6 est affecté à C_3 car dist(M_3,6)<dist(M_2,6) et dist(M_3,6)<dist(M_1,6)

  On a C_1={1}, M_1=1,

  C_2={2}, M_2=2

  C_3={3, 6,7,8,13,15,17}, M_3=69/7=9.86

K-Means : Exemple (suite)

- dist(3,M2)<dist(3,M3) ➔3 passe dans C_2.
  Tous les autres objets ne bougent pas.
  C_1={1}, M_1=1, C_2={2,3}, M_2=2.5, C_3={6,7,8,13,15,17} et M_3= 66/6=11

- dist(6,M_2)<dist(6,M_3)➔6 passe dans C2.
  Tous les autres objets ne bougent pas.
  C_1={1}, M_1=1, C_2={2,3,6}, M_2=11/3=3.67, C3={7,8,13,15,17}, M_3= 12

- dist(2,M1)<dist(2,M2)➔2 passe en C1.
  dist(7,M2)<dist(7,M3)➔ 7 passe en C2.
  Les autres ne bougent pas.
  C1={1,2}, M1=1.5, C2={3,6,7}, M2=5.34, C3= {8,13,15,17}, M3=13.25

- dist(3,M1)<dist(3,M2)➔3 passe en 1.
  dist(8,M2)<dist(8,M3)➔8 passe en 2
  C1={1,2,3}, M1=2, C2={6,7,8}, M2=7, C3={13,15,17}, M3=15

➔ Plus rien ne bouge : l'algorithme s'arrête

Commentaires sur la méthode des *K-Means*

- Force
  - *Relativement efficace*: *O*(*tkn*), où *n* est # objets, *k* est # clusters, et *t est* # itérations. Normalement *k*, *t* << *n*.
  - Tend à réduire $E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$
- Faiblesses
  - N'est pas applicable en présence d'attributs qui ne sont pas du type intervalle (moyenne=?)
  - On doit spécifier k (nombre de clusters)
  - Les clusters sont construits par rapports à des objets inexistants (les milieux)
  - Ne peut pas découvrir les groupes *non-convexes*

La méthode des *K-Medoids (PAM)*

- Trouver des objets représentatifs (medoïdes) dans les clusters (au lieu de la moyenne)
- Principe
  - Commencer avec un ensemble de medoïdes puis itérativement remplacer un par un autre si ça permet de réduire la distance globale
  - Efficace pour des données de petite taille

Algorithme des k-Medoïdes

Choisir arbitrairement k medoides

Répéter

affecter chaque objet restant au medoïde le plus proche

Choisir aléatoirement un non-medoïde O_r

Pour chaque medoïde O_j

Calculer le coût TC du remplacement de O_j par O_r

Si TC < 0 alors

Remplacer O_j par O_r

Calculer les nouveaux clusters

Jusqu'à ce ce qu'il n'y ait plus de changement

PAM (Partitioning Around Medoids) (1987)

Choisir arbitrairement ***k*** objets représentatifs

- Pour toute paire (h,j) d'objets t.q h est choisi et j non, calculer le coût $\boldsymbol{TC_{jh}}$ du remplacement de j par h
  - Si $TC_{ih} < 0$, ***j*** est remplacé par ***h***
  - Puis affecter chaque objet non sélectionné au medoïde qui lui est le plus similaire
- Répéter jusqu'à ne plus avoir de changements

La méthode des *K-Medoids*

- TC_{jh} représente le gain en distance globale que l'on va avoir en remplaçant h par j

- Si TC_{jh} est négatif alors on va perdre en distance. Ca veut dire que les clusters seront plus compacts.

- $TC_{jh}=\sum_i dist(h,i)-dist(j,i)= \sum_i C_{ijh}$

La méthode des *K-Medoids:* Exemple

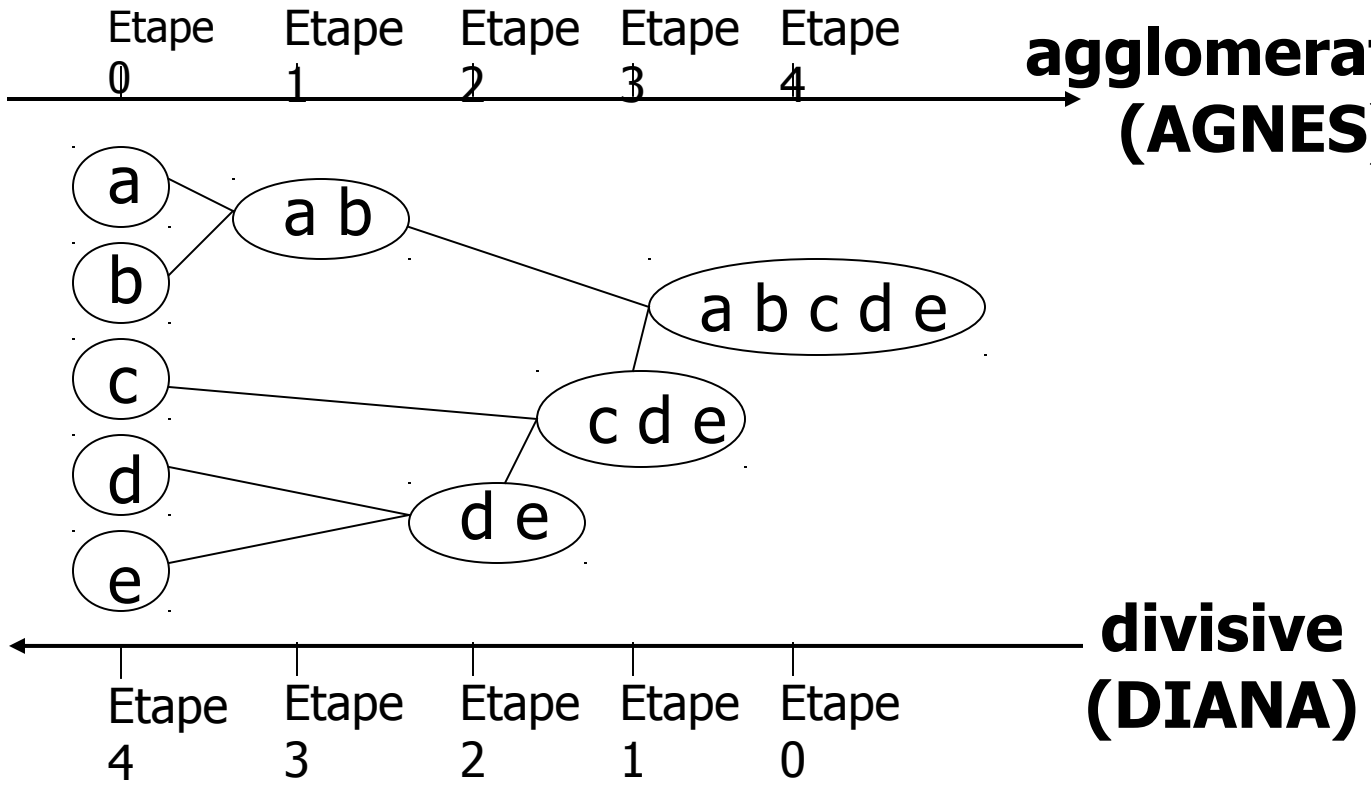
- Soit A={1,3,4,5,8,9}, k=2 et M={1,8} ensemble des medoides
  ➔ C1={1,3,4} et C2={5,8,9}
  $E_{\{1,8\}}$=dist(3,1)^2+dist(4,1)^2+dist(5,8)^2+dist(9,8)^2=23

- Comparons 1 et 3 ➔ M={3,8} ➔ C1={1,3,4,5} et C2={8,9}
  $E_{\{3,8\}}$ =dist(1,3)^2+dist(4,3)^2+dist(5,3)^2+dist(9,8)^2=10
  $TC_{1,3}$ = $E_{\{3,8\}}$ - $E_{\{1,8\}}$= -13 <0 donc le remplacement est fait.

- Comparons 3 et 4 ➔ M={4,8} ➔ C1 et C2 inchangés et
  $E_{\{4,8\}}$=dist(1,4)^2+dist(3,4)^2+dist(5,4)^2+dist(8,9)^2= 12➔ 3 n'est pas remplacé par 4
- Comparons 3 et 5 ➔ M={5,8} ➔ C1 et C2 inchangés et E{5,8}>E{3,8}

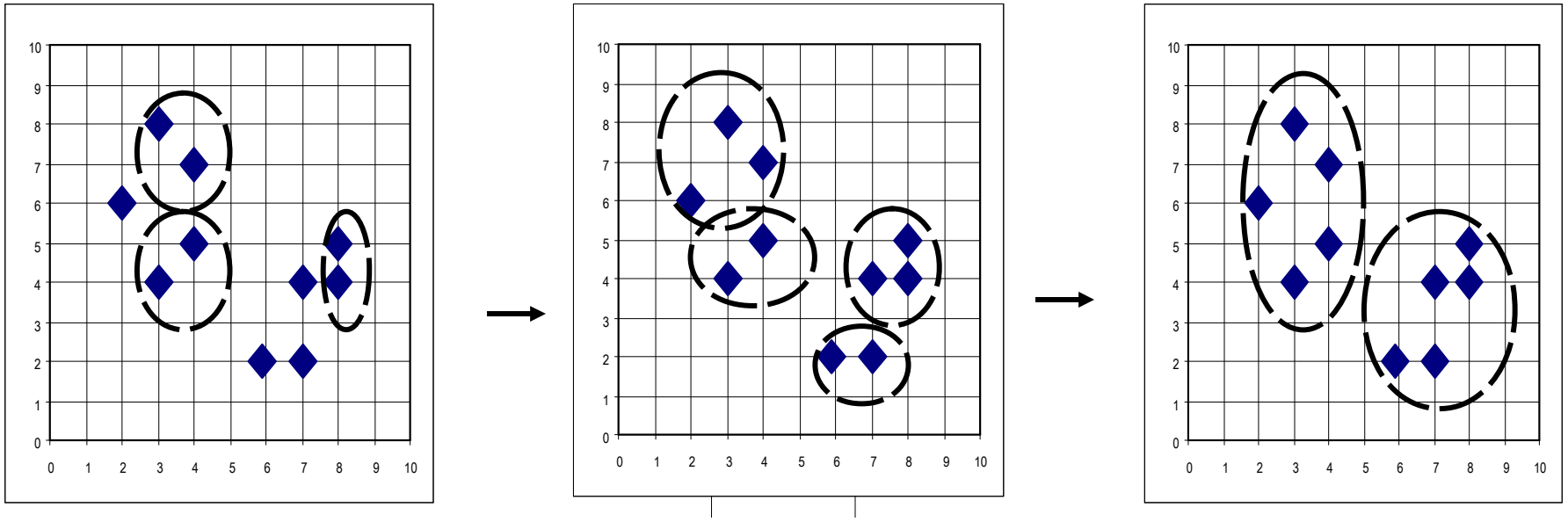
Clustering Hiérarchique

- Utiliser la matrice de distances comme critère de regroupement. ***k*** n'a pas à être précisé, mais a besoin d'une condition d'arrêt



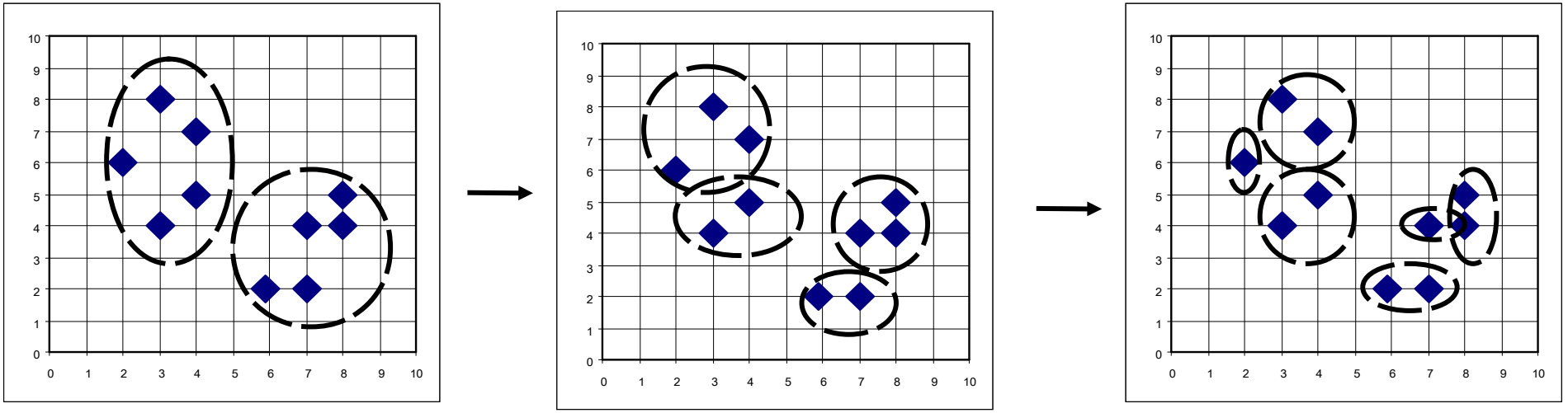
AGNES (Agglomerative Nesting)

- Utilise la matrice de dissimilarité.
- Fusionne les nœuds qui ont la plus faible dissimilarité
- On peut se retrouver dans la situation où tous les nœuds sont dans le même groupe



DIANA (Divisive Analysis)

- L'ordre inverse de celui d'AGNES
- Il se peut que chaque objet forme à lui seul un groupe



Critères de fusion-éclatement

- Exemple: pour les méthodes agglomératives, C1 et C2 sont fusionnés si
  - il existe o1 ∈ C1 et o2∈ C2 tels que dist(o1,o2) ≤ seuil, ou
  - il n'existe pas o1 ∈ C1 et o2∈ C2 tels que dist(o1,o2) ≥ seuil, ou
  - distance entre C1 et C2 ≤ seuil avec

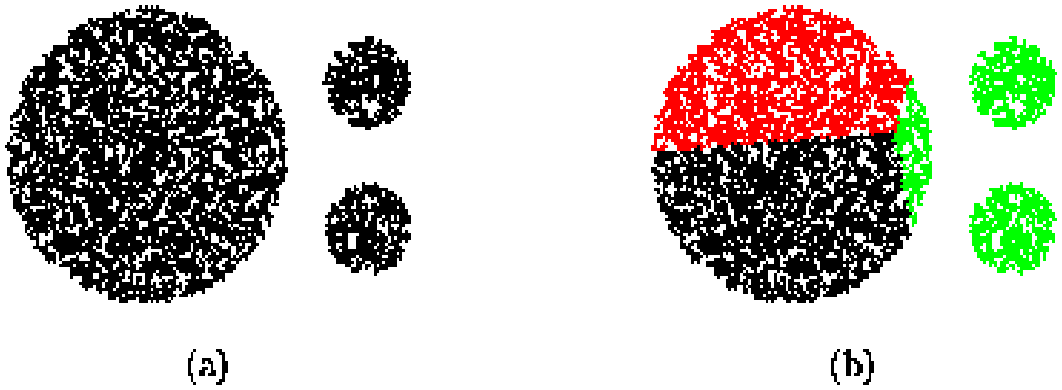
Lien unique

$$dist(C_1, C_2) = \frac{1}{n1 * n2} \sum_{o1 \in C1, o2 \in C2} dist(o1, o2)$$

et n1=|C1|.

- Ces techniques peuvent être adaptées pour les méthodes divisives

CURE (Clustering Using REpresentatives)



- Les méthodes précédentes donnent les groupes (b)
- CURE: (1998)
  - Arrête la création de clusters dès qu'on en a *k*
  - Utilise plusieurs points représentatifs clusters

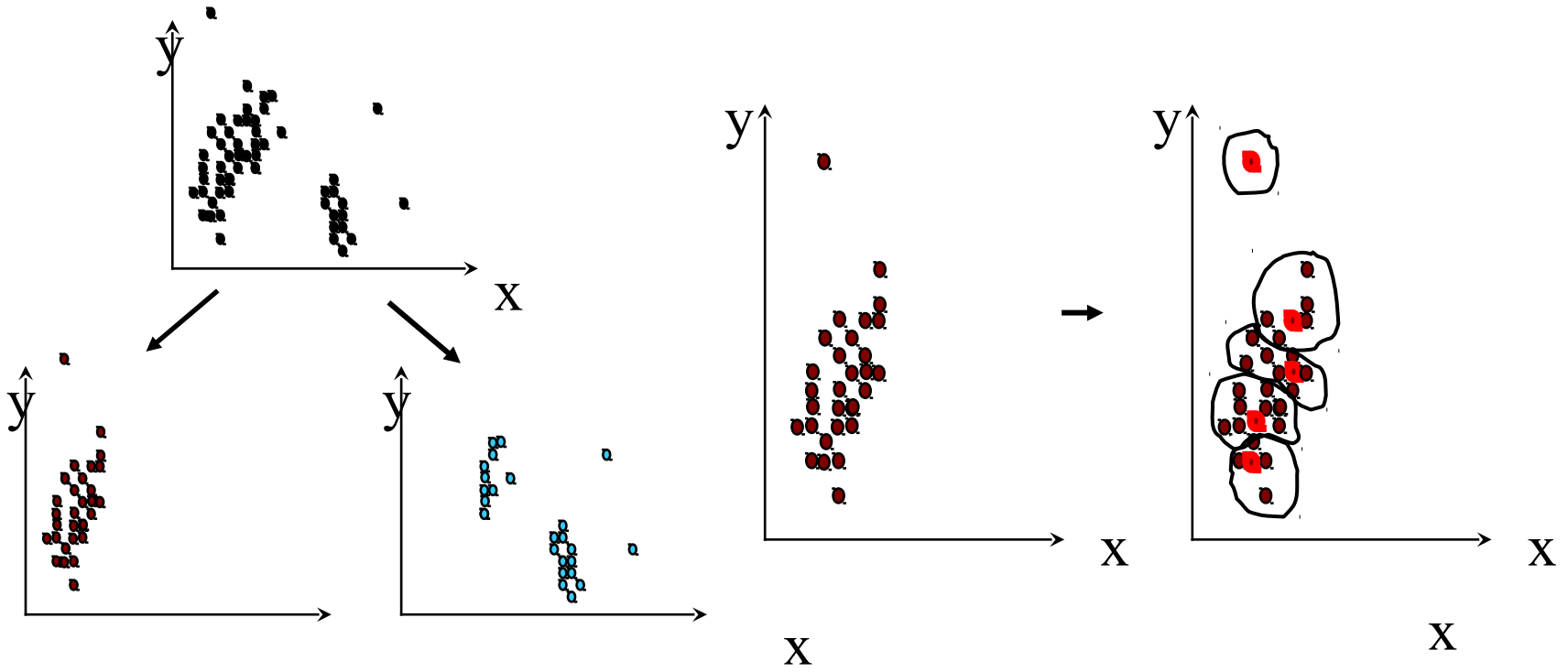
Cure: l'algorithme

- Prendre un sous-ensemble *s*
- Partitionner s en *p* partitions de taille *s/p*
- Dans chaque partition, créer *s/pq* clusters
- Eliminer les exceptions (points aberrants)
- Regrouper les clusters partiels

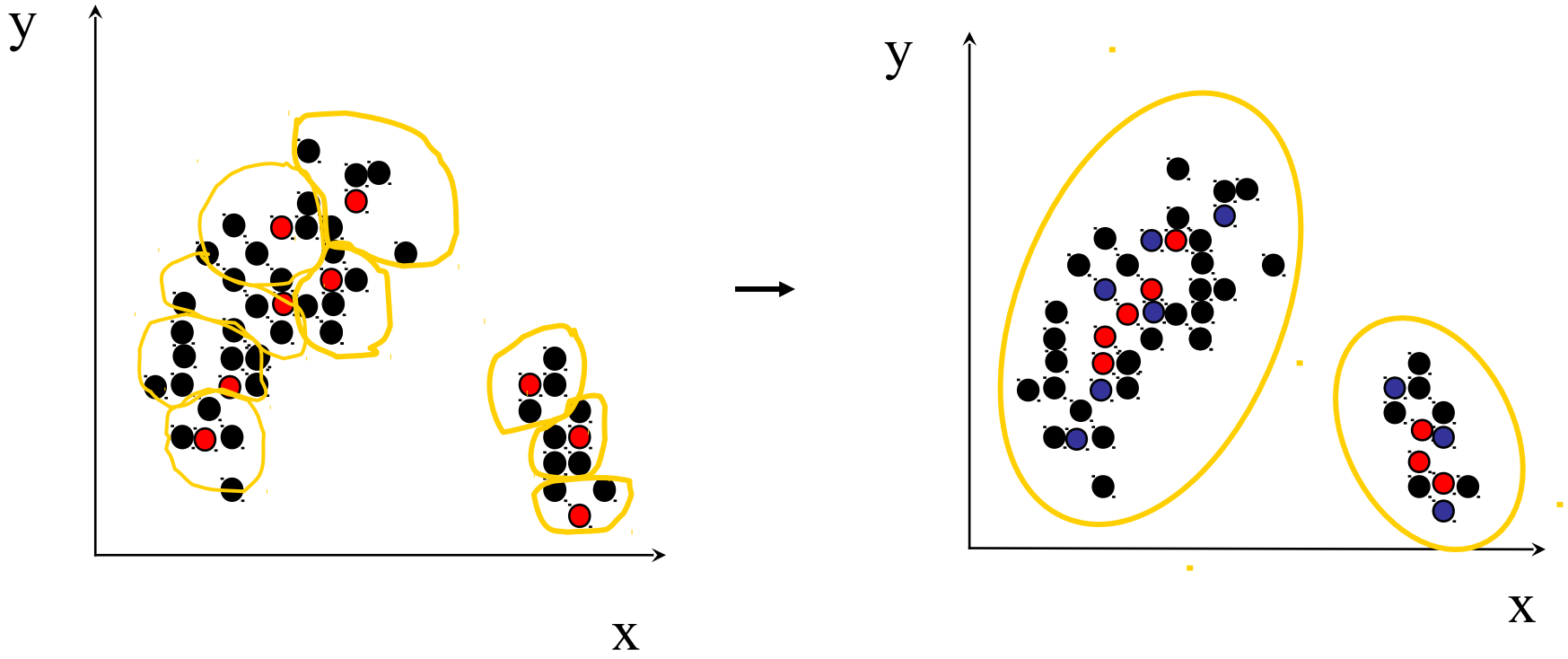
Partitionnment et Clustering

- s = 50
- p = 2
- s/p = 25

- s/pq = 5



Cure: Rapprochement des points représentatifs



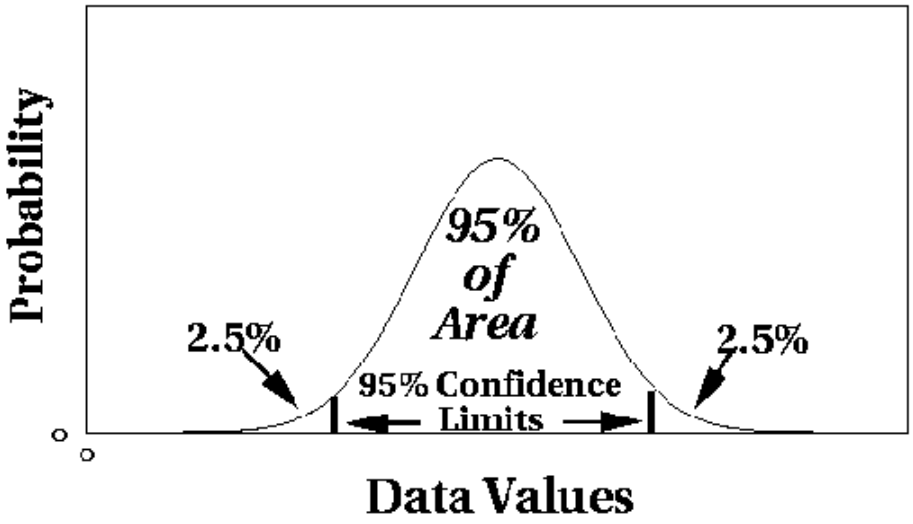
- Rapprocher les points représentatifs vers le centre de gravité par un facteur α.
- Plusieurs points représentatifs permettent de figurer la forme du cluster

Autres utilisations de métriques

Découverte d'exceptions

- Ce sont les objets qui sont considérablement différents du reste, exemple: ornithorynque, kiwi
- Problème
  - Trouver *n* objets qui sont les plus éloignés du reste
- Applications:
  - fraude
  - Analyse médicale
  - …

Approache statistique



On suppose que les données suivent une loi de distribution statistique (ex: loi normale)

- Utiliser les tests de discordance
  - Proba(Xi=val)< β alors X est une exception
- Problèmes
  - La plupart des tests sont sur un attribut
  - Dans beaucoup de cas, la loi de distribution est inconnue

Approche Basée sur la Distance

- Une (α, β)-exception est un object *o* dans T tel qu'il y a au moins α objets *o'* de T avec dist(*o*,*o'*)> β
- Une exception est donc une donnée où α et β sont grands.