Regroupement

(clustering)

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 distance

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

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 que les variables soient des intervalles (continues), catégories, booléennes ou ordinales
- En pratique, on utilise souvent une pondération des variables

Types des variables

- Intervalles:
- Binaires:
- catégories, ordinales, ratio:
- Différents types:

Intervalle (continues, réelles)

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

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

où

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

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

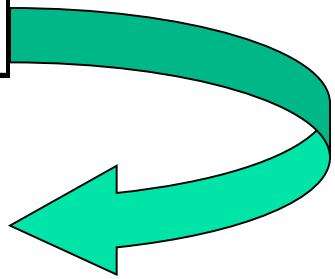
$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

Exemple

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

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

$$M_{salaire} = 11074 \qquad S_{salaire} = 37$$



Personne1	-2	-2
Personne2	2	0,7027027
Personne3	0	1,2972973
Personne4	0	0

Similarité entre objets

- Distance en O1 et O2 = 1/(similarité entre O1 et O2)
  - Si on a une fonction de distance => on a une fonction de similarité
- Ex: la *distance de Minkowski* :

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + ... + |x_{i_p} - x_{j_p}|^q)}$$

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

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

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + ... + |x_{i_p} - x_{j_p}|$$

Similarité entre objets(I)

- *Si q = 2, d est la distance Euclidienne* :

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + ... + |x_{i_p} - x_{j_p}|^2)}$$

  - Propriétés
    - $d(i,j) \geq 0$
    - $d(i,i) = 0$
    - $d(i,j) = d(j,i)$
    - $d(i,j) \leq d(i,k) + d(k,j)$

Exemple: distance de Manhattan

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

→ d(p1,p2)=120

d(p1,p3)=132

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

	Age	Salaire
Personne1	-2	-2
Personne2	2	0,7
Personne3	0	1,3
Personne4	0	0

→ d(p1,p2)=6,7

d(p1,p3)=5,3

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

Variables binaires

- Une table de contingence pour données binaires

		Objet *j*		
		1	0	*sum*
Objet *i*	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

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

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

Mesures de distances

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

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

Exemple 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

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

$d(o_i, o_j)$=3/4

Variables binaires (I)

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

- Variable asymétrique: Ex. Test HIV. Le test peut être positif ou négatif (1 ou 0) 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

Variables binaires(II)

- Exemple

Nom	Sexe	Fièvre	Toux	Test-1	Test-2	Test-3	Test-4
Jack	M	O	N	P	N	N	N
Mary	F	O	N	P	N	P	N
Jim	M	O	P	N	N	N	N

- Sexe est un attribut symétrique
- Les autres attributs sont asymétriques
- O 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

Variables Nominales

- Une généralisation des variables 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 de variables binaires
  - Créer une variable binaire pour chaque modalité (ex: variable rouge qui prend les valeurs vrai ou faux)

Variables Ordinales

- Une variable ordinale peut être discrète ou continue
- 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 de Variables de différents Types

- Pour chaque type de variables 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 $x_{if} = x_{jf}$
  - *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 une variable de type intervalle

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 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* :
  - Choisir k objets formant ainsi k clusters
  - Répéter
    - affecter chaque objet O au cluster C_i de centre M_i tel que dist(O,M_i) est minimale
    - Recalculer M_i de chaque cluster (le barycentre)
  - Jusqu'à ce qu'il n'y ait plus de changement

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}, M3=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

Commentaires sur la méthode des *K-Means*

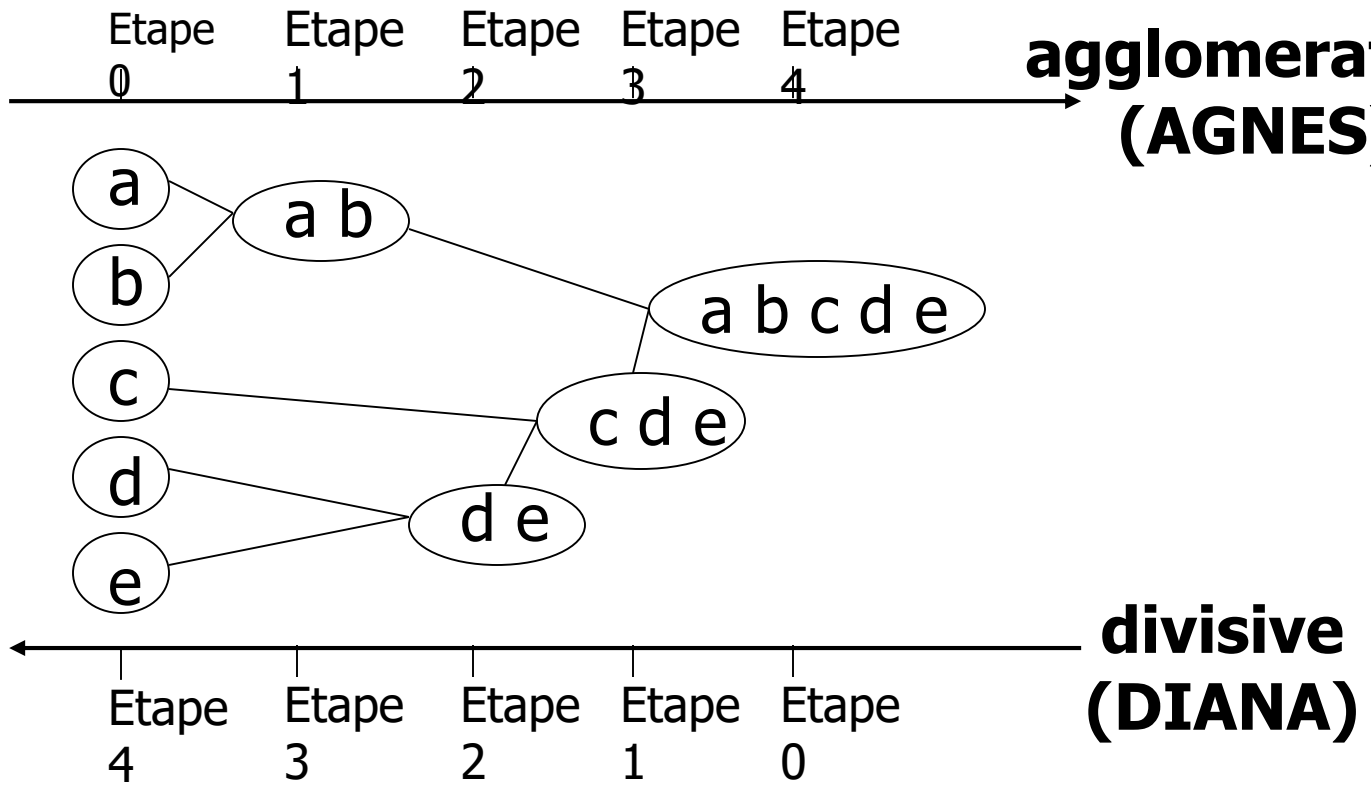
- Force
  - *Relativement efficace*: *O*(*tkn*), où *n* est # objets, *k* est # clusters, et *t est* # itérations. Normalement, *k*, *t* << *n*.
  - Tend à réduire

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \left| p - m_i \right|^2$$

- 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*
  - Sensible aux exceptions

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



Critères de fusion-éclatement

- Exemple: pour les méthodes agglomératives, C1 et C2 sont fusionnés si
  - Lien unique
    - il existe o1 $\in$ C1 et o2$\in$ C2 tels que dist(o1,o2) $\leq$ seuil, ou
    - il n'existe pas o1 $\in$ C1 et o2$\in$ C2 tels que dist(o1,o2) $\geq$ seuil, ou
  - distance entre C1 et C2 $\leq$ seuil avec

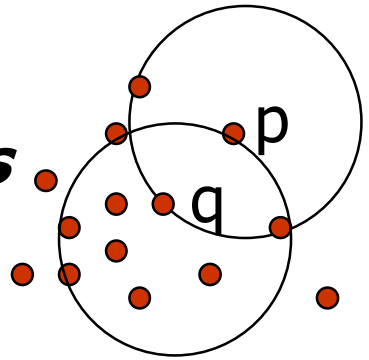
$$dist\,(C_1, C_2) = \frac{1}{n1 * n2} \sum_{o1 \in C1, o2 \in C2} dist(o1, o2)$$

et n1=|C1|.

- Ces techniques peuvent être adaptées pour les méthodes divisives

Clustering basé sur la densité: DBscan

- Voit les clusters comme des régions denses séparées par des régions qui le sont moins (bruit)
- Deux paramètres***:***
  - ***Eps***: Rayon maximum du voisinage
  - ***MinPts***: Nombre minimum de points dans le voisinage-Eps d'un point
- ***Voisinage : $V_{Eps}(p)$***: ***{q ∈ D \| dist(p,q) <= Eps}***
- Un point **q** est directement densité-accessible à partir de **p** resp. à ***Eps***, ***MinPts*** si
  - 1) ***p*** ∈ ***$V_{Eps}(q)$***
  - 2) \| ***$V_{Eps}(q)$***\| >= ***MinPts***



MinPts = 5

Eps = 1

Principe de DBSCAN

- p est attractif ssi le nombre de ses voisins est supérieur à MinPts
- A partir de l'ensemble de points **E** sur lequel on veut appliquer DBSCAN, on construit un graphe **G** comme suit:
  - Les sommets de **G** sont les points de **E**
  - Deux sommets p et q sont reliés par une arête si et seulement si
    - ils sont voisins: dist(p,q)<Eps et
    - Un des deux est attractif
- Une fois le graphe construit, deux sommets feront partie du même groupe ssi il y a un chemin dans le graphe **G** qui les relie.

DBSCAN : intuitivement

- p et q sont dans le même cluster ssi on peut aller de l'un vers l'autre *en restant dans une zone dense*.

- Avec DBSCAN, deux points très distants l'un de l'autre, donc très différents, peuvent appartenir au même groupe

- DBSCAN peut retourner des groupes non convexes (contrairement à K-Means)