Fragmentation

Généralités

Définitions

- Fragmentation (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

Mesures associées fragment

- Le centre $\vec{C} = \frac{1}{N}\sum_{i=1}^{N}\vec{X}_i$
- Le rayon $\vec{R} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\vec{X}_i - \vec{C})^2}$
- Le diamètre $\vec{D} = \sqrt{\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N}(\vec{X}_i - \vec{X}_j)^2}$

Types de Fragmentation

- 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 modèles: Un modèle est supposé pour chaque cluster ensuite vérifier chaque modèle sur chaque groupe pour choisir le meilleur

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

Mesurer la qualité d'une fragmentation

- 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 en fonction des variables
- En pratique, on utilise souvent une pondération des variables

z-score

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

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

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

  - Calculer la mesure standardisée

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

Exemple

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

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

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

	Age	Salaire
Personne1	-2	-0,5
Personne2	2	0,175
Personne3	0	0,324
Personne4	0	2

Similarité : Variables numériques

- La *distance de Minkowski*

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

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 $q = 1$, la distance de Manhattan
- Si q=2, distance Euclidienne

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	-0,5
Personne2	2	0,175
Personne3	0	0,324
Personne4	0	0

→ d(p1,p2)=4,675
d(p1,p3)=2,324

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

Similarité : Variables booléennes

- 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 vaut 1 et j vaut 1

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

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

Similarité : Variables booléennes

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

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

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

- Coefficient de Jaccard

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

Similarité : Variables booléennes

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

Similarité : Variables booléennes

- 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

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

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

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

Similarité : 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)

Similarité : 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

Algorithmes à partionnement

(partition à ***k*** fragments de ***n*** objets)

k-Means

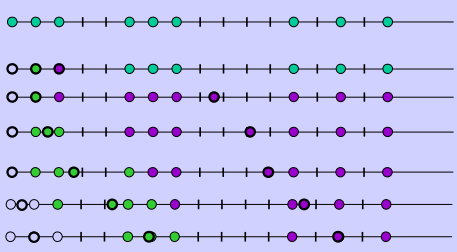
- Méthode des k-moyennes choisir K éléments initiaux "centres" des K groupes
  - placer les objets dans le groupe de centre le plus proche
  - recalculer le centre de gravité de chaque groupe
  - itérer l'algorithme jusqu'à ce que les objets ne changent plus de groupe
- Encore appelée méthode des centres mobiles

k-Means : Algorithme

- J. MacQueen, 1967

  Some methods for classification and analysis of multivariate observations. In Proceedings of 5 'h Berkeley Symposium on Mathematics, Statistics and Probabilii7, 1:281- 298
- Étapes:
  - fixer le nombre de fragments: k
  - choisir aléatoirement k tuples (graines)
  - assigner chaque tuple à la graine la plus proche
  - recalculer les k graines
  - tant que les k graines ont été changées
    - réassigner les tuples
    - recalculer les k graines

k-Means- Exemple A={1,2,3,6,7,8,13,15,17}



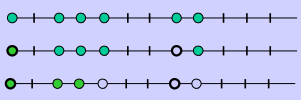
k-Means

- Force
  - *Complexité O*(*tkn*), où *n* est # objets, *k* est # clusters, et *t est* # itérations.
- Faiblesses
  - N'est pas applicable en présence d'attributs qui ne sont pas du type intervalle
  - On doit spécifier k (nombre de clusters)
  - Les clusters sont construits par rapports à des objets inexistants (les milieux)
  - Sensible aux exceptions

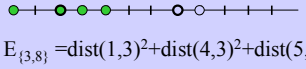
k-Médoïdes : Algorithme

- L. Kaufman, P.J. Rousseeu
  Finding Groups in Data, Wiley New York, 1989
- Etapes
  - Choisir arbitrairement k médoïdes
  - Répéter
    Affecter chaque objet restant au médoïde le plus proche
    Choisir aléatoirement un non-médoïde O_r
    Pour chaque médoïde 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
    Finsi
    FinPour
    Jusqu'à ce ce qu'il n'y ait plus de changement

k-Médoïdes- Exemple A={1,3,4,5,8,9}

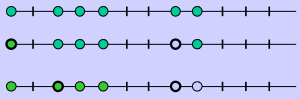


$E_{\{1,8\}}=dist(3,1)^2+dist(4,1)^2+dist(5,8)^2+dist(9,8)^2=23$



$E_{\{3,8\}}=dist(1,3)^2+dist(4,3)^2+dist(5,3)^2+dist(9,8)^2=10$

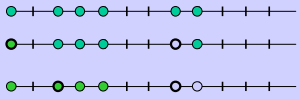
k-Médoïdes- Exemple A={1,3,4,5,8,9}



$E_{\{3,8\}} = dist(1,3)^2+dist(4,3)^2+dist(5,3)^2+dist(9,8)^2=10$

$E_{\{4,8\}}=dist(1,4)^2+dist(3,4)^2+dist(5,4)^2+dist(8,9)^2= 12$

k-Médoïdes- Exemple A={1,3,4,5,8,9}



$E_{\{3,8\}} = dist(1,3)^2+dist(4,3)^2+dist(5,3)^2+dist(9,8)^2=10$

$E_{\{5,8\}}=dist(1,5)^2+dist(3,5)^2+dist(4,5)^2+dist(8,9)^2= 22$

k-Médoïdes

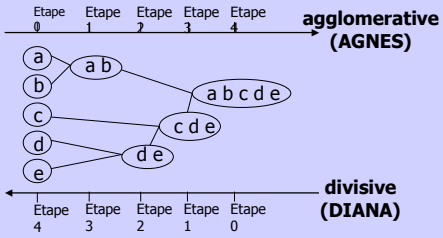
- Force
  - Les clusters sont construits par rapports à des objets existants
  - Peu sensible aux exceptions
- Faiblesses
  - Complexité O(k(n-k)2)
  - N'est pas applicable en présence d'attributs qui ne sont pas du type intervalle
  - On doit spécifier k (nombre de clusters)

Algorithmes hiérarchique

(nombre de fragments non précisé)

Algorithme Hiérarchique

Utilise une matrice de distances comme critère de regroupement.



Algorithme Hiérarchique : critère

- On fixe un seuil
- C1 et C2 sont fusionnés si
  - il existe o1 ∈ C1 et o2∈ C2 tels que dist(o1,o2) ≤ seuil
  - il n'existe pas o1 ∈ C1 et o2∈ C2 tels que dist(o1,o2) ≥ seuil
  - distance entre C1 et C2 ≤ seuil avec

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

- méthodes divisives : On adapte

BIRCH (1996)

•Tian Zhang, Raghu Ramakrishnan, and Miron Livny, 1996
BIRCH: An Efficient Data Clustering Method for Very Large Databases ACM SIGMOD 1996, pages 103--114, 1996.

•Algorithme

Construction t incrémentalement un arbre (CF-tree)

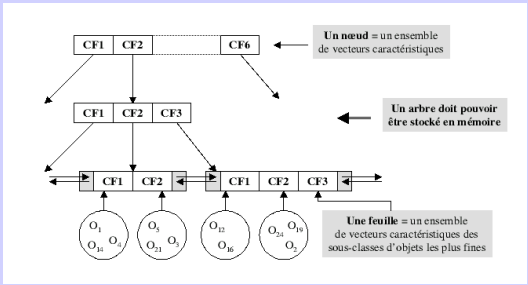
- chaque niveau représente une phase de clustering, chaque feuille est un ensemble d'éléments

– Construction de l'arbre

- identification du fragment en utilisant une distance (descente dans l'arbre)
- ajout de l'élément à la feuille
- si le diamètre dépasse un seuil on découpe la feuille

–méthode traditionnelles sur les feuilles

BIRCH (1996)



BIRCH (1996)

- Force
  - Trouve des clusters en une seule passe
  - Ajustement automatique du seuil
- Faiblesses
  - Complexité O(dim*n*hauteur(arbre))
  - Ne considère que les données numériques
  - Sensible à l'ordre de parcours

Basé sur les statistiques

Quel Effet ?

Isoler des sous arbres trop "petits" ou trop "gros" par rapport à la valeur moyenne d'un paramètre

Sommets $\xrightarrow{L}$ **R**

Comment décider?

Pour un arbre de taille n,
Construire $[\beta_n, \gamma_n]$

si L(s)$\notin [\beta_n, \gamma_n]$ alors l'arbre est trop !

Basé sur les statistiques : feuilles

Combien d'arbre de taille n ayant k feuilles ?

$$B_{n,k} = \frac{1}{n-1}\binom{n-1}{k}\binom{n-1}{k-1}$$

Combien d'arbres de taille n ?

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

Probabilité d'avoir un arbre de taille n ayant k feuilles

$$p_n(k) = \frac{B_{n,k}}{C_n}$$

Basé sur les statistiques : feuilles

Probabilité d'avoir un arbre de taille n ayant k feuilles

$$p_n(k) = \frac{B_{n,k}}{C_n}$$

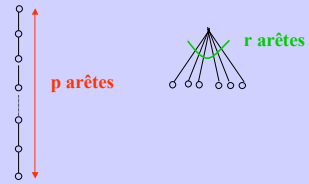
Moyenne et Ecart Type

$\mu_L = n/2$ $\sigma_L = (n/8)^{(1/2)}$

Si n>10, L suit une loi de Gauss

$$[\beta_n, \gamma_n] = \left[\frac{n}{2} - u_\alpha\sqrt{\frac{n}{8}}, \frac{n}{2} + u_\alpha\sqrt{\frac{n}{8}}\right]$$

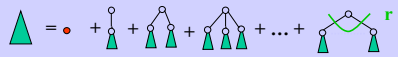
Basé sur les statistiques : feuilles : Exemple



A_r = {arbres ayant une arité au plus r}
B_p = {arbres ayant des segments de longueur au plus p}
$C_{r,}$ = {arbres ayant une arité au plus r
et des segments de longueur au plus }

Auber, Delest, 2002

Basé sur les statistiques : feuilles : arité bornée



$$A_r = x\,y + x\,A_r + x\,A_r^2 + x\,A_r^3 + \ldots + x\,A_r^r$$

$$A_r = \frac{x\,(1 - A_r^{r+1})}{1 - A_r} + x\,y - x$$

Basé sur les statistiques : feuilles : arité bornée

Calculer la plus petite racine de

$$\left\{ A_r = \frac{x\,(1 - A_r^{r+1})}{1 - A_r} + x\,y - x \right|_{y=1}$$

$$\left\{ \frac{d}{dA_r}\left(A_r = \frac{x\,(1 - A_r^{r+1})}{1 - A_r} + x\,y - x \right) \right|_{y=1}$$

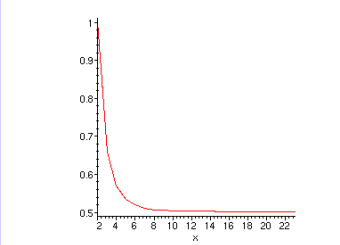
$$\Downarrow$$

$$1 - 2\,A_r + A_r^{r+2} + r\,A_r^{r+1} - r\,A_r^{r+2} = 0$$

Basé sur les statistiques : feuilles : arité bornée

Calculer la plus petite racine de

$$1 - 2\,A_r + A_r^{r+2} + r\,A_r^{r+1} - r\,A_r^{r+2} = 0$$



Basé sur les statistiques : feuilles : arité bornée

la plus petite singulatité est

$$a_r\ ,\ x_r$$

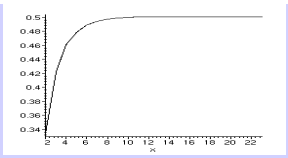
Moyenne

$$G = \frac{x\,(1 - A_r^{r+1})}{1 - A_r} + x\,y - x$$

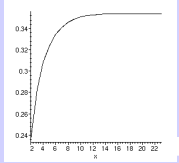
$$\mu_r = \left. \frac{\frac{dG}{dy}}{x\,\frac{dG}{dx}} \right| \quad y = 1,\ x = x_r,\ A_r = a_r$$

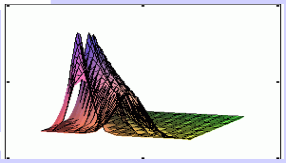
Ecart-Type …..

Basé sur les statistiques : feuilles : arité bornée



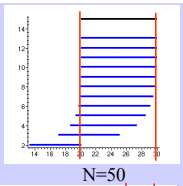
Moyenne



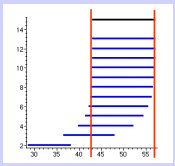


Ecart-Type

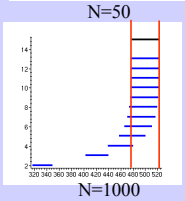
Basé sur les statistiques : feuilles : arité bornée



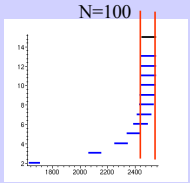
N=50



N=100

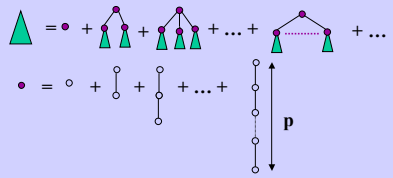


N=1000



N=5000

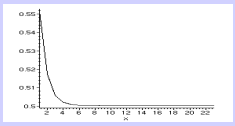
Basé sur les statistiques:feuilles:segment borné



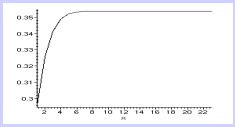
$$\begin{cases} B_p = z\,y + z\,B_p^2\left(\dfrac{1}{1 - B_p}\right) \\ z = \dfrac{x\,(1 - x^{p+1})}{1 - x} \end{cases}$$

…… résoudre ….

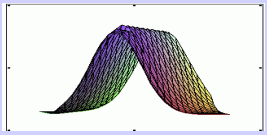
Basé sur les statistiques:feuilles:segment borné



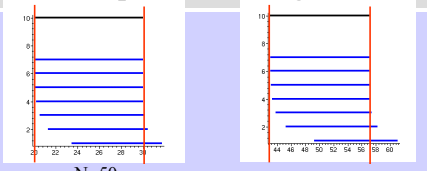
Moyenne



Ecart-type

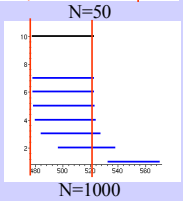


Basé sur les statistiques:feuilles:segment borné

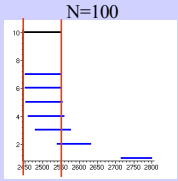


N=50

N=100

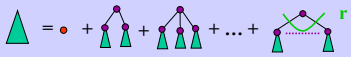


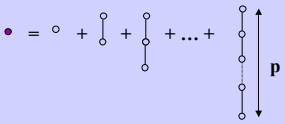
N=1000



N=5000

Basé sur les statistiques:feuilles: arité et segment borné

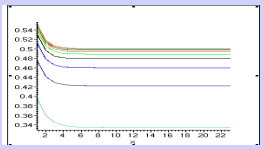


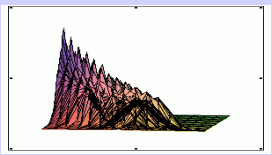


$$\begin{cases} C_{r,p} = z\,y + z\,C_{r,p}^{2}\left(\dfrac{1}{1 - C_{r,p}}\right) - z - z\,C_{r,p} \\ z = \dfrac{x\,(1 - x^{p+1})}{1 - x} \end{cases}$$

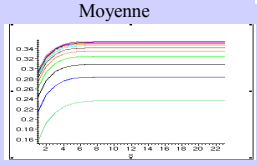
…… Résoudre ….

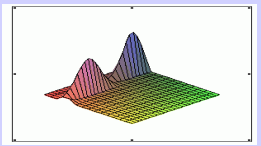
Basé sur les statistiques:feuilles: arité et segment borné





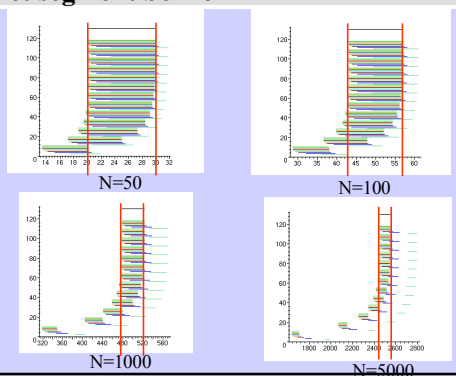
Moyenne





Ecart-type

Basé sur les statistiques:feuilles: arité et segment borné



Basé sur les statistiques

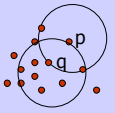
- Force
  - Complexité O(n)
  - Découverte de motifs aberrants
- Faiblesses
  - Necessite d'un modèle
  - En général 1 modèle=1 méthode

Algorithmes basé sur la densité

(nombre de fragments non précisé)

Principe

- 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 ***p*** est directement densité-accessible à partir de ***q*** resp. à ***Eps***, ***MinPts*** si
  - 1) ***$p \in V_{Eps}(q)$***
  - 2) ***$|V_{Eps}(q)| >= MinPts$***



MinPts = 5

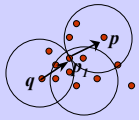
Eps = 1

Principes

- Accessibilité:
  - p est accessible à partir de q resp. à Eps, $MinPts$ si il existe $p_1, \ldots, p_n$:

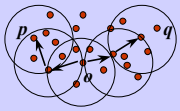
    $p_1 = q$, $p_n = p$ et

    p_{i+1} est directement densité accessible à partir de p_i



- Connexité
  - p est connecté à q resp. à Eps, $MinPts$ si il existe un point o :

    p et q accessibles à partir de o resp. à Eps et $MinPts$.

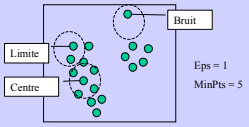


Principe

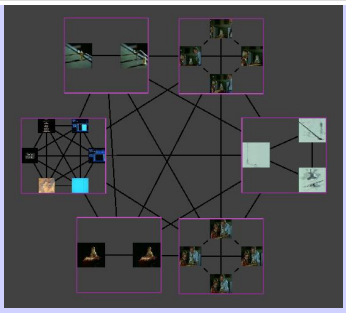
- Pour chaque point p, calculer son V(p).
  - Si |V(p)|> MinPts alors construire un cluster avec p comme représentant.
  - Pour chaque représentant p, chercher les représentants q qui lui sont accessibles.
  - Fusionner les clusters
- Le processus se termine quand les clusters ne changent plus

DBSCAN

- Martin Ester, Hans-Peter Kriegel, Jörg Sander, Spatial Data Mining: A Database Approach, Fifth Symposium on Large Spatial Databases,1997
- Algorithme
  Tant qu'il reste des points
  - Choisir ***p***
  - Récupérer tous les points accessibles à partir de ***p***
  - Si ***p*** est un centre, un cluster est formé.
  - si ***p*** est une limite, alors il n'y a pas de points accessibles de ***p***



DBSCAN : exemple



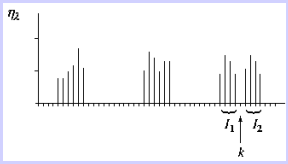
Benois-Pineau, Delest, Don, EI'04

DBSCAN

- Force
  - Complexité O(nlog(n))
  - Très efficace sur des données spaciales
- Faiblesse
  - Changement de paramètres ⇒Changement résultats

Par regroupement sur la distribution

- Auber, Chiricota, Delest, 2003

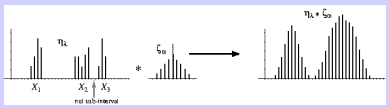


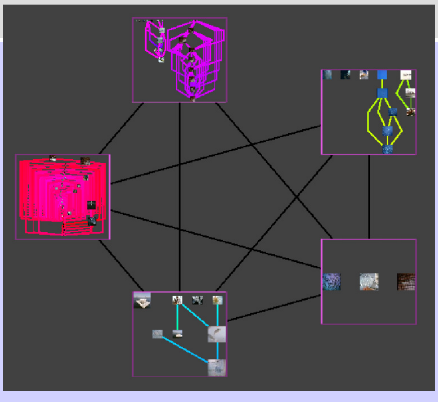
Histogramme Strahler

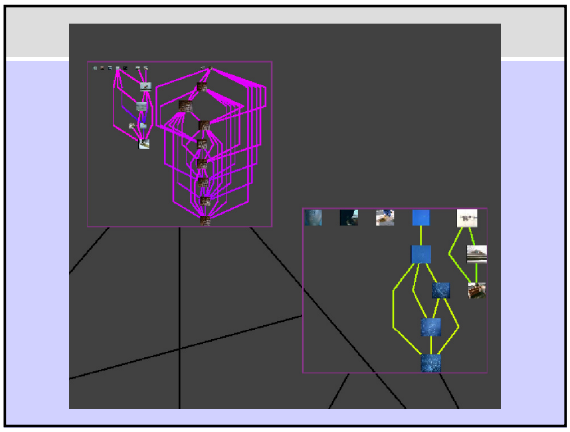
Par regroupement sur la distribution

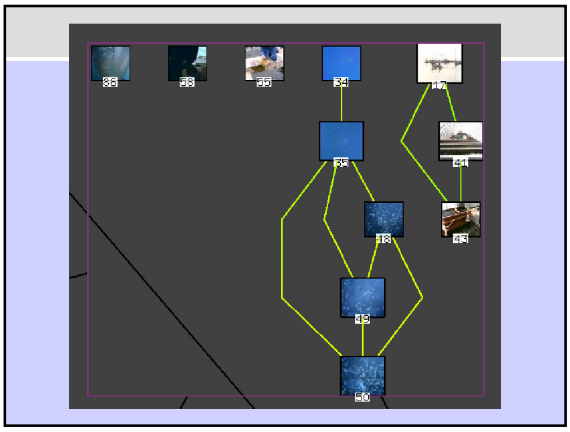
Convolution
- **Définie pour des fonctions quelconques**
- **Dans le cas des distributions liées aux métrique**
  histogramme f:N→N
  h: N→N

$$(f * h)(r) = \sum_{s=-\infty}^{+\infty} h(r-s)f(s).$$









Par convolution, sur la distribution

- Force
  - Interactivité
  - Très efficace sur des données de type graphe
- Faiblesse
  - Complexité $|supp(f)|^2(max(f)-min(f))^2$
  - Pas de contrôle du modèle