Data Mining: Clustering

Jef Wijsen

Université de Mons (UMONS)

Outline

Outline

La problématique

- Regrouper les données en plusieurs groupes (=clusters) de manière à ce que chaque groupe soit homogène et se distingue des autres groupes.
- Contrairement à la classification où on dispose d'un ensemble d'apprentissage avec des classes connues, les clusters sont inconnus a priori.

Mesures de similarité et de distance

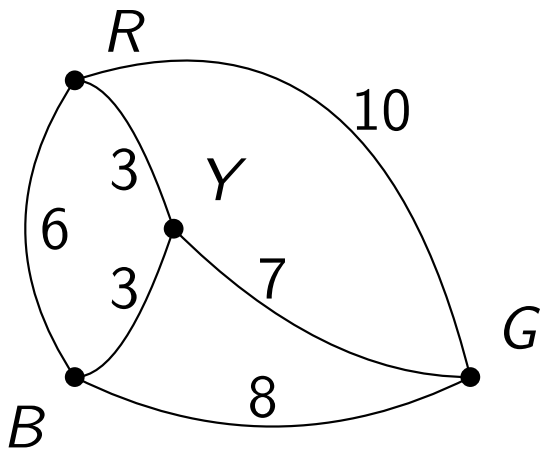
Soit $\mathbb{O}$ un ensemble d'objets. Une fonction $d : \mathbb{O} \times \mathbb{O} \longrightarrow \mathbb{R}$ définit une distance si elle satisfait les propriétés suivantes pour tout $x, y, z \in \mathbb{O}$:

$$
\begin{aligned}
&d(x, y) \geq 0 \\
&d(x, y) = 0 \text{ ssi } x = y \\
&d(x, y) = d(y, x) \\
&d(x, z) \leq d(x, y) + d(y, z)
\end{aligned}
$$

Exemple

$\mathbb{O} = \{R, B, G, Y\}$ et

d	R	B	G	Y
R	0	6	10	3
B	6	0	8	3
G	10	8	0	7
Y	3	3	7	0



Qu'est-ce que le centre d'un cluster ?

Soit $C \subseteq \mathbb{O}$. Qu'est-ce que le centre de C ?
Au moins deux définitions sont raisonnables :

1. Un objet m du cluster (i.e. $m \in C$) pour lequel $\sum_{x \in C} d(m, x)^2$ est minimal (on appelle m aussi medoïde ou médiane).
2. Un objet $c \in \mathbb{O}$, pas nécessairement dans C, pour lequel $\sum_{x \in C} d(c, x)^2$ est minimal (on appelle c aussi centroïde ou moyenne).

Exemple

Soit $C = \{R, B, G\}$.

$$\begin{aligned} d(R,R)^2 + d(R,B)^2 + d(R,G)^2 &= 136 \\ d(B,R)^2 + d(B,B)^2 + d(B,G)^2 &= 100 \\ d(G,R)^2 + d(G,B)^2 + d(G,G)^2 &= 164 \end{aligned}$$

B est donc l'objet le plus central de C. Notez néanmoins :

$$d(Y,R)^2 + d(Y,B)^2 + d(Y,G)^2 = 67$$

Exemples en $\mathbb{R}^n$

Soient $\vec{x} = (x_1, x_2, \ldots x_n)$ et $\vec{y} = (y_1, y_2, \ldots y_n)$ deux points en $\mathbb{R}^n$.

Distance euclidienne : $d_{Eucl}(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

Distance Manhattan ou "city block" :

$$d_{Manh}(\vec{x}, \vec{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

Distance de Minkowski : Soit $q \in \mathbb{N}$, $q > 0$.

$$d_{Mink(q)}(\vec{x}, \vec{y}) = \sqrt[q]{\sum_{i=1}^{n} |x_i - y_i|^q}$$

Notez : $d_{Eucl}(\vec{x}, \vec{y}) = d_{Mink(2)}(\vec{x}, \vec{y})$ et $d_{Manh}(\vec{x}, \vec{y}) = d_{Mink(1)}(\vec{x}, \vec{y})$

Qu'est-ce qu'un cluster ?

Partitionner un ensemble $S \subseteq \mathbb{O}$ en plusieurs clusters.
Plusieurs charactérisations du concept "cluster" sont raisonnables. Par ex.

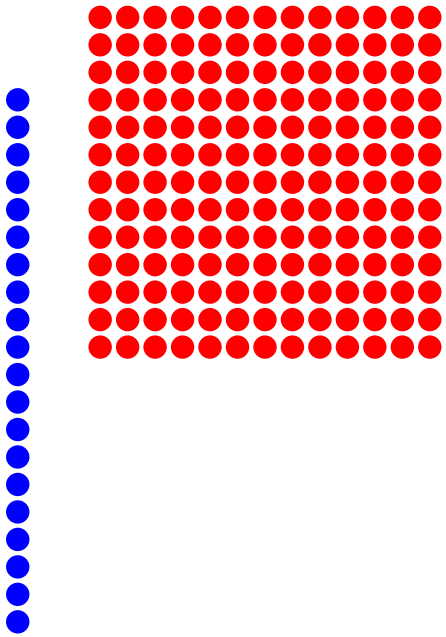
Centrisme Chaque objet est plus proche du centre de son propre cluster que de tout autre centre. Il suffit donc de spécifier les centres pour connaître les clusters.

Séparatisme Chaque objet est plus proche de tout objet de son propre cluster que de n'importe quel objet d'un autre cluster.

Atteignabilité Chaque objet appartient au même cluster que son voisin le plus proche.

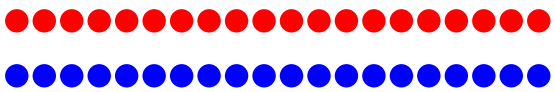
Exemples en $\mathbb{R}^2$

Le clustering suivant satisfait "atteignabilité" mais pas "séparatisme", ni "centrisme".



Exemples en $\mathbb{R}^2$

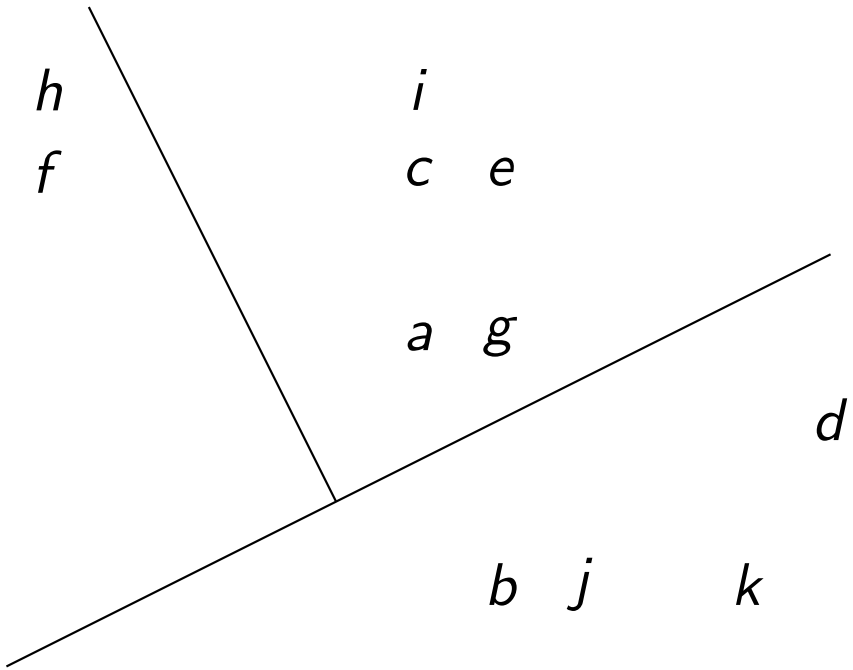
Le clustering suivant satisfait "centrisme" et "atteignabilité" mais pas "séparatisme".



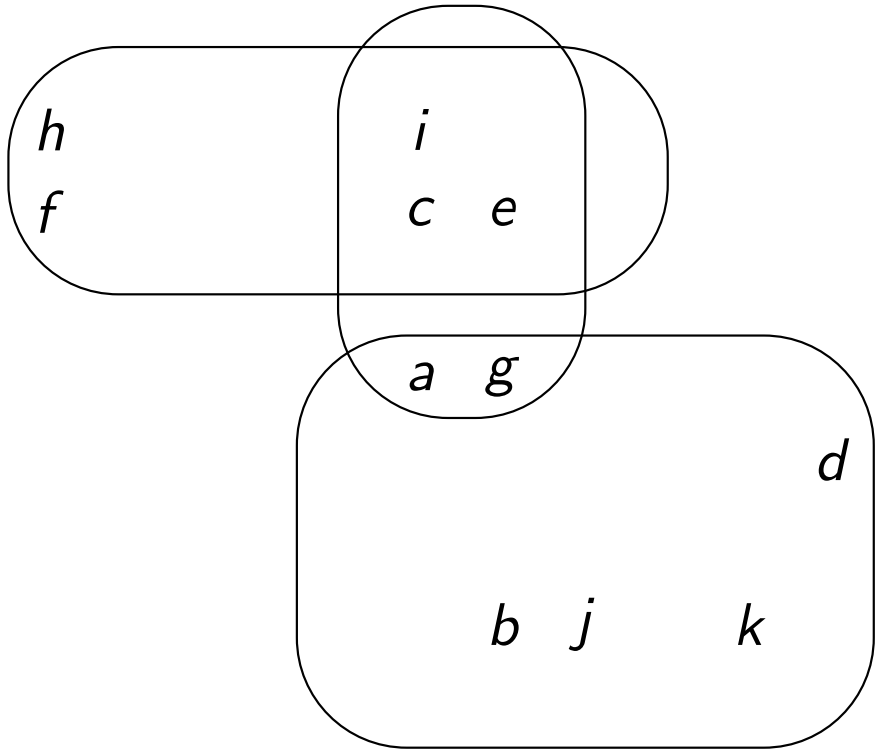
Le clustering suivant satisfait "centrisme" mais pas "atteignabilité".

Outline

Les clusters disjoints



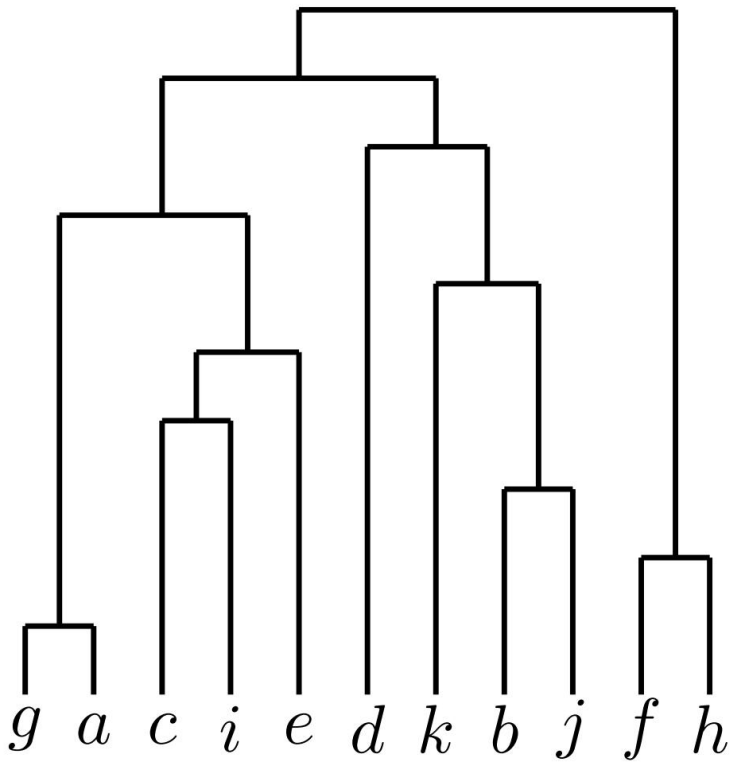
Les clusters pas forcément disjoints



Les clusters probabilistes

	1	2	3
a	0.3	0.4	0.3
b	0.1	0.2	0.7
c	0.4	0.5	0.1
$\vdots$		$\vdots$	

Dendrogram : une hiérarchie de clusters



Clustering en Weka

Plusieurs "clusterers", entre autres :

- `clusterers.SimpleKMeans`
- `clusterers.FarthestFirst`
- `clusterers.EM` (expectation-maximization)
- `clusterers.HierarchicalClusterer`
- `clusterers.DBScan`

Quatre "cluster modes" :

Use training set : Cluster the same set that the clusterer is trained on.

Supplied test set : Cluster a user-specified dataset.

Percentage split : Train on a percentage of the data and cluster the remainder.

Classes to clusters evaluation : Evaluate clusters with respect to a class.

Outline

Le clustering vu comme un problème d'optimisation

On souhaite partitionner un ensemble S en $k \geq 2$ clusters.
Soient $C_1, C_2, \ldots, C_k$ des clusters avec centres $c_1, c_2, \ldots, c_k$ respectivement. Définissons la dispersion intra-cluster comme :

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} d(c_i, x)^2$$

(SSE : Sum of the Squared Error)
Le but est de trouver un clustering avec une dispersion intra-cluster minimale.

Principe de kMeans clustering

Partitionner un ensemble S en k clusters.

1. Choisir les moyennes $m_1, m_2, \ldots, m_k$.
2. Attribuer tout objet de S à la moyenne la plus proche. Soient $C_1, C_2, \ldots, C_k$ les ensembles d'objets attribués respectivement à $m_1, m_2, \ldots, m_k$.
3. Ajuster les moyennes :

$$\begin{aligned} m_1 &:= \text{la moyenne de } C_1 \\ m_2 &:= \text{la moyenne de } C_2 \\ &\ldots \\ m_k &:= \text{la moyenne de } C_k \end{aligned}$$

4. Goto 2.

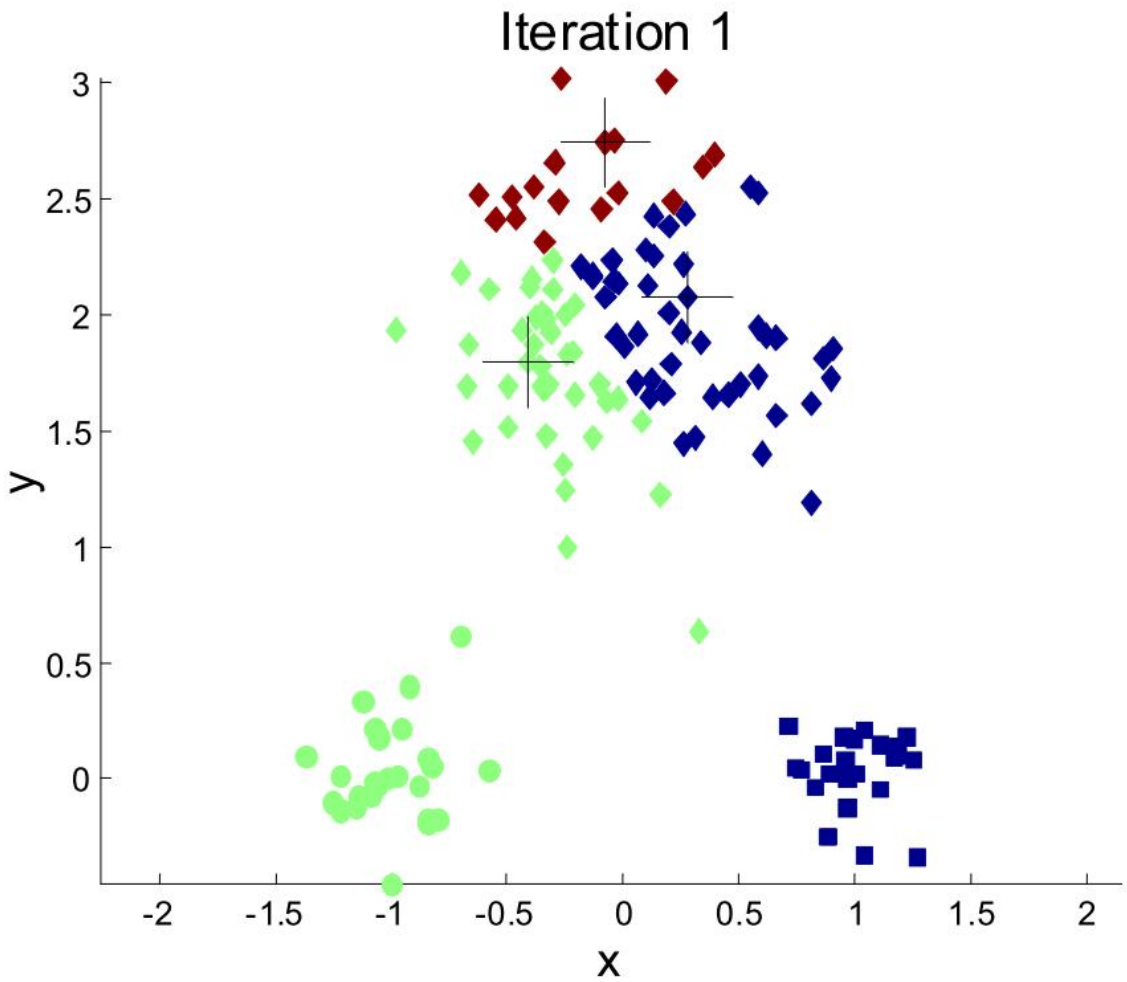
kMeans : algorithme

Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.

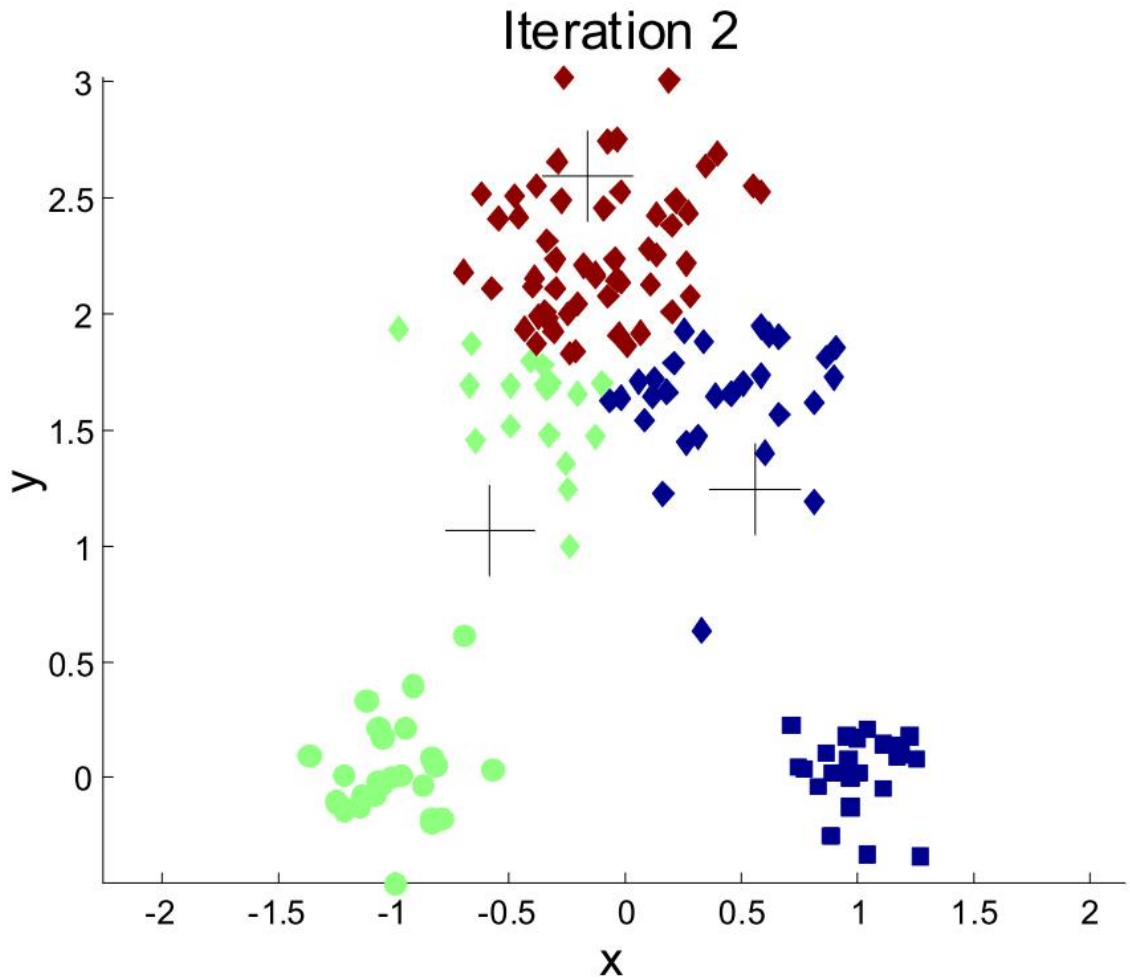
Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

kMeans : Example



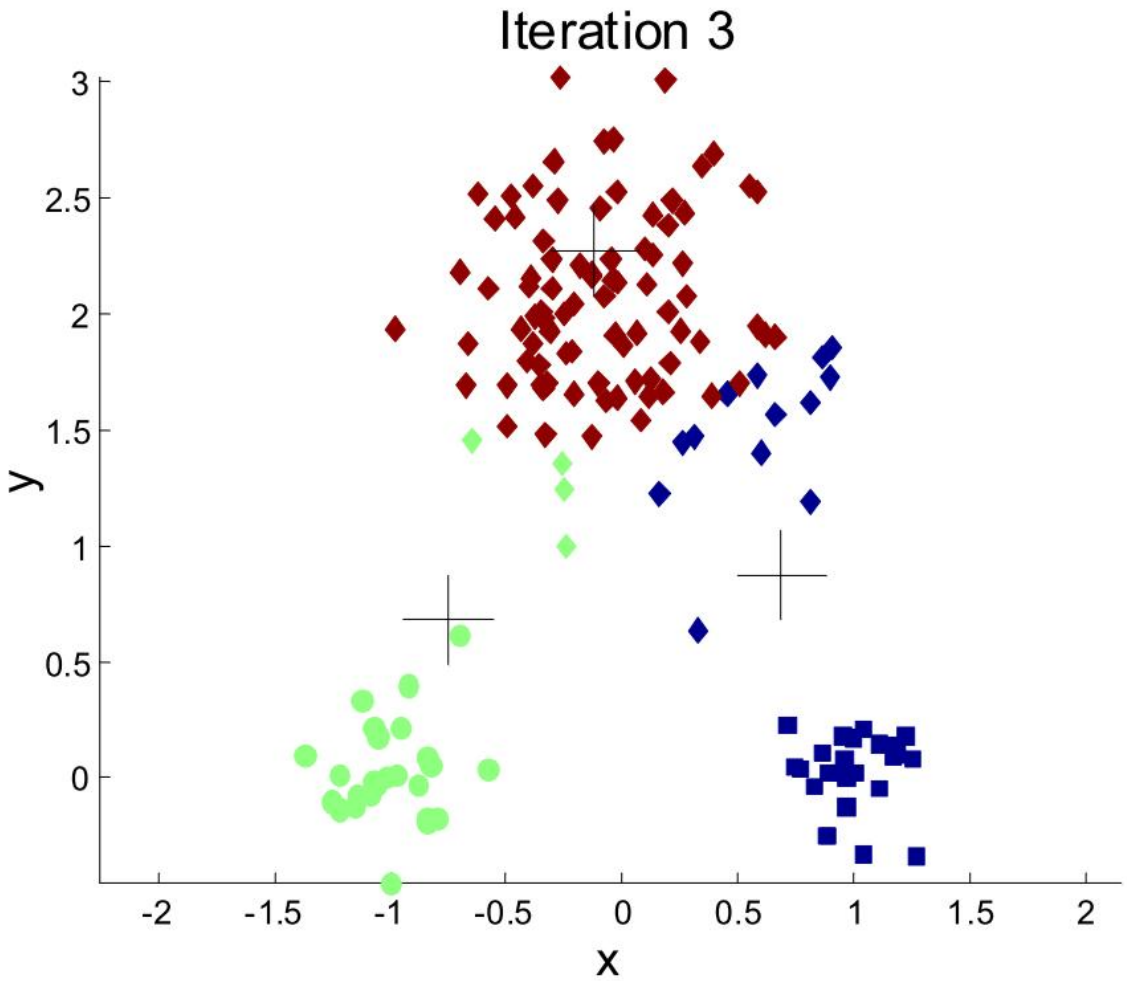
Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

kMeans : Example



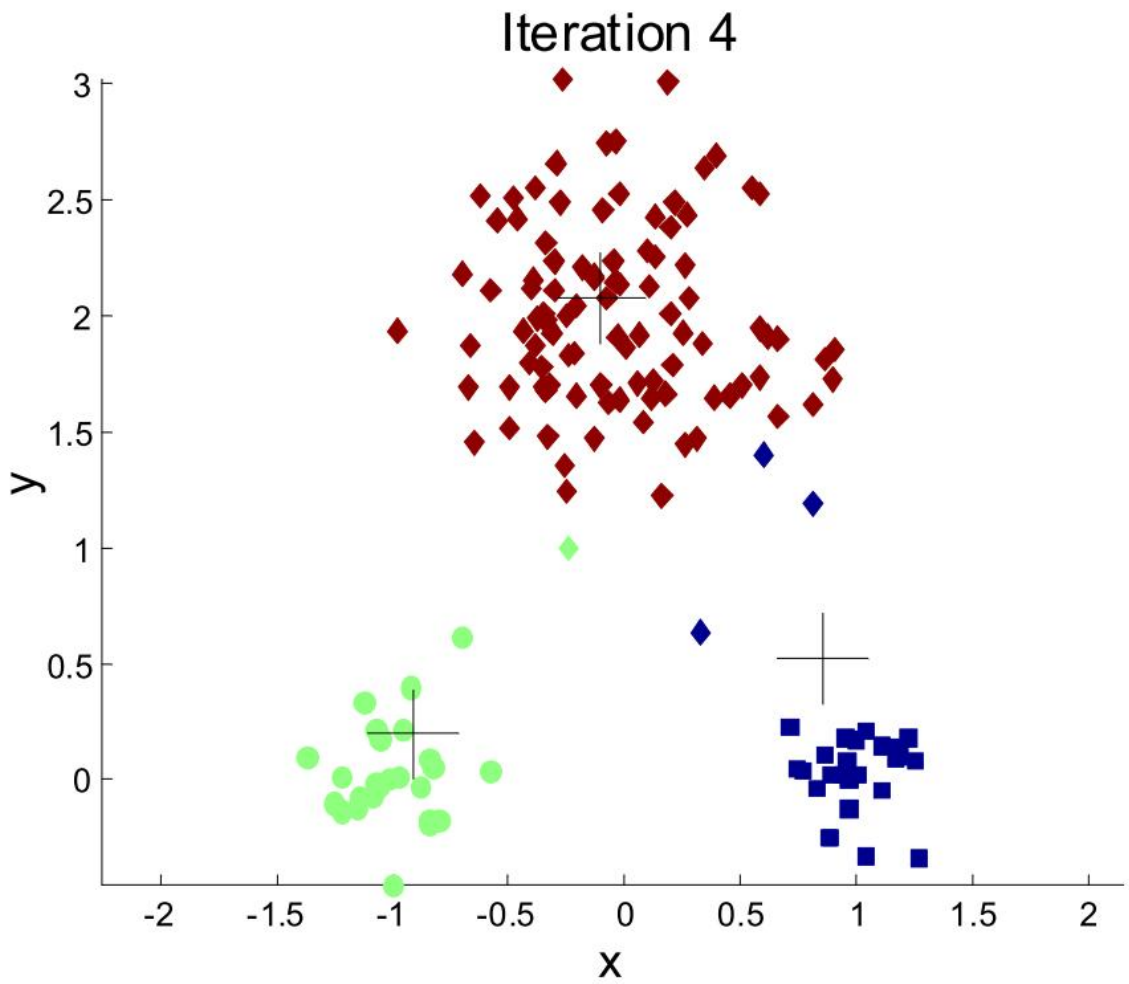
Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

kMeans : Example



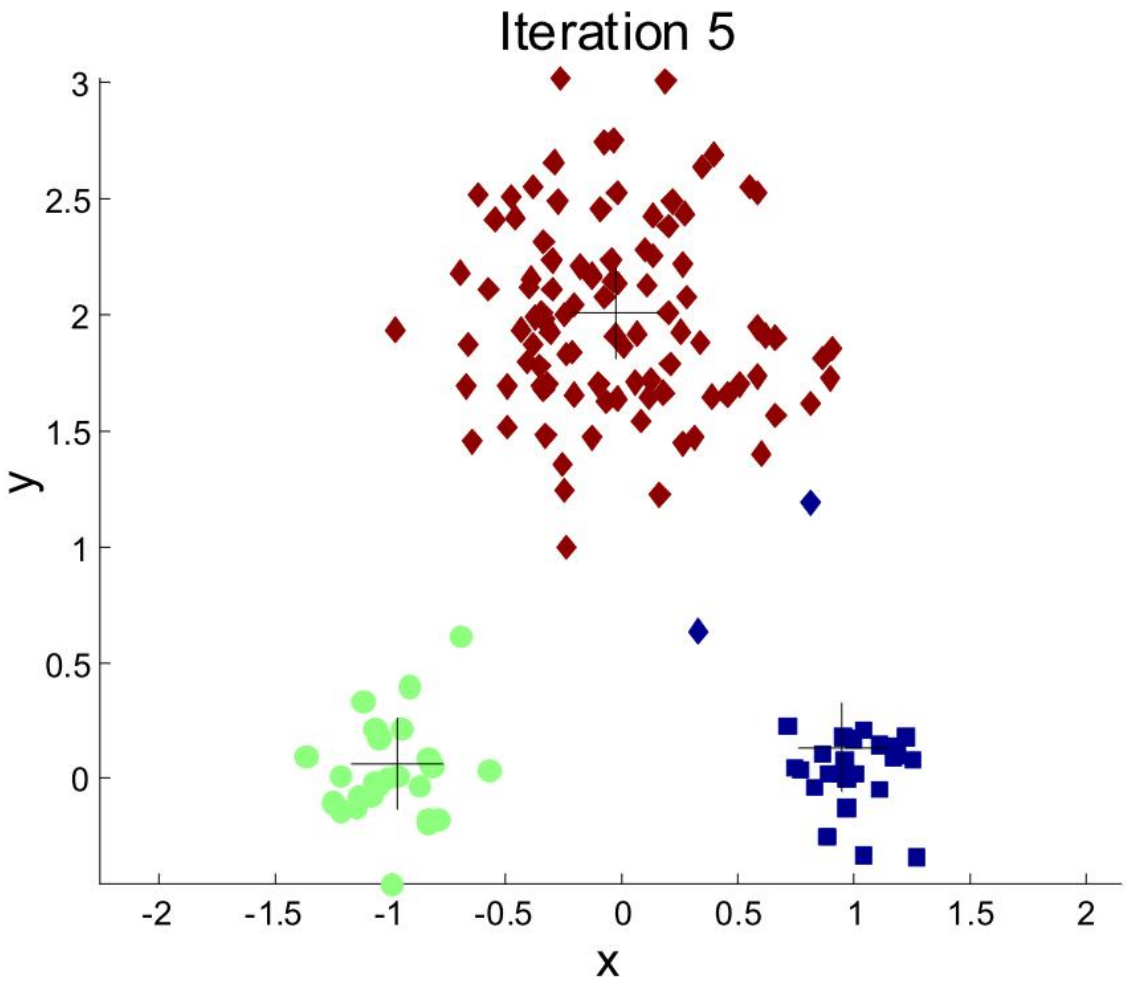
Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

kMeans : Example



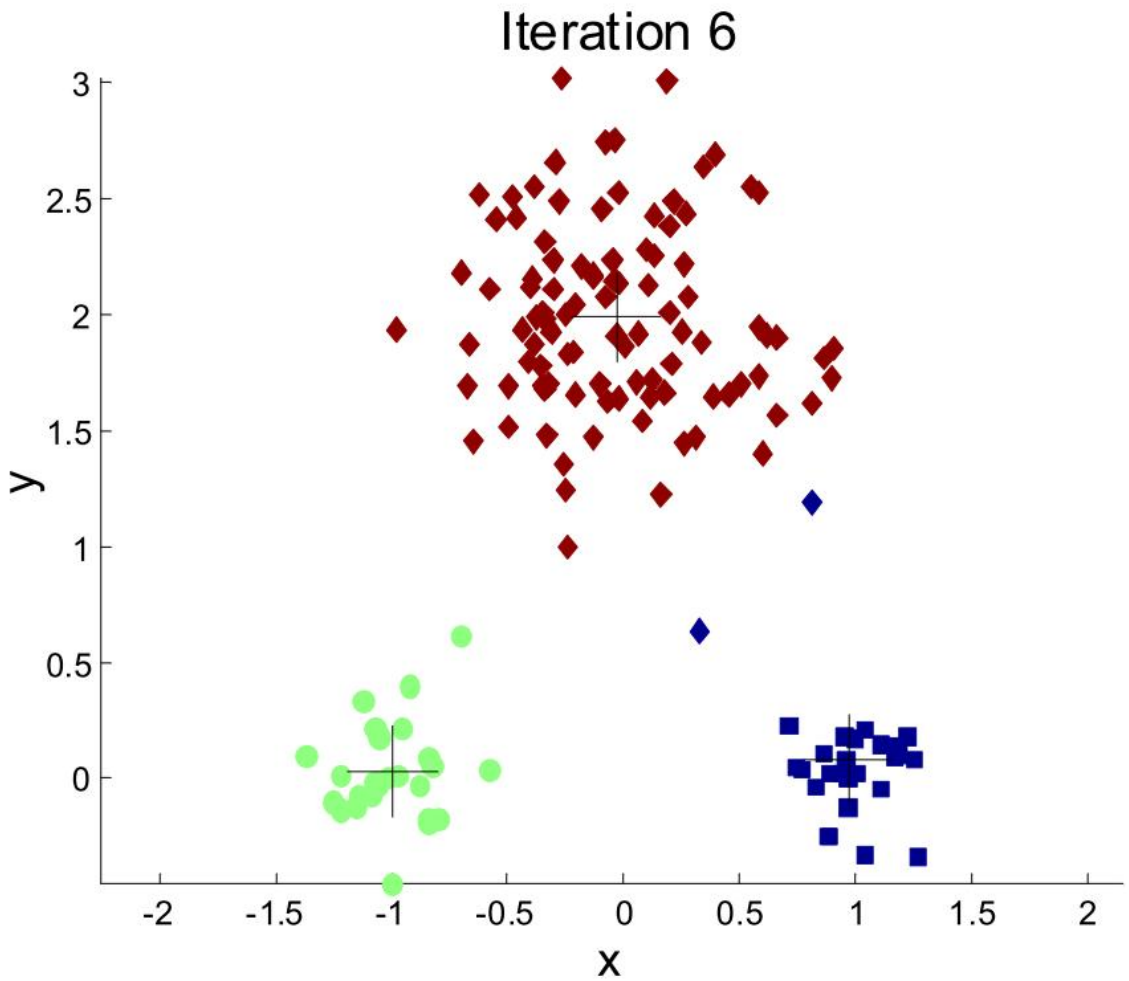
Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

kMeans : Example



Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

kMeans : Example

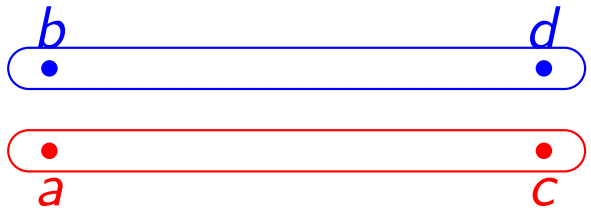


Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

Discussion

- Comment déterminer k ?
- kMeans garantit "centrisme" (voir transparent 8), mais pas "atteignabilité".

Résultat si on démarre avec a, b :



Une meilleure solution est $\{\{a, b\}, \{c, d\}\}$.

kMedoids clustering

Partitionner un ensemble S en k clusters.

1. Choisir les medoïdes $m_1, m_2, \ldots, m_k \in S$.
2. Chercher $m_j \in \{m_1, m_2, \ldots, m_k\}$ et $p \in S \setminus \{m_1, m_2, \ldots, m_k\}$ tel que remplacer m_j par p améliore le clustering.
3. Goto 2.

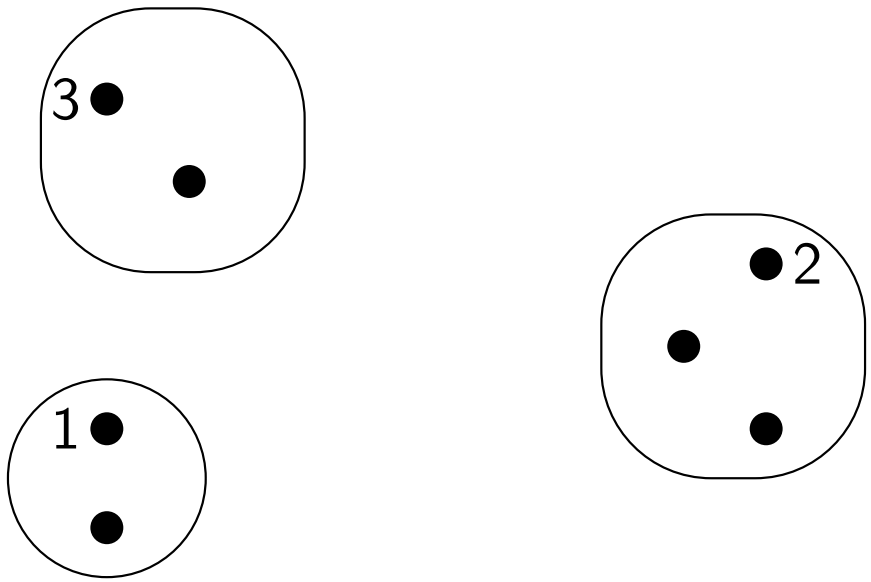
Farthest First

Pour un objet o et en ensemble S d'objets, définissons $d(o, S) := \min\{d(o, p) \mid p \in S\}$.

- Choisir un point m_1.
- Choisir pour m_2 le point le plus éloigné de m_1.
- Choisir pour m_3 le point le plus éloigné de $\{m_1, m_2\}$.
- Choisir pour m_4 le point le plus éloigné de $\{m_1, m_2, m_3\}$.
- ...
- Choisir pour m_k le point le plus éloigné de $\{m_1, m_2, \ldots, m_{k-1}\}$.

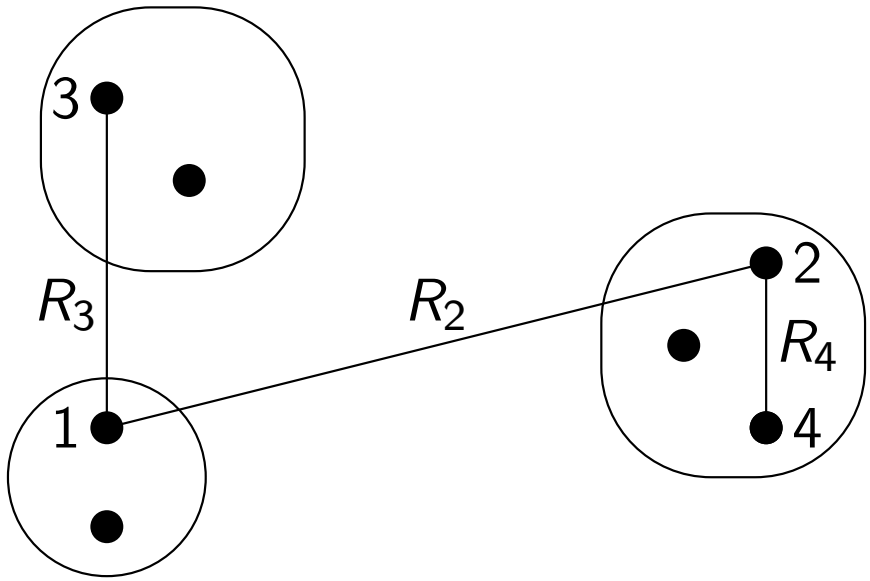
Farthest First $k = 3$

- Le premier point est choisi au hasard.

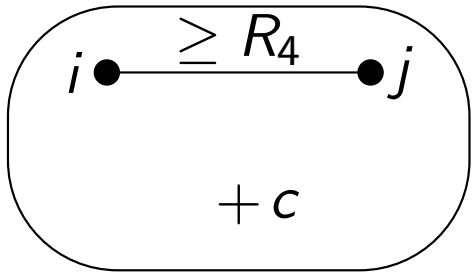


Farthest First $k = 3$: garantie de performance

- Le rayon de chaque cluster est $\leq R_4$.



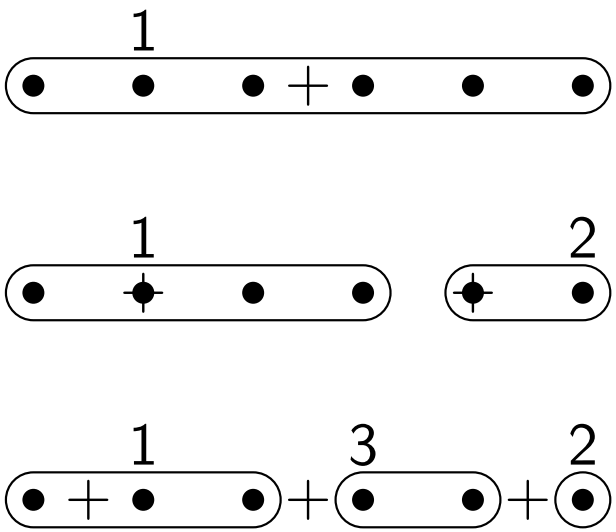
- Tout 3-clustering contiendra un cluster regroupant deux points parmi $\{1, 2, 3, 4\}$ (pigeon hole principle). Soit c le centre de ce cluster.
- $i, j \in \{1, 2, 3, 4\}$ et $i \neq j$.



Soit r le rayon de ce cluster. Évidemment, $r \geq d(c, i)$ et $r \geq d(c, j)$. Puisque $d(i, c) + d(c, j) \geq d(i, j) \geq R_4$, $\boxed{r \geq R_4/2}$.

- Tout 3-clustering contiendra donc un cluster avec rayon $\geq R_4/2$.
- Les clusters trouvés pas Farthest First ont tous un rayon qui est au pire deux fois ce rayon minimal.

Exemple



	Rayon maximal	
	Farthest First	Optimal(+)
	4	2.5
	2	1
	1	0.5

Exercice

Supposons 1000 points en 2D distribués de manière uniforme entre $(0, 0)$ et $(100, 100)$. Supposons $m_1 = (50, 75)$.

1. Comment l'algorithme `FarthestFirst` va-t-il partitionner ce jeu de données en deux clusters ?
2. Comment l'algorithme `SimpleKMeans` va-t-il partitionner ce jeu de données en deux clusters à partir des deux centres de `FarthestFirst` ?

Générer des nombres aléatoires

Pour générer des nombres aléatoires en MS Excell :

1. suivre *Outils* > *Macros complémentaires* et cocher *Utilitaire d'analyse*;
2. puis utiliser *Outils* > *Utilitaire d'analyse* > *Génération de nombres aléatoires*.

Note pratique

Voir `http://www.xycoon.com/nor_random.htm`
Approximation pour générer des points selon une distribution normale avec $\mu = 0$ et $\sigma = 1$:

$$\sum_{i=1}^{12} U_i - 6$$

avec U_i un nombre aléatoire entre 0 et 1.
Il est clair que le resultat se trouve entre -6 et $+6$, avec une moyenne de 0.

Outline

Distribution normale

Distribution normale avec moyenne μ et écart type σ :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$$

La probabilité qu'une valeur se trouve dans l'interval $[a, b]$:

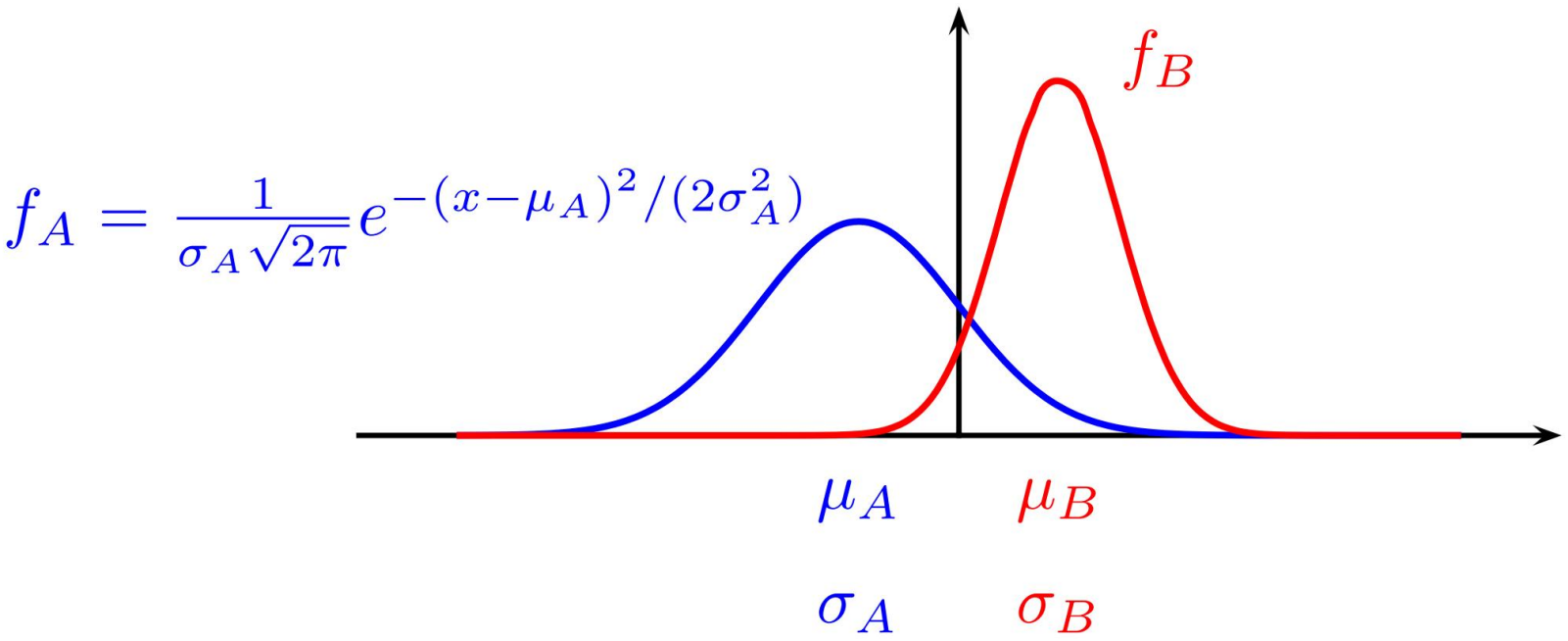
$$\int_a^b f(x)dx$$

On a $\int_{-\infty}^{+\infty} f(x)dx = 1$ $\int_{\mu-\sigma}^{\mu+\sigma} f(x)dx \approx 0.68$ $\int_{\mu-2\sigma}^{\mu+2\sigma} f(x)dx \approx 0.95$
Pour petit ε,

$$\int_{a-\varepsilon/2}^{a+\varepsilon/2} f(x)dx \approx \varepsilon \cdot f(a)$$

Génération des clusters

Soit $p_A + p_B = 1$. Générer n valeurs de manière suivante :
(i) Générer $p_A \cdot n$ valeurs selon la distribution f_A;
(ii) Générer $p_B \cdot n$, valeurs selon la distribution f_B.



Cette génération dépend de cinq paramètres : $p_A, \mu_A, \sigma_A, \mu_B, \sigma_B$.

Retrouver les clusters

Soit S un ensemble avec n valeurs.

- On suppose que les valeurs on été générées comme expliqué ci-dessus.
- On souhaite retrouver les cinq paramètres.
- Plus précisément, on souhaite trouver les cinq paramètres qui maximisent le log-likelihood :

$$L = \sum_{x \in S} \log(\underbrace{p_A \times f_A(x) + p_B \times f_B(x)}_{\sim \text{ probabilité de générer } x})$$

Estimer les cinq paramètres

Supposons qu'on observe deux clusters A et B :

- $A = \{x_1, x_2, \ldots, x_{m_A}\}$
- $B = \{y_1, y_2, \ldots, y_{m_B}\}$

Estimer les cinq paramètres $\mu_A, \sigma_A, \mu_B, \sigma_B, p_A$ est facile :

$$\begin{aligned} \textit{sample mean} \quad \mu_A &= \frac{1}{m_A} \sum_{i=1}^{m_A} x_i \\ \textit{sample variance} \quad \sigma_A^2 &= \frac{1}{m_A - 1} \sum_{i=1}^{m_A} (x_i - \mu_A)^2 \\ p_A &= \frac{m_A}{m_A + m_B} \end{aligned}$$

Idem pour B.

EM-clustering

On demande de diviser l'ensemble S avec n valeurs en deux clusters A et B. On ne connaît aucun des cinq paramètres.

1. Choisir deux clusters A et $B = S \setminus A$ de départ.
2. Calculer $\mu_A, \sigma_A, \mu_B, \sigma_B, p_A$.
3. Expectation : Calculer $Pr(A \mid x)$ et $Pr(B \mid x)$ pour tout $x \in S$.
4. Maximization :

$$\mu_A := \frac{\sum_{x \in S} Pr(A|x) \cdot x}{\sum_{x \in S} Pr(A|x)} \quad \sigma_A^2 := \frac{\sum_{x \in S} Pr(A|x)(x-\mu_A)^2}{\sum_{x \in S} Pr(A|x)} \quad p_A := \frac{\sum_{x \in S} Pr(A|x)}{n}$$

   Idem pour B.

5. Goto 3.

Une fois les cinq paramètres connus, la probabilité $Pr(A \mid x)$ qu'une valeur x appartient à cluster A est calculé comme suit :

$$\begin{aligned} Pr(A \mid x) &= \frac{Pr(x \mid A) \times Pr(A)}{Pr(x)} \sim \frac{f_A(x) \times p_A}{Pr(x)} \\ Pr(B \mid x) &= \frac{Pr(x \mid B) \times Pr(B)}{Pr(x)} \end{aligned}$$

On sait calculer $Pr(A \mid x)$ à partir de :

$$\begin{aligned} \frac{Pr(A \mid x)}{Pr(B \mid x)} &= \frac{f_A(x) \times p_A}{f_B(x) \times p_B} \\ Pr(A \mid x) + Pr(B \mid x) &= 1 \end{aligned}$$

Dès lors :

$$Pr(A \mid x) = \frac{f_A(x) \times p_A}{f_A(x) \times p_A + f_B(x) \times p_B}$$

De même manière :

$$Pr(B \mid x) = \frac{f_B(x) \times p_B}{f_A(x) \times p_A + f_B(x) \times p_B}$$

Comme chez Naive Bayes, les dénominateurs sont les mêmes et peuvent être “oubliés” dès le début.

Exercice

- Utiliser MS Excell pour génerer deux clusters en 2D.
- Choisir un cluster avec centre $(0, 0)$ et une distribution normale $N(0, 1)$ selon les deux axes.
- Choisir un cluster avec centre $(2, 2)$ et une distribution normale $N(2, 1)$ selon les deux axes.
- Chaque cluster contient > 100 points.
- Retrouver les clusters à l'aide de EM.

Outline

Principe de bottom-up hierarchical clustering

1. Au départ, chaque objet constitue un cluster avec un seul élément.
2. A plusieurs reprises, joindre les clusters les plus proches. (Comment mesurer la distance entre deux clusters ?)

Bottom-up hierarchical clustering : algorithme

Algorithm 8.3 Basic agglomerative hierarchical clustering algorithm.

1: Compute the proximity matrix, if necessary.
2: **repeat**
3: Merge the closest two clusters.
4: Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
5: **until** Only one cluster remains.

Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

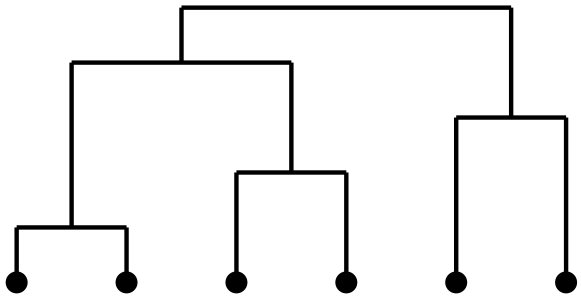
Single-link et complete-link

Single-link The distance between C and C' is defined as $\min\{d(o, o') \mid o \in C, o' \in C'\}$.

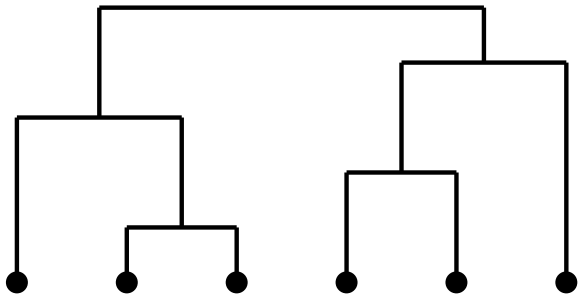
Complete-link The distance between C and C' is defined as $\max\{d(o, o') \mid o \in C, o' \in C'\}$.

Exercice Appliquer sur $S = \{0, 2, 5, 9\}$.

Hiéarchie vs performance



optimal 3-clustering
mauvais 2-clustering



optimal 2-clustering
mauvais 3-clustering

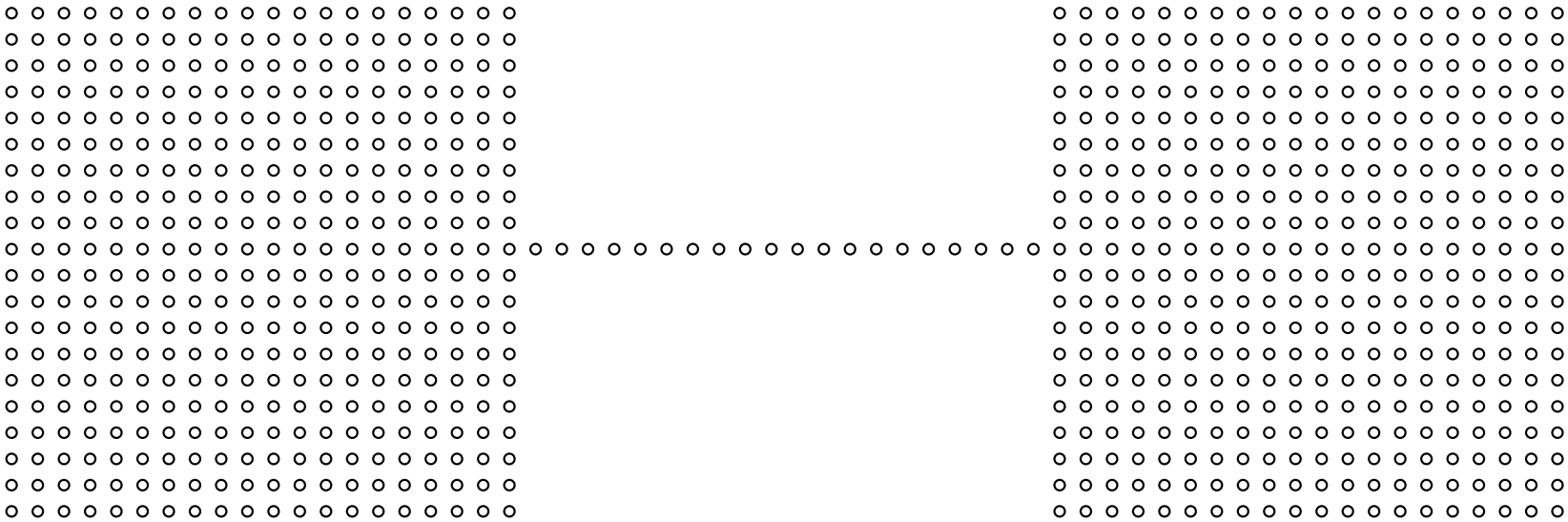
Outline

Principe

Fixer un seuil ε.

1. Si $d(\vec{x}, \vec{y}) \leq \varepsilon$, alors $\vec{x}$ et $\vec{y}$ appartiennent au même cluster.
2. Si $\vec{x}$ et $\vec{y}$ appartiennent au même cluster et $d(\vec{y}, \vec{z}) \leq \varepsilon$, alors $\vec{x}$ et $\vec{z}$ appartiennent au même cluster.
3. Deux points appartiennent au même cluster seulement si les règles 1 et 2 l'imposent.

Problème des "ponts"



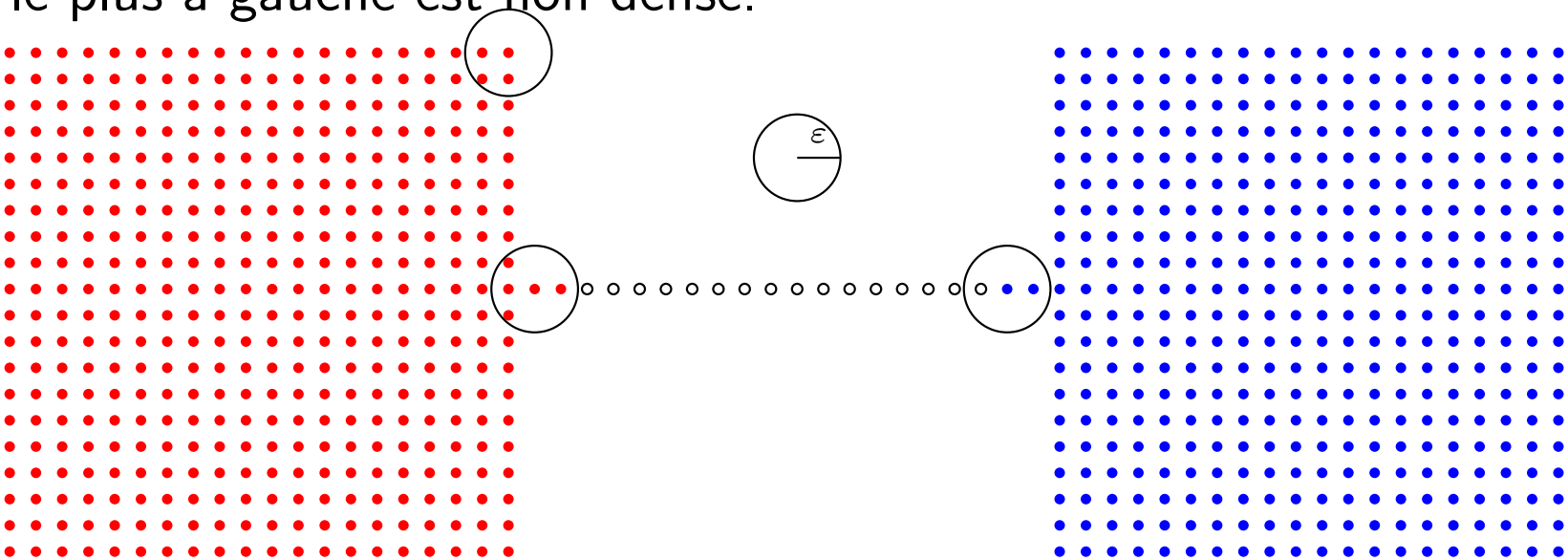
Principe de DBSCAN

Fixer un seuil ε. Un point est $\vec{x}$ est dense s'il y a au moins δ points $\vec{y}$ qui satisfont $d(\vec{x}, \vec{y}) \leq \varepsilon$.

1. Si $d(\vec{x}, \vec{y}) \leq \varepsilon$ et $\vec{x}$ est dense, alors $\vec{x}$ et $\vec{y}$ appartiennent au même cluster.
2. Si $\vec{x}$ et $\vec{y}$ appartiennent au même cluster et $d(\vec{y}, \vec{z}) \leq \varepsilon$ et $\vec{y}$ est dense, alors $\vec{x}$ et $\vec{z}$ appartiennent au même cluster.
3. Deux points appartiennent au même cluster seulement si les règles 1 et 2 l'imposent.

Problème des "ponts"

Supposons que la distance entre deux points voisins est plus petit que ε. Prenons $\delta = 4$. Les points noirs sont non-denses et n'appartiennent à aucun cluster! Le point rouge le plus à droite est non-dense. Le point bleu le plus à gauche est non-dense.



Core points, border points, and noise points

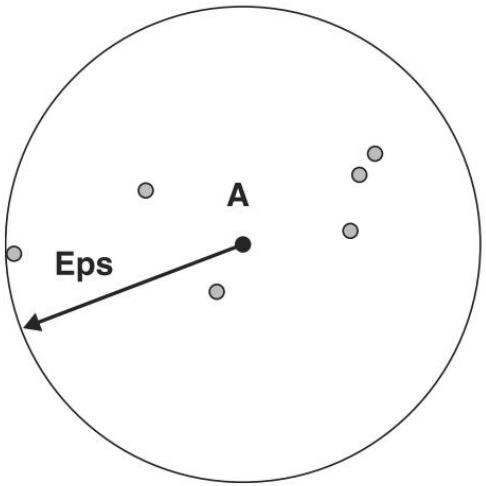


Figure 8.20. Center-based density.

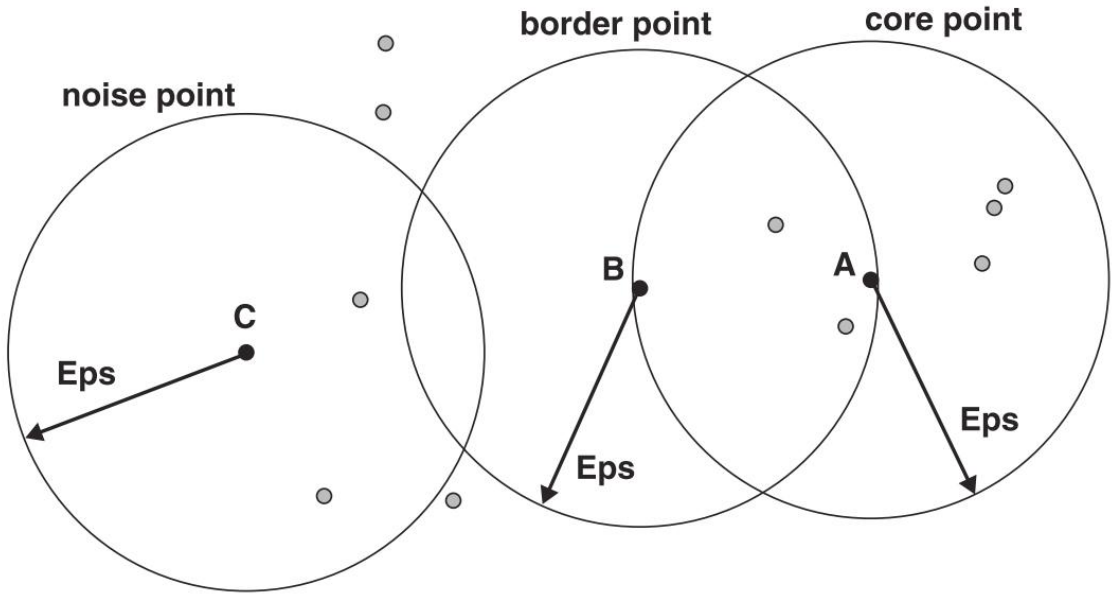


Figure 8.21. Core, border, and noise points.

Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

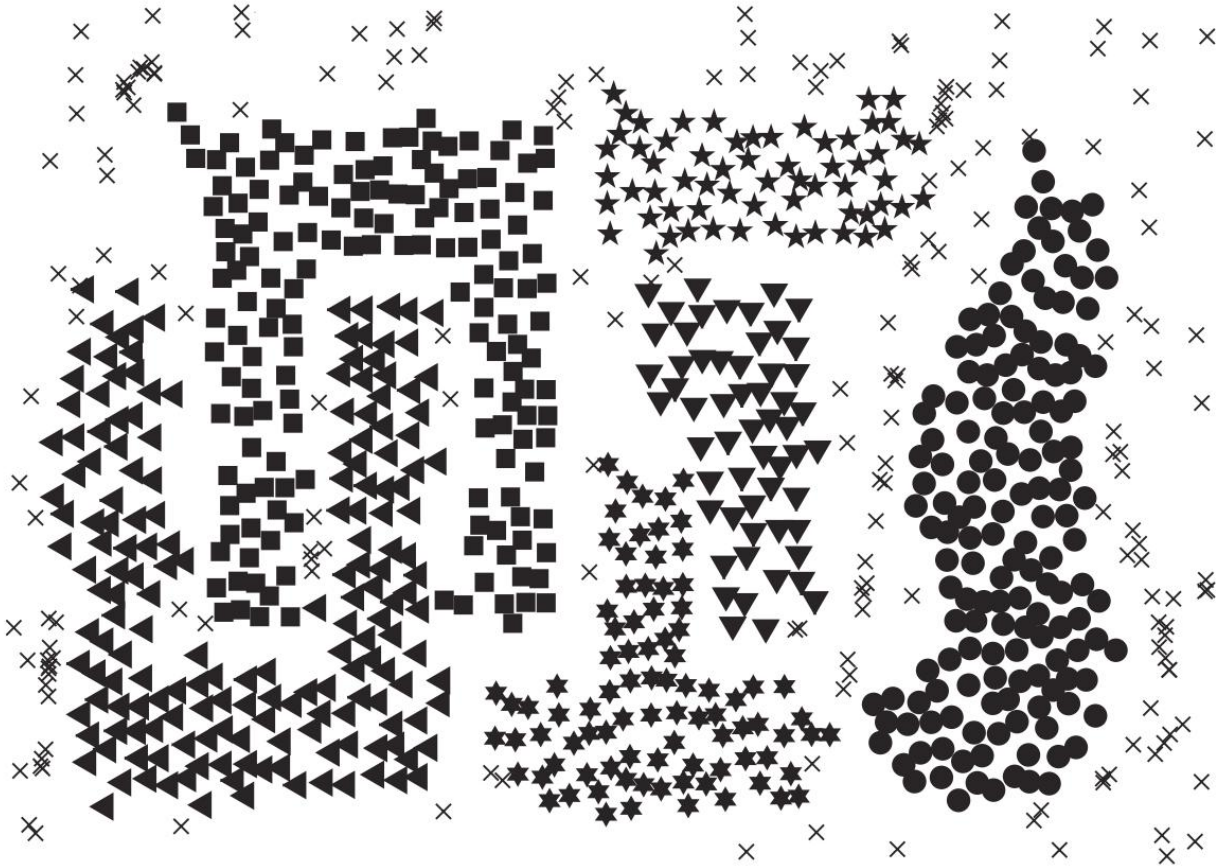
DBSCAN : algorithme

Algorithm 8.4 DBSCAN algorithm.

1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points that are within *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

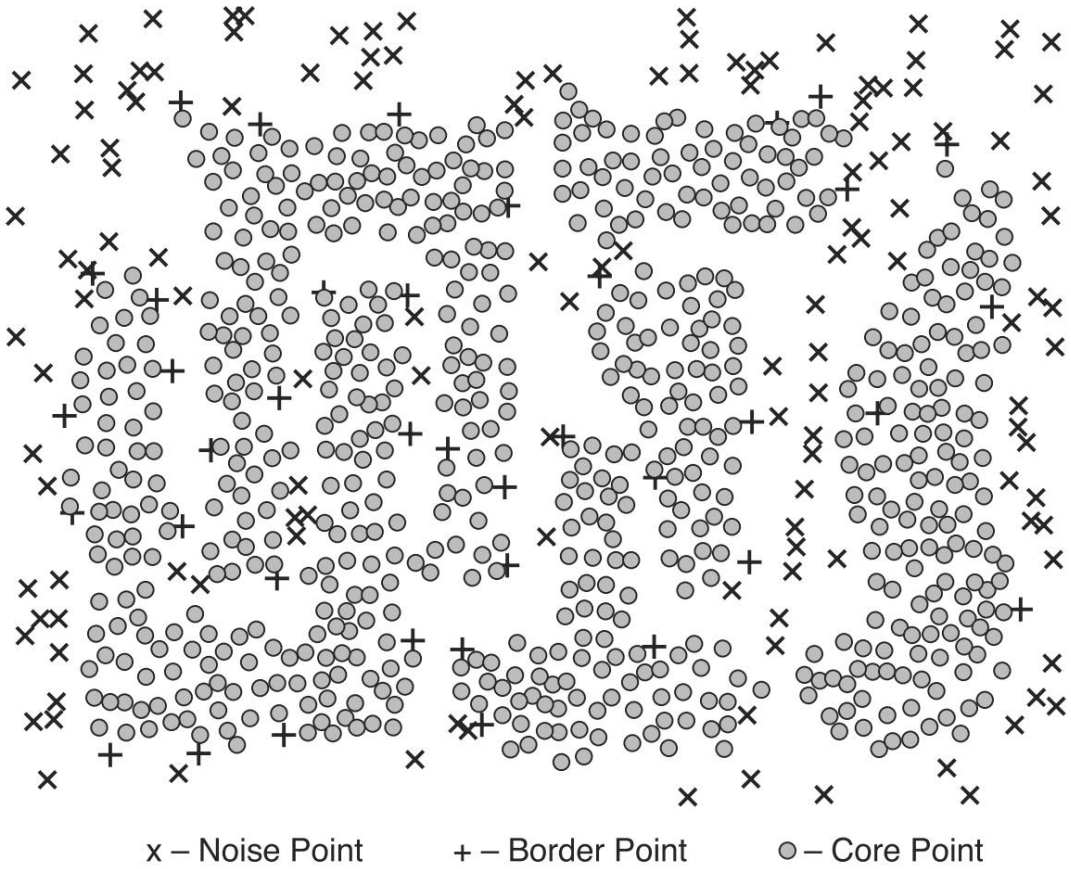
DBSCAN : Example



(a) Clusters found by DBSCAN.

Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

DBSCAN : Example



(b) Core, border, and noise points.

Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

Exercice DBSCAN

- Exécuter DBSCAN sur `dbscan.arff` avec différentes valeurs pour epsilon et minPoints.

Outline

L'encodage

- Partitionner $S = \{o_1, \ldots, o_n\}$ en k clusters $C_1, C_2, \ldots, C_k$. Soit $K = \{1, 2, \ldots, k\}$. Un élément $\langle i_1, i_2, \ldots, i_n \rangle \in K^n$ représente le clustering où $o_1 \in C_{i_1}, o_2 \in C_{i_2}, \ldots, o_n \in C_{i_n}$.
- Notez que chaque permutation de $\langle 1, 2, \ldots, k \rangle$ donne le même clustering!
- Chaque élément de K^n est un individu (ou chromosome). L'aptitude d'un individu pourrait être l'inverse de la dispersion intra-cluster.
- Une population est un ensemble d'individus.

Les opérateurs

- Le cross-over de $\langle i_1, \ldots, i_n \rangle$ et $\langle j_1, \ldots, j_n \rangle$ entre les positions l et $l+1$ donne $\langle i_1, \ldots, i_l, j_{l+1}, \ldots, j_n \rangle$ et $\langle j_1, \ldots, j_l, i_{l+1}, \ldots, i_n \rangle$.
- La mutation change une valeur dans un individu de manière arbitraire.
- Le principe de survival of the fittest crée une nouvelle population avec le même nombre d'individus. La probabilité d'un individu de se retrouver (une ou plusieurs fois) dans la prochaine génération, est proportionnel à son aptitude (roue de la fortune).

L'algorithme

fixer la population de départ

loop

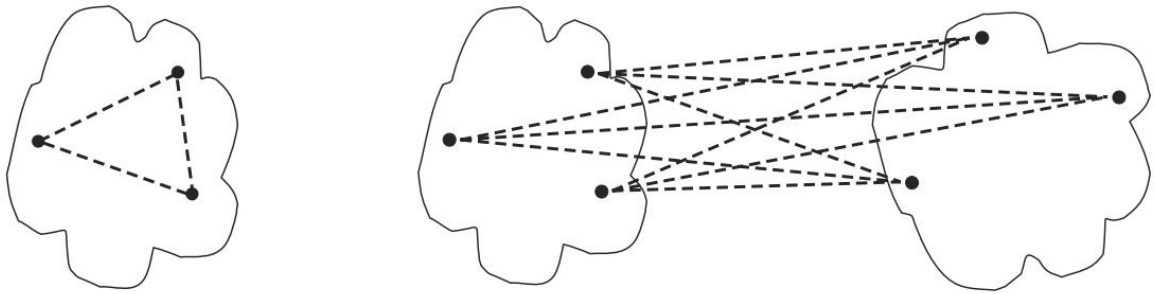
appliquer cross-over et mutation

choisir les individus de la prochaine génération (survival of the fittest)

end-loop

Outline

Cohesion and Separation

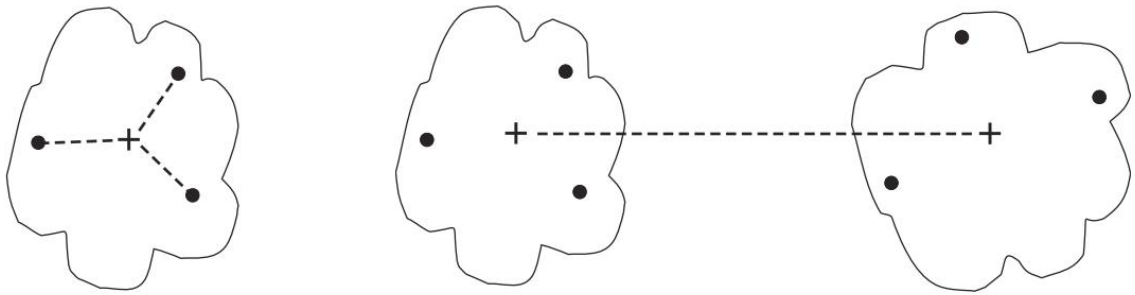


(a) Cohesion. (b) Separation.

Figure 8.27. Graph-based view of cluster cohesion and separation.

Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

Cohesion and Separation w.r.t. Prototypes

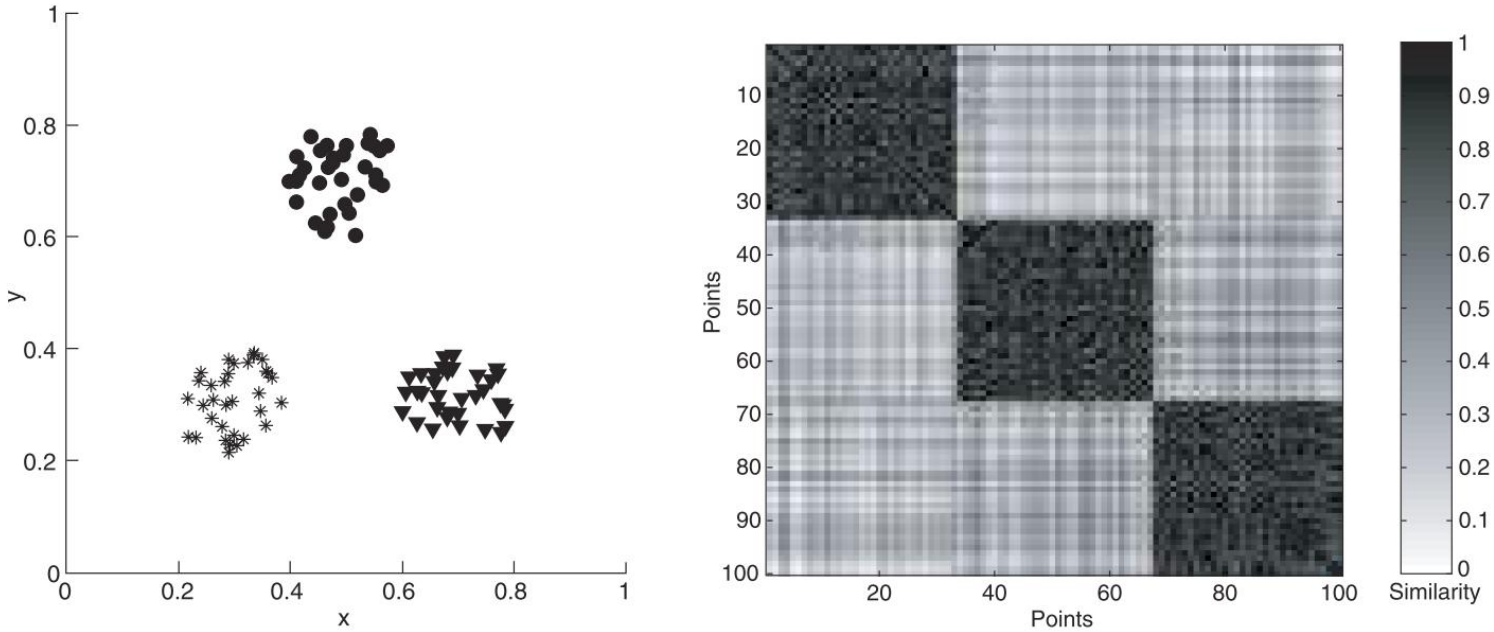


(a) Cohesion. (b) Separation.

Figure 8.28. Prototype-based view of cluster cohesion and separation.

Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

Visualize Similarity Matrix



(a) Well-separated clusters.

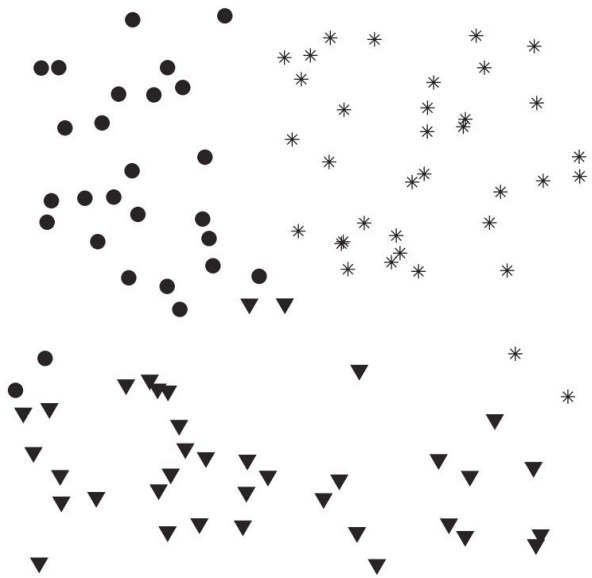
(b) Similarity matrix sorted by K-means cluster labels.

Figure 8.30. Similarity matrix for well-separated clusters.

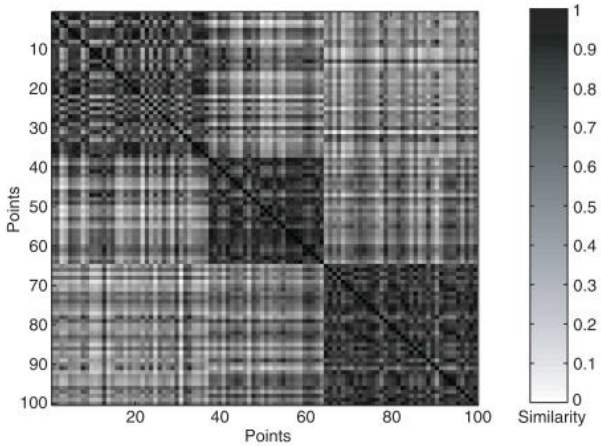
This technique is not limited to clustering in 2D!

Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

Visualize Similarity Matrix



(c) Three clusters found by K-means.



(b) Similarity matrix sorted by K-means cluster labels.

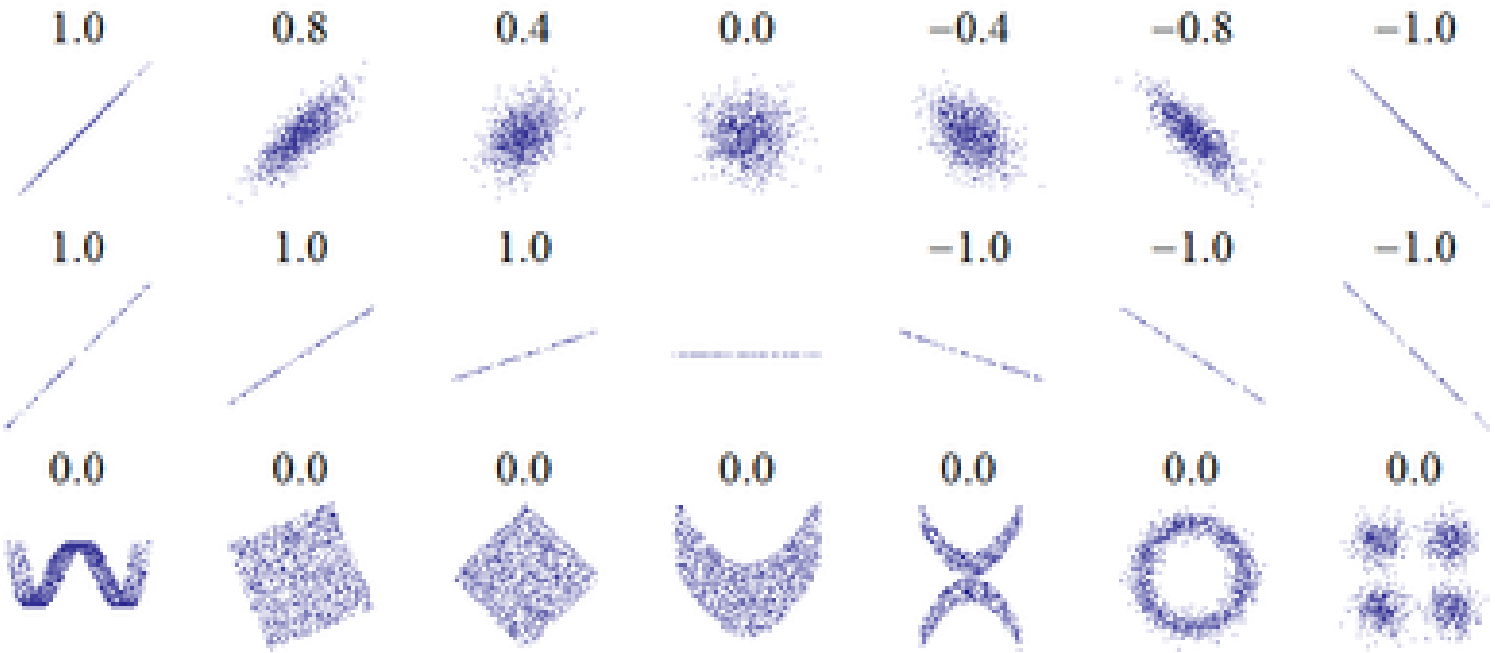
Source: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: *Introduction to Data Mining*. Addison Wesley, 2006

Pearson's Correlation Coefficient

A measure of the linear relationship between attributes.
Let $\vec{x} = (x_1, x_2, \ldots, x_n)$ and $\vec{y} = (y_1, y_2, \ldots, y_n)$.

$$
\begin{aligned}
\text{corr}(\vec{x}, \vec{y}) &= \frac{\text{covar}(\vec{x}, \vec{y})}{\sqrt{\text{var}(\vec{x})} \cdot \sqrt{\text{var}(\vec{y})}} \\
\text{covar}(\vec{x}, \vec{y}) &= \frac{1}{n-1} \Sigma_{k=1}^{n} (x_k - \bar{x})(y_k - \bar{y}) \\
\text{var}(\vec{x}) &= \frac{1}{n-1} \Sigma_{k=1}^{n} (x_k - \bar{x})^2 \\
\bar{x} &= \frac{1}{n} \Sigma_{k=1}^{n} x_k
\end{aligned}
$$

Pearson Correlation Coefficient



Source: Wikipedia

Note

On the following slides, correlation between
$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$ and $\begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}$
means correlation between
$(a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33})$ and
$(b_{11}, b_{12}, b_{13}, b_{21}, b_{22}, b_{23}, b_{31}, b_{32}, b_{33})$.

Correlation for Partitional Clustering

Assume a partitional clustering of n points $p_1, p_2, \ldots, p_n$.
Compute correlation between similarity matrix $S[n \times n]$ and ideal similarity matrix $C[n \times n]$ based on cluster labels.

- $S_{i,j}$ is the similarity between points p_i and p_j;
- $C_{i,j} = \begin{cases} 1 & \text{if } p_i \text{ and } p_j \text{ belong to the same cluster} \\ 0 & \text{otherwise} \end{cases}$

Correlation for Hierarchical Clustering

Assume a hierarchical clustering of n points $p_1, p_2, \ldots, p_n$.
Compute correlation between distance matrix $D[n \times n]$ and cophenetic distance matrix $C[n \times n]$.

- $D_{i,j}$ is the distance between points p_i and p_j;
- $C_{i,j}$ is the cophenetic distance between points p_i and p_j.

The cophenetic distance between p_i and p_j is defined as follows.

Let A be the smallest cluster that contains both p_i and p_j. Clearly, A has two "child" clusters B_1 and B_2 such that $p_i \in B_1$ and $p_j \in B_2$. The cophenetic distance between p_i and p_j is the distance between B_1 and B_2.