

Clustering

Daniel Aloise <daniel.aloise@polymtl.ca>

Apprentissage non supervisé

- L'apprentissage non supervisé vise à caractériser la distribution des données, et les relations (distances) entre les enregistrements.
- Il n'y a pas de connaissances a priori, pas d'ensemble d'entraînement.
- On va explorer aujourd'hui le type le plus populaire d'apprentissage non supervisé : **classification automatique - clustering**

Classification automatique

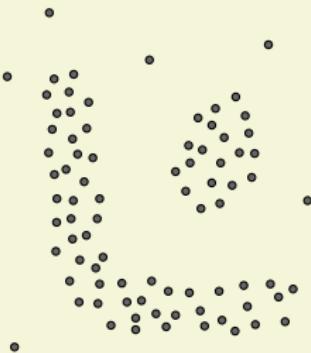
- Étant donné un ensemble d'objets, la classification **non supervisé** automatique a pour but de trouver des sous-ensembles (**clusters**) d'objets homogènes.

Classification automatique

points proches assemblés et pts éloignés seront mis de côté

- Étant donné un ensemble d'objets, la classification automatique a pour but de trouver des sous-ensembles (**clusters**) d'objets homogènes.

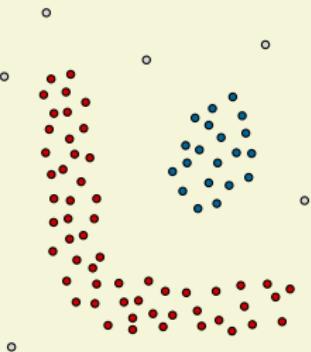
Exemple



Classification automatique

- Étant donné un ensemble d'objets, la classification automatique a pour but de trouver des sous-ensembles (**clusters**) d'objets homogènes.

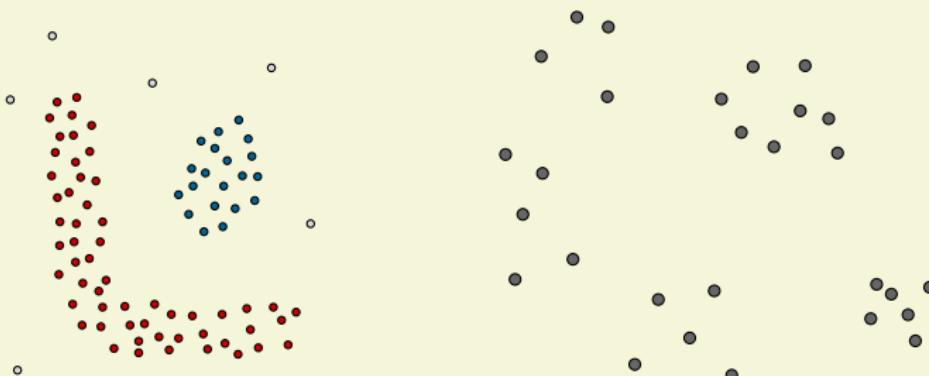
Exemple



Classification automatique

- Étant donné un ensemble d'objets, la classification automatique a pour but de trouver des sous-ensembles (**clusters**) d'objets homogènes.

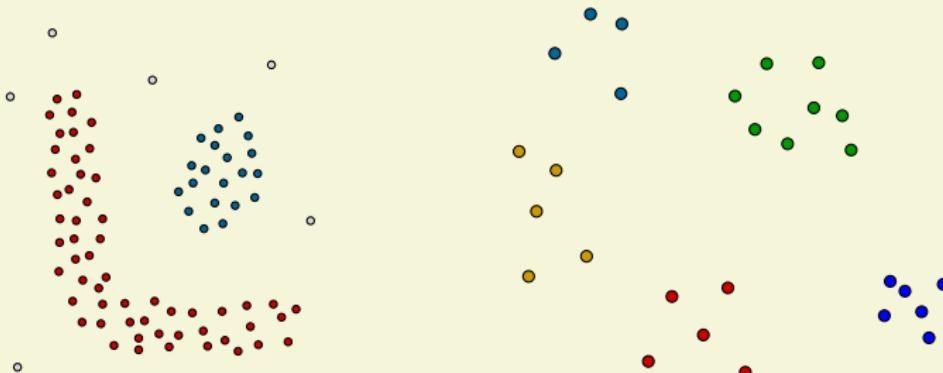
Exemple



Classification automatique

- Étant donné un ensemble d'objets, la classification automatique a pour but de trouver des sous-ensembles (**clusters**) d'objets homogènes.

Exemple



Classification automatique

non supervisés

- Les méthodes de classification automatique contribuent à détecter des groupes latents d'un ensemble de données.
- Le problème est étudié depuis le XVIII siècle par des naturalistes :

"Il me paraît que le seul moyen de faire une méthode instructive et naturelle, c'est de mettre ensemble les choses qui se ressemblent, et de séparer celles qui diffèrent les unes des autres." (Buffon)

rapprocher ce qui se ressemble et séparer ce qui est différent!

Applications

régions



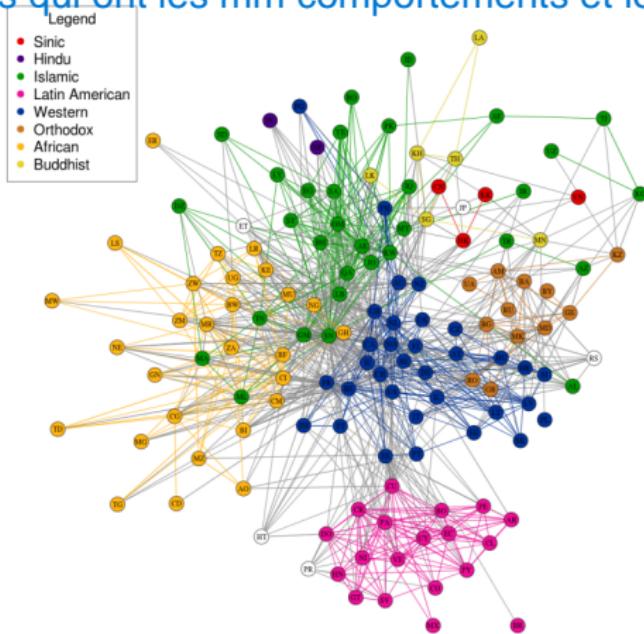
regroupement par IDH (2013)



selon caractéristique choisie on va grouper les données qui se ressemblent

Applications

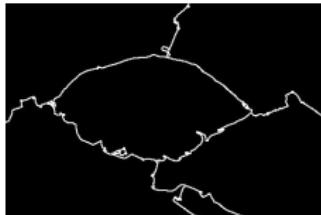
- Trouver des communautés dans les graphes :
Fb: trouver gens qui ont les mm comportements et les regrouper ensemble



Source : "The mesh of civilizations and international email flows"
Daniel Aloise <daniel.aloise@polymtl.ca> — Clustering — 3 juin 2020
al. (2013) 6/59

Applications

- Segmentation d'images :



(Coupe normalisée - Hansen, Ruiz & Aloise, *Pattern Recognition*, 2012)

Applications

Modélisation sur des ensembles réduits construire un modèle prédictif distinct pour chaque cluster.

ex Fb trouver communautés et entraîner modèle pr predire pour les gens qui se comportent de mm manière

Réduction des données remplacer / représenter chaque groupe d'éléments par son représentant.

Détection des valeurs aberrantes quels objets sont éloignés des centres de clusters ou coincés dans des clusters minuscules ?

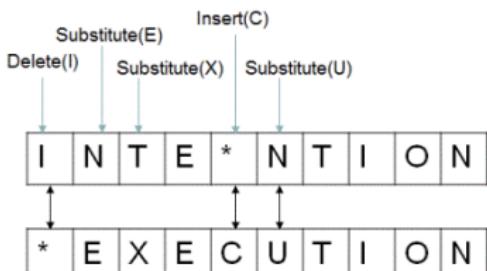
La moyenne n'a pas de sens des fois
ex: photos de chien
moyenne d'images de chiens ne va pas donner bon résultat

Ingrediénts

- Une collection $X = \{X_1, \dots, X_n\}$ de n objets de dimension d à classifier.
- Une matrice de dissimilarité $D = (d_{ij})$ entre les objets de X est calculée, tel que d_{ij} pour $i, j = 1, \dots, n$ satisfait :
 - $d_{ij} = d_{ji} \geq 0$;
 - $d_{ii} = 0$. si les instances sont le mm
- Ces valeurs n'ont pas besoin de satisfaire les inégalités triangulaires, c'est-à-dire, d'être des distances !

Mesures de dissimilarité

- De nombreuses mesures de similarité naturelles ne sont pas des mesures de distance :
 - Coefficient de corrélation (-1 à 1) valeur comprise entre -1 et 1
 - Produit scalaire entre des vecteurs donne aussi dissimilarité
- Tandis que d'autres le sont sans que cela soit évident :
 - *edit distance*



Distances euclidiennes

- La métrique de distance euclidienne traditionnelle pèse toutes les dimensions de la même manière :

$$d_{ij} = \sqrt{\sum_{s=1}^d (X_i^s - X_j^s)^2}$$

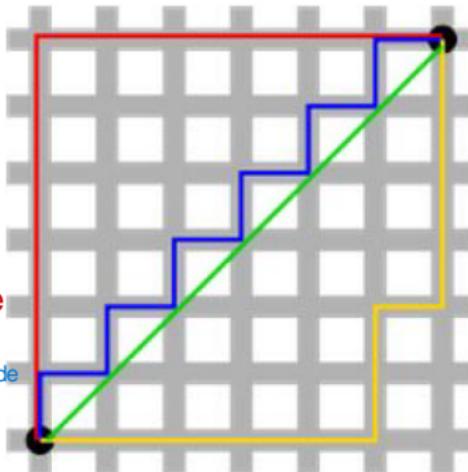
- On peut utiliser des coefficients pour donner des poids différents à chaque dimension.
- N'oubliez pas de **normaliser** pour rendre les dimensions comparables.

Distances de Minkowski

- Généralise la distance euclidienne :

$$d_{ij} = \left(\sum_{s=1}^d |X_i^s - X_j^s|^k \right)^{1/k}$$

- $k = 1$: distance de Manhattan **Feuille Notes manhattan et composante maximale**
- $k = \infty$: composante maximale (l'attribut pour lequel la différence est la plus grande (distance la + grande))
- Devons-nous nous soucier des différences dans toutes les dimensions, ou seulement des plus grandes ?



Distances de Minkowski

- Qui de $X_1 = (2, 0)$ et $X_2 = (1, 5, 1, 5)$ est le plus éloigné de l'origine $(0, 0)$?

Distances de Minkowski

- Qui de $X_1 = (2, 0)$ et $X_2 = (1, 5, 1, 5)$ est le plus éloigné de l'origine $(0, 0)$?
- Si $k = 1$, les distances sont 2 et 3, alors X_2
- Si $k = 2$, les distances sont 2 et 2.12, alors X_2
- Si $k = \infty$, les distances sont 2 et 1.5, alors X_1

Distances de Minkowski

- Qui de $X_1 = (2, 0)$ et $X_2 = (1, 5, 1, 5)$ est le plus éloigné de l'origine $(0, 0)$?
- Si $k = 1$, les distances sont 2 et 3, alors X_2
- Si $k = 2$, les distances sont 2 et 2.12, alors X_2
- Si $k = \infty$, les distances sont 2 et 1.5, alors X_1
- La métrique de distance définit le point le plus proche !

Distances pour des données binaires

- 1 Est-ce qu'il y a du sens calculer des distances euclidiennes pour des données binaires ?

Non

Distances pour des données binaires

- ① Est-ce qu'il y a du sens calculer des distances euclidiennes pour des données binaires ?
- ② Avez-vous donc des suggestions de distances pour des données binaires ?

Le type des données influence directement le choix de la métrique de distance !

Jaccard distance

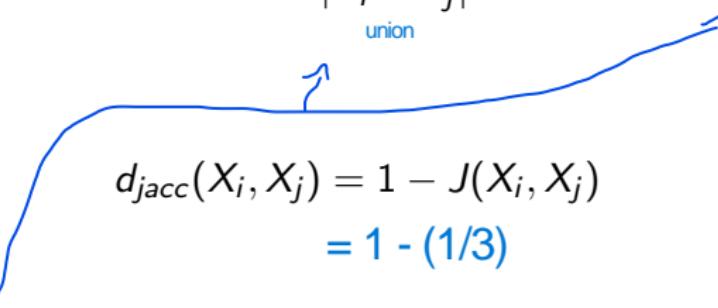
Jaccard index

$$J(X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|} \stackrel{\text{intersection}}{\underset{\text{union}}{\longrightarrow}} [0, 1] = 1/3$$

Jaccard distance

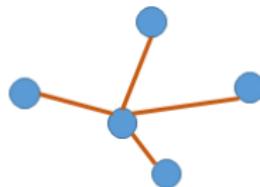
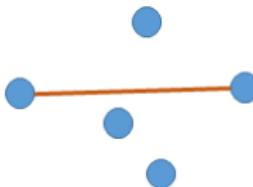
$$\begin{aligned} X_1 &= [0, 1, 0, 0, 0, 0] \\ X_2 &= [1, 1, 1, 0, 0, 0] \end{aligned}$$

↑ ↑ ↑

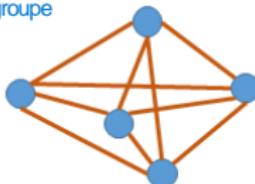

$$\begin{aligned} d_{jacc}(X_i, X_j) &= 1 - J(X_i, X_j) \\ &= 1 - (1/3) \end{aligned}$$

Critères de clustering

- Un **critère** de classification exprime l'**homogénéité** et/ou la **séparation** des classes trouvées.
- L'homogénéité d'un cluster C_ℓ est souvent mesurée par les :



diamètre
la + grande distance appartenant à un groupe

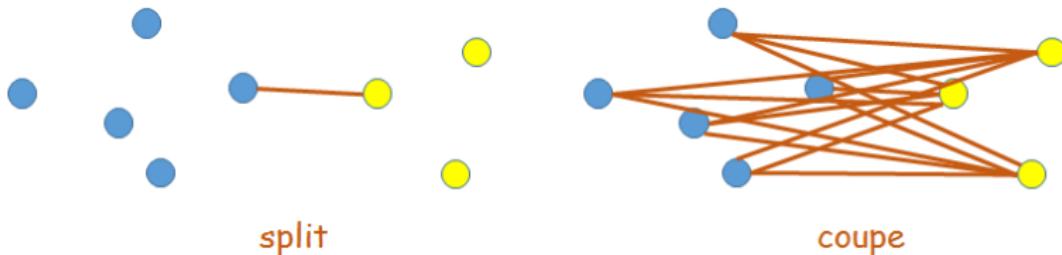


clique
ensemble des distances entre chacun de pts avec chacun des pts

Critères de clustering

à quel point deux groupes sont différents un à l'autre

- La séparation de C_ℓ peut être exprimée par les :



- Deux familles de critères :

- maximisation de mesures de séparation on veut que groupes soient le plus loin possibles
- minimisation de mesures d'homogénéité chacun des groupes soit le + compact et homogène possible, qu'ils soient proches et similaires

Types de clustering

- Les types les plus couramment utilisés sont la **partition** et la **hiérarchie de partitions** :

(i) **Partition** $P_k = \{C_1, C_2, \dots, C_k\}$ de X en k classes :

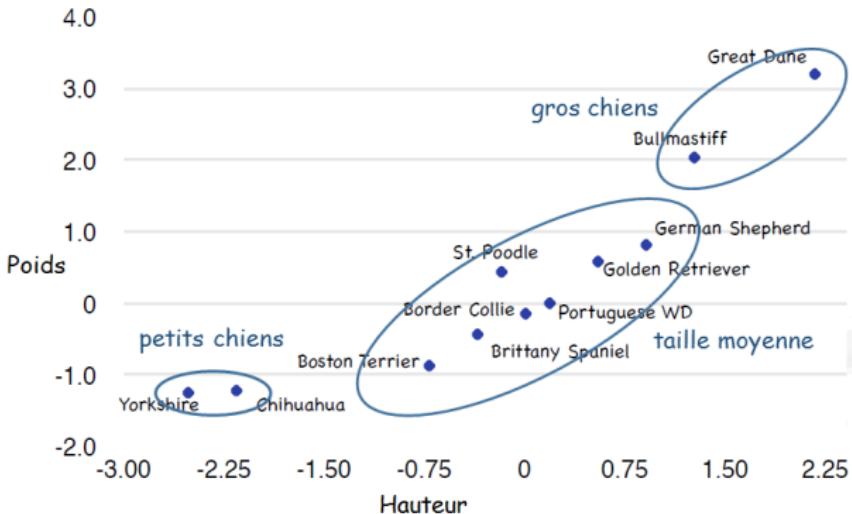
(i a) $C_i \neq \emptyset \quad i = 1, 2, \dots, k;$

(i b) $C_i \cap C_j = \emptyset \quad i, j = 1, 2, \dots, k \text{ et } i \neq j;$

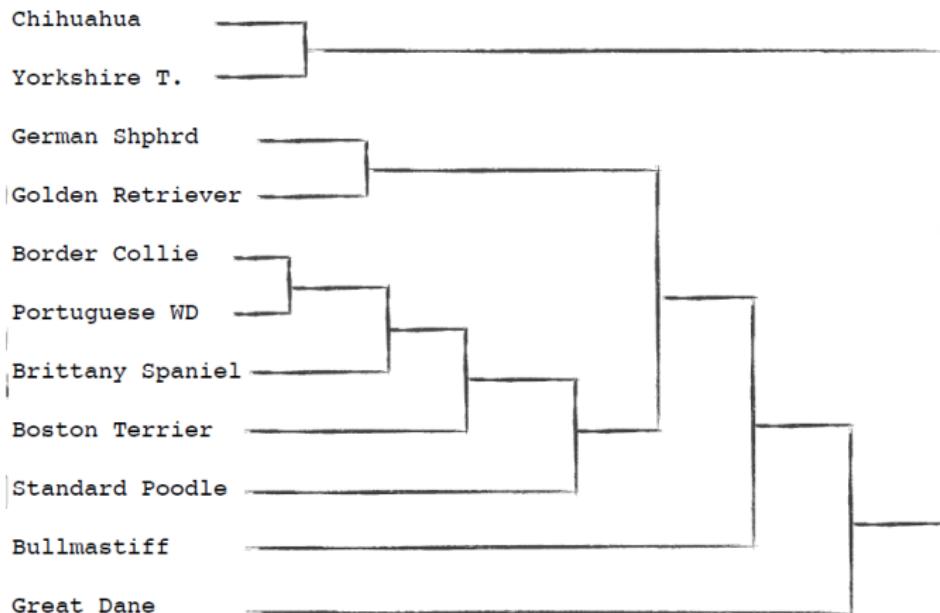
(i c) $\bigcup_{j=1}^k C_j = O;$

(ii) **Hiérarchie** : ensemble imbriqué de partitions de X

Exemple - Partition



Exemple - Hiérarchie



Partitionnement

- Le nombre de partitions différentes de n enregistrements en k clusters est donné par un nombre de Stirling de deuxième ordre :

$$S(n, k) = \frac{1}{k!} \sum_{\ell=0}^{k-1} (-1)^\ell \binom{k}{\ell} (k - \ell)^n.$$

nk	1	2	3	4	5	6	7	8	9	10
1	1									
2	1	1								
3	1	3	1							
4	1	7	6	1						
5	1	15	25	10	1					
6	1	31	90	65	15	1				
7	1	63	301	350	140	21	1			
8	1	127	966	1701	1050	266	28	1		
9	1	255	3025	7770	6951	2646	462	36	1	
10	1	511	9330	34105	42525	22827	5880	750	45	1

- Ex. $S(15, 3) = 2.375.101$, $S(20, 4) = 45.232.115.901$, et
 $S(100, 5) \approx 10^{68}$ - ces problèmes sont petits !

Quelques faits

- La complexité d'un problème de classification automatique dépend du critère utilisé :
 - maximiser le split peut être résolu en temps polynomial
 - minimiser le diamètre est NP-difficile
- *Téorème de l'impossibilité de Kleinberg (2002)* : il n'existe pas un critère de classification qui soit le **meilleur** pour tous les jeux possibles de données.



+ on a de dimension, + il faut des pts pour couvrir l'espace

- *Malédiction de la dimensionnalité* : objets ayant une grande quantité de attributs ont tendance à être **également différents**.

volume sphère devient de + en + petit quand le volume du cube augmente



Algorithmes

Types :

- ① Regroupement basé sur les centres
 - Minimisent les distances intra-clusters (erreur)
- ② Regroupement hiérarchique
 - Construit des clusters imbriqués ^{groupes} en les fusionnant ou en les divisant successivement
- ③ Regroupement basé sur la densité
 - Robuste au bruit
 - Bon pour identifier des groupes de formes arbitraires
- ④ Regroupement par des mélanges
 - Produit des clusters *soft*

Type 1 : Algorithme de k -moyennes

- Données $X = \{x_1, \dots, x_n\}$
- k est le nombre de clusters
- Centres μ_i , avec $i = 1, \dots, k$

K-MEANS(x_1, \dots, x_n, k)

initialize μ_1, \dots, μ_k (e.g. randomly) initialiser le centre des clusters aléatoirement

Repeat

 classify x_1, \dots, x_n to each's nearest μ_i regarder le groupe le + proche des pts

 recalculate μ_1, \dots, μ_k recalculer. Vu que le centre a bougé on va recalculer jusqu'à ce que le centre ne bouge plus

Until no change in μ_1, \dots, μ_k

Return(μ_1, \dots, μ_k)

Type 1 : Algorithme de k -moyennes

Mesure de distance

- Distance $d(x, \mu)$ entre un élément x et un centre μ

$$d(x, \mu) = \|x - \mu\|^2$$

- Alors, x_i est attribué à μ_{j^*} seulement si

j qui minimise la distance entre un point et le centre

$$j^* := \underset{j \in \{1, \dots, k\}}{\operatorname{argmin}} \{\|x_i - \mu_j\|^2\}$$

Mise à jour de centres

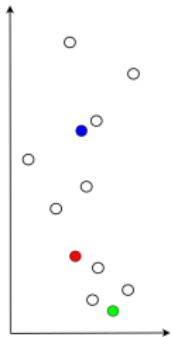
- Soit $C = \{C_1, \dots, C_k\}$ l'ensemble de clusters

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

moyenne des pts qui appartiennent au cluster

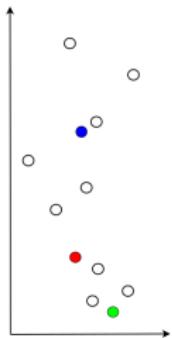
- Le centre est donc situé au **centroïde** du cluster

Algorithme de k -moyennes

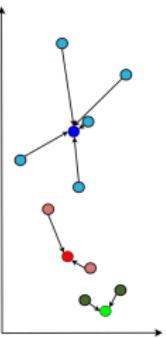


(a)

Algorithme de k -moyennes

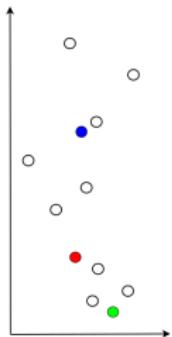


(a)

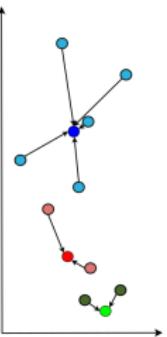


(b)

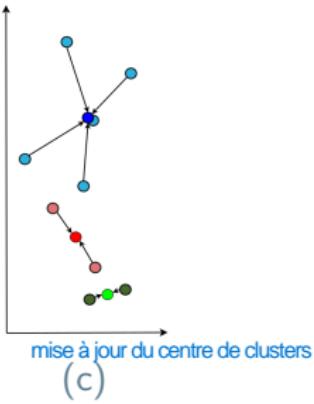
Algorithme de k -moyennes



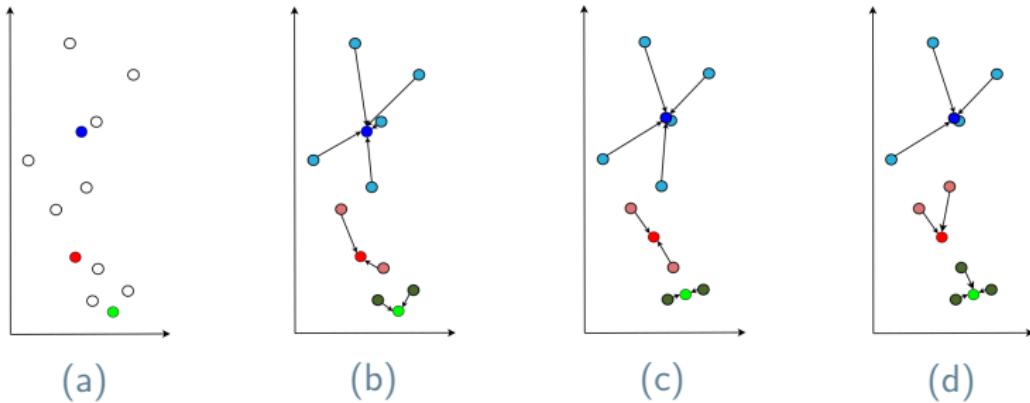
(a)



(b)

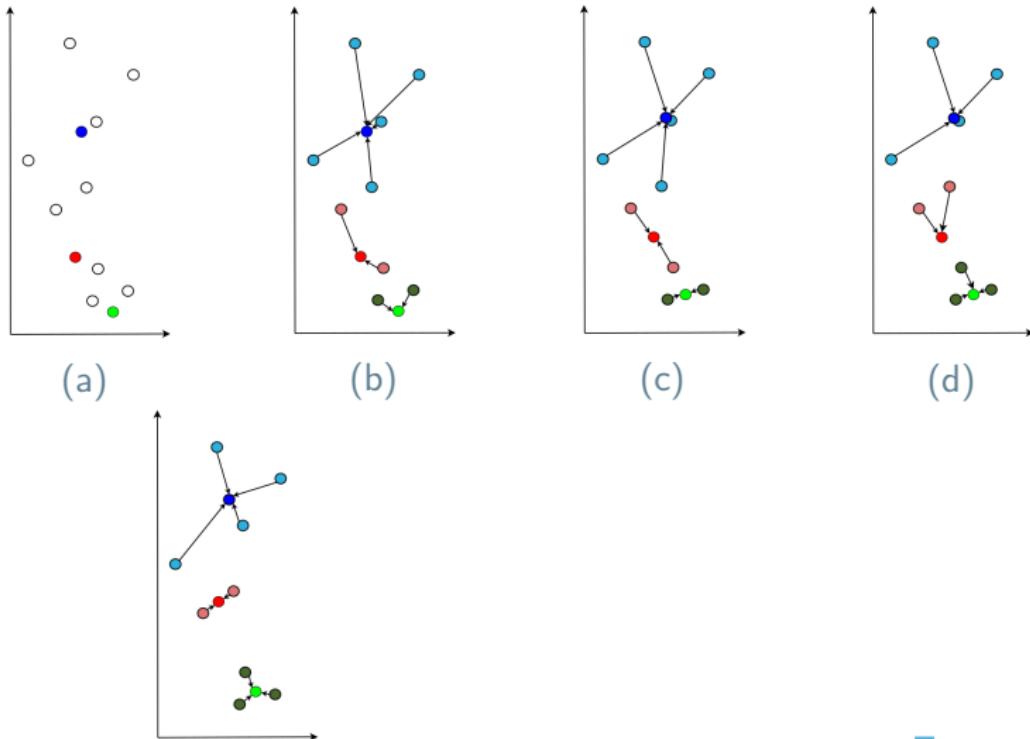
(c)
mise à jour du centre de clusters

Algorithme de k -moyennes

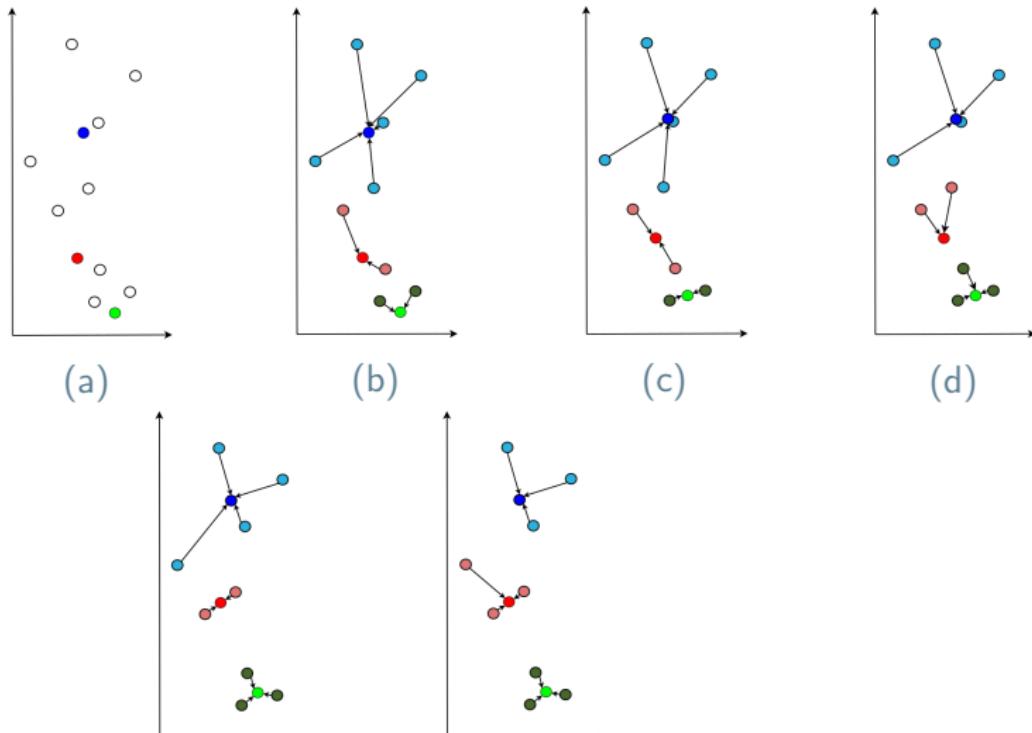


individus assignés au centre le
+ proche

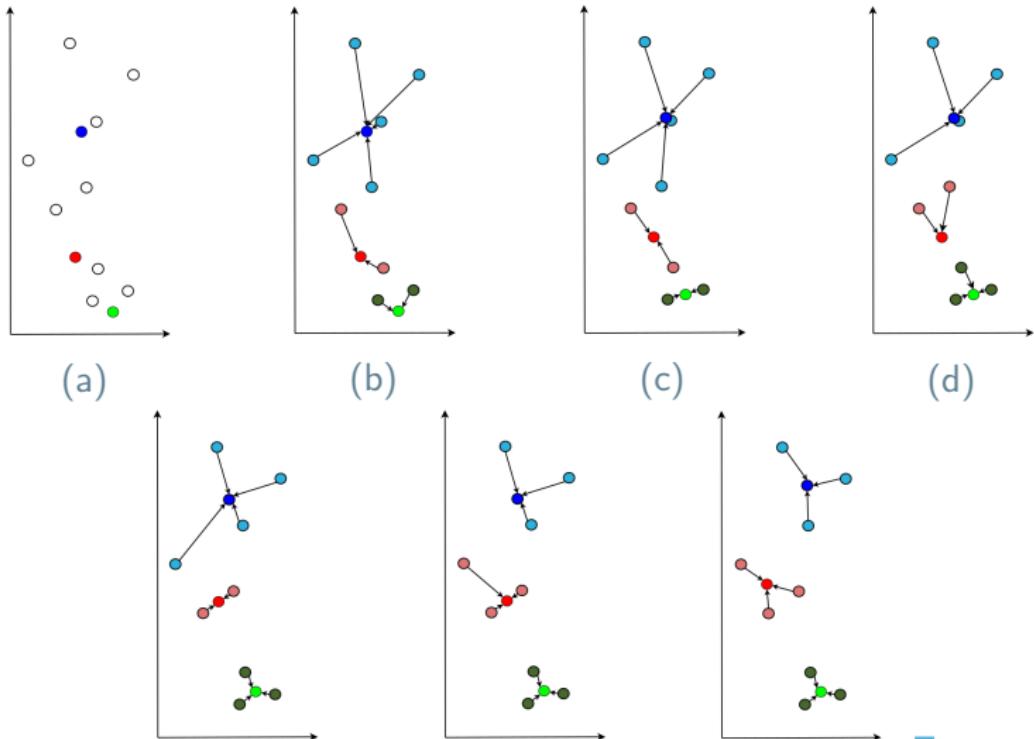
Algorithme de k -moyennes



Algorithme de k -moyennes



Algorithme de k -moyennes



Algorithme de k -moyennes

Characteristics

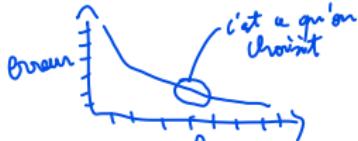
- Très simple et performant
- Minimisent la variance (erreurs au carré) \Rightarrow clusters ronds
- Méthode d'optimisation locale
- Sûrement l'algorithme le plus populaire pour faire de regroupements

Problèmes

- Sensible à l'initialisation des centres
- La décision sur le nombre de clusters n'est pas évidente pour les applications réelles

Combien de clusters on doit trouver ?

- Le «bon» nombre de clusters est généralement inconnu avant le regroupement.
- Pour k -moyennes, l'erreur diminue lors que l'on ajoute des clusters.
- Pourtant, l'erreur diminue lentement, une fois dépassé le bon nombre de clusters.
- **My rule of thumb :** le bon nombre est celui que te permet de raconter une histoire sur tes données.



Type 2 : Algorithmes hiérarchiques

Algorithme de l'emballage!

Forward: on part de 0 et on rajoute petit à petit des individus

Backward: on part de plusieurs individus et on exclut petit à petit

Approche d'agglomération

- ① Commence avec une partition initiale avec n singletons
- ② Fusionne successivement les clusters en fonction d'un critère local
- ③ Répète jusqu'à ce que toutes données appartiennent au même cluster

Approche de division (moins utilisée)

- ① Commence par un cluster initial avec toutes les données
- ② Effectue des bipartitions successives d'un cluster à la fois
- ③ Répète jusqu'à l'obtention de singleton clusters

Approche d'agglomération

Single-linkage Algorithm

Fusionne deux clusters C_i et C_j si $D(C_i, C_j)$ est la plus petite valeur parmi toutes les paires de clusters

$$D(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} \text{distance entre 2 clusters}$$

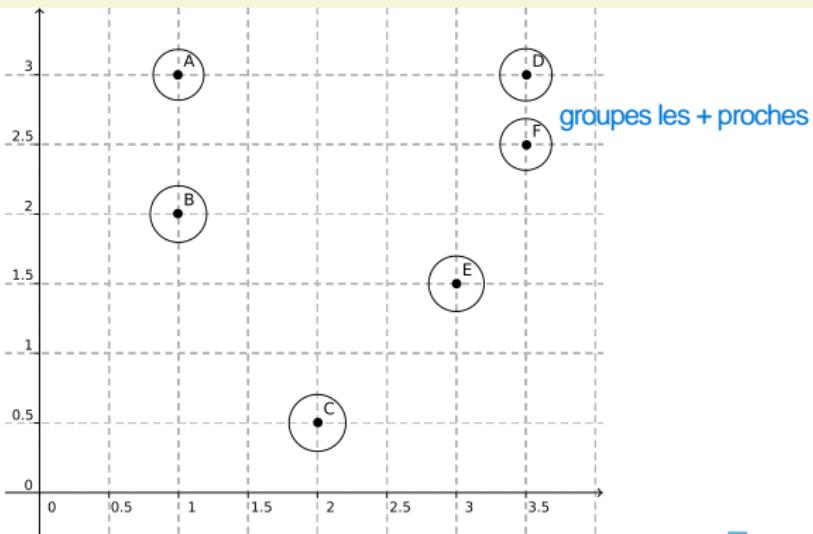
plus petite distance entre cluster 1 et 2

Approche d'agglomération

Single-linkage Algorithm

Fusionne deux clusters C_i et C_j si $D(C_i, C_j)$ est la plus petite valeur parmi toutes les paires de clusters

$$D(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$

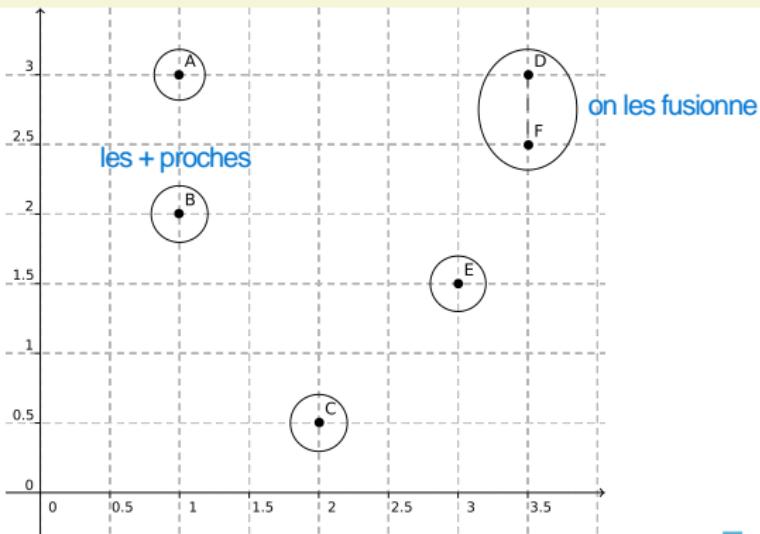


Approche d'agglomération

Single-linkage Algorithm

Fusionne deux clusters C_i et C_j si $D(C_i, C_j)$ est la plus petite valeur parmi toutes les paires de clusters

$$D(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$

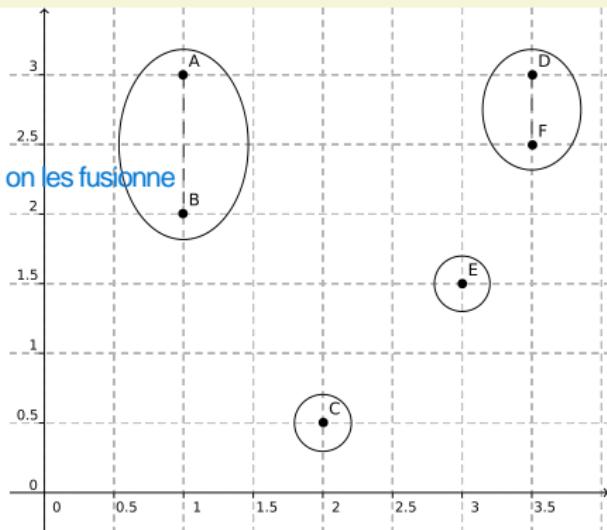


Approche d'agglomération

Single-linkage Algorithm

Fusionne deux clusters C_i et C_j si $D(C_i, C_j)$ est la plus petite valeur parmi toutes les paires de clusters

$$D(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$

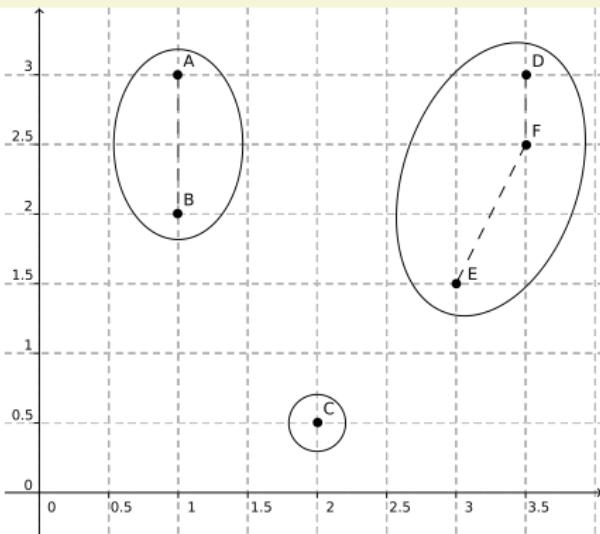


Approche d'agglomération

Single-linkage Algorithm

Fusionne deux clusters C_i et C_j si $D(C_i, C_j)$ est la plus petite valeur parmi toutes les paires de clusters

$$D(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$

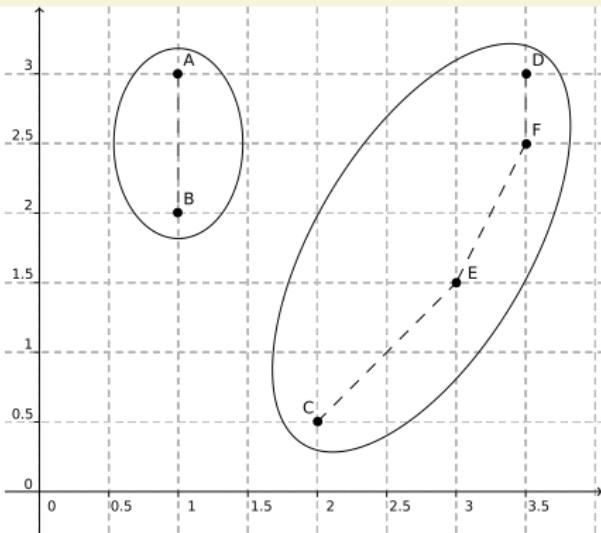


Approche d'agglomération

Single-linkage Algorithm

Fusionne deux clusters C_i et C_j si $D(C_i, C_j)$ est la plus petite valeur parmi toutes les paires de clusters

$$D(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$

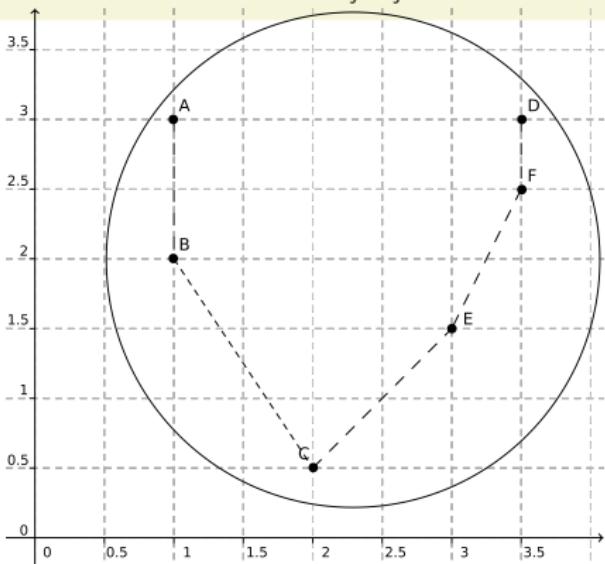


Approche d'agglomération

Single-linkage Algorithm

Fusionne deux clusters C_i et C_j si $D(C_i, C_j)$ est la plus petite valeur parmi toutes les paires de clusters

$$D(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$



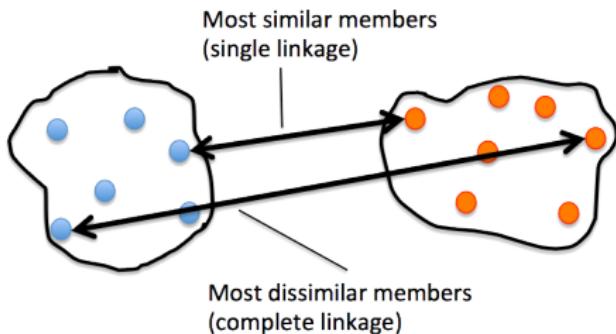
Approche d'agglomération

Complete-linkage Algorithm

Fusionne deux clusters C_i et C_j si $D(C_i, C_j)$ est la plus petite valeur parmi toutes les paires des clusters

la plus petite distance

$$D(C_i, C_j) = \max_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$



Approche d'agglomération

Complete-linkage Algorithm

Exercice : Considérez l'ensemble d'objets donné ainsi que leurs dissimilarités

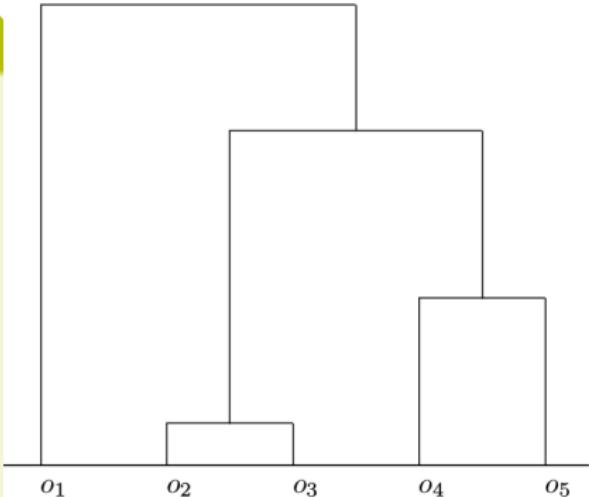
	o_1	o_2	o_3	o_4	o_5
o_1	0	4	9	7	11
o_2	4	0	1	6	3
o_3	9	1	0	8	6
o_4	7	6	8	0	4
o_5	11	3	6	4	0

Approche d'agglomération

Complete-linkage Algorithm

Exercice : Considérez l'ensemble d'objets donné ainsi que leurs dissimilarités

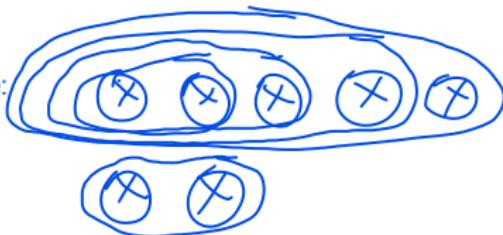
	o_1	o_2	o_3	o_4	o_5
o_1	0	4	9	7	11
o_2	4	0	1	6	3
o_3	9	1	0	8	6
o_4	7	6	8	0	4
o_5	11	3	6	4	0



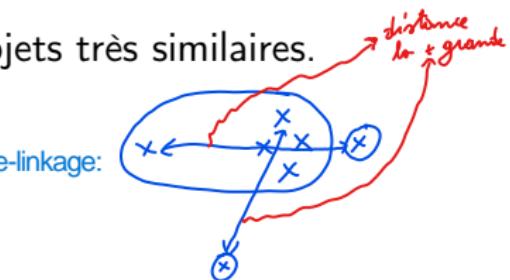
Problèmes

- **single-linkage** peut regrouper ensemble des objets très différents. on peut faire des chaines, des clusters reliés les uns des autres
- **complete-linkage** peut séparer des objets très similaires.

Single-linkage:



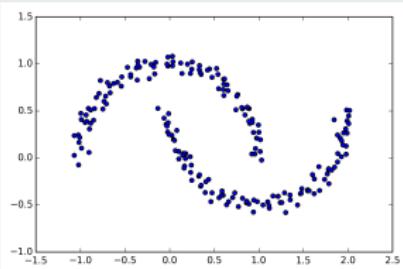
complete-linkage:



Type 3 : DBSCAN

Motivation

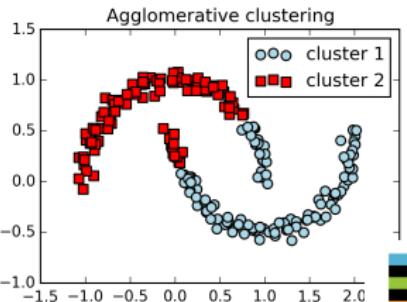
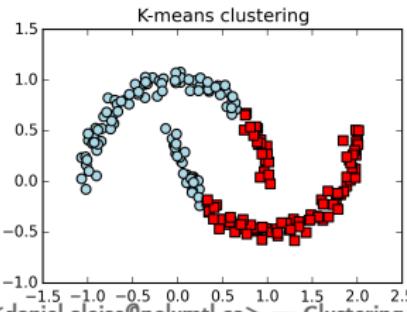
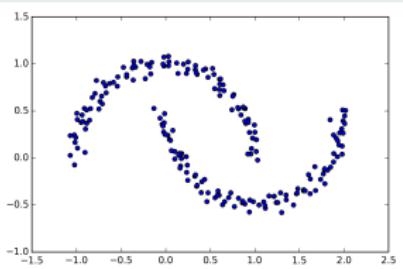
Qu'est-ce que se passe si on applique les algorithmes précédents sur les données ci-dessous ?



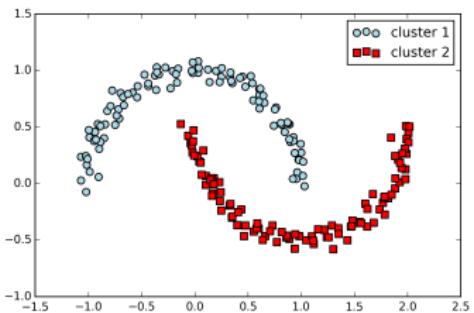
Type 3 : DBSCAN

Motivation

Qu'est-ce que se passe si on applique les algorithmes précédents sur les données ci-dessous ?



Type 3 : DBSCAN



Motivation

- ① Chaque cluster sera une région dense de points
- ② Les régions peu denses séparent les clusters
- ③ Aproposé pour des jeux de données avec des structures non régulières, avec la présence d'outliers ou de bruit.

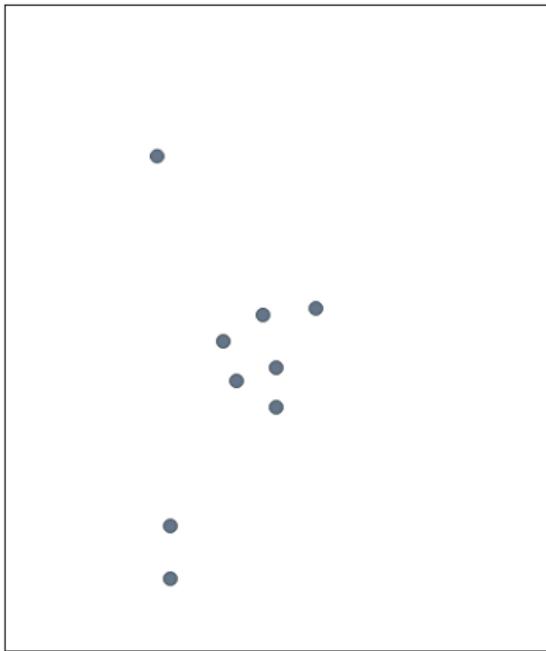
Type 3 : DBSCAN

Density-Based Spatial Clustering of Applications with Noise

- Demande deux paramètres
 - ① ϵ -distance : si deux points X_i et X_j sont à une distance $d_{ij} \leq \epsilon$, ils sont voisins.
 - ② m : nombre minimal de points requis pour former un cluster
- Comment ça marche (résume) :
 - ① On commence avec un point au hasard X_i
 - ② On regroupe X_i avec ses ϵ -voisins pour composer un cluster
 - ③ Si un point est trouvé comme part d'un cluster, son ϵ -voisinage est lui aussi part du même cluster
 - ④ Si ce processus abouti à un cluster d'au moins m points, on le garde. Sinon ils sont considérés du bruit.
 - ⑤ On retourne au pas no. 1 jusqu'à ce que tous les points soient considérés

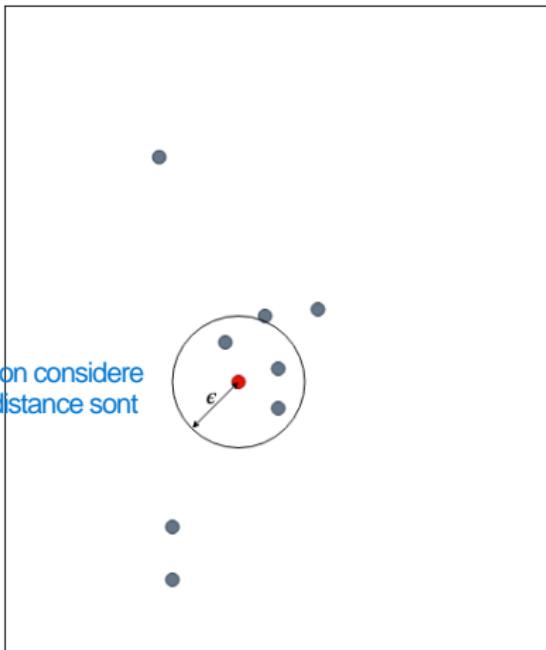
Point considéré comme du bruit si aucun voisin
ex: Noise = {x7}

Type 3 : DBSCAN

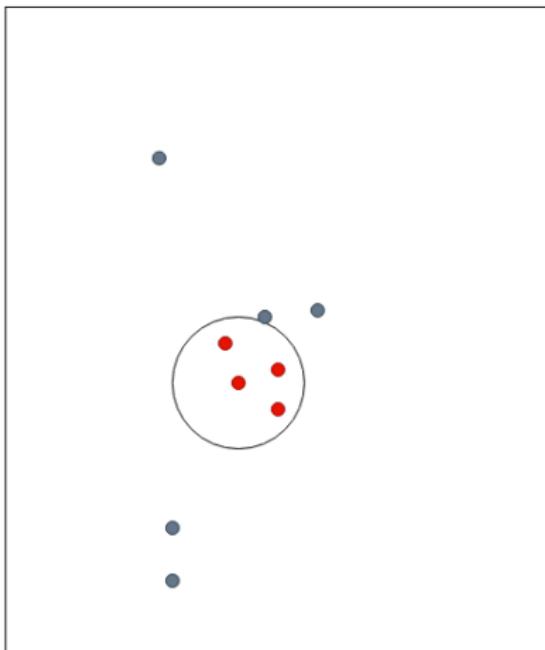


Type 3 : DBSCAN

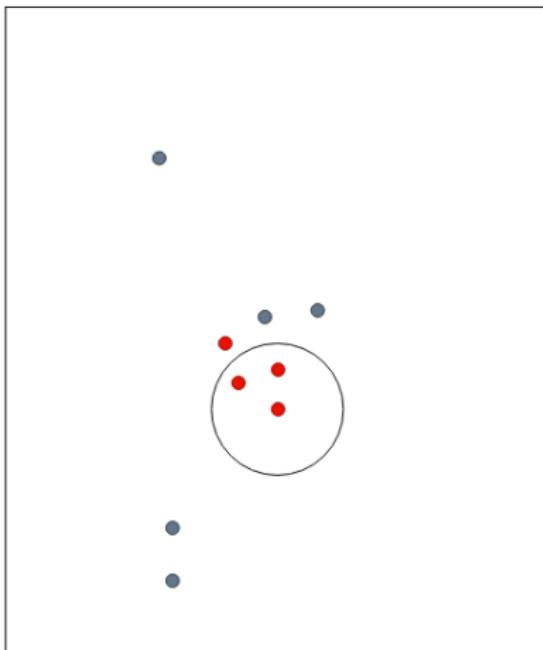
epsilon est la distance et on considère que tous les pts dans cette distance sont nos pts



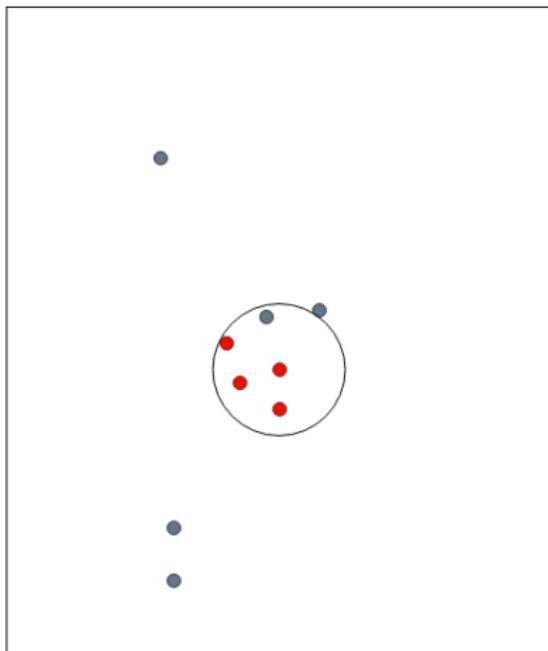
Type 3 : DBSCAN



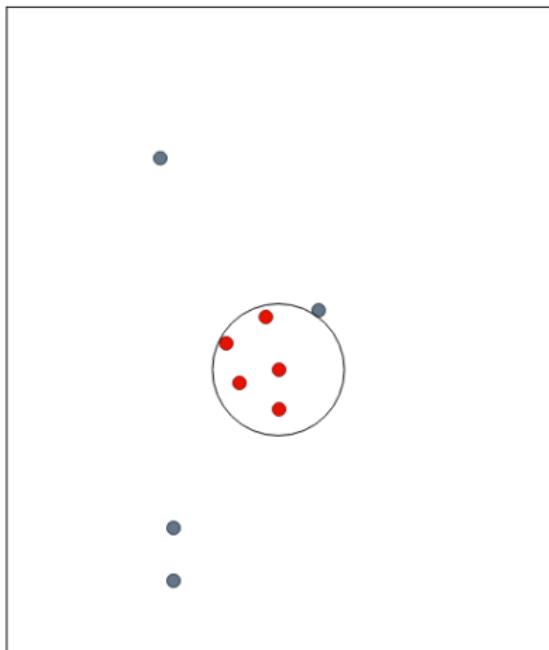
Type 3 : DBSCAN



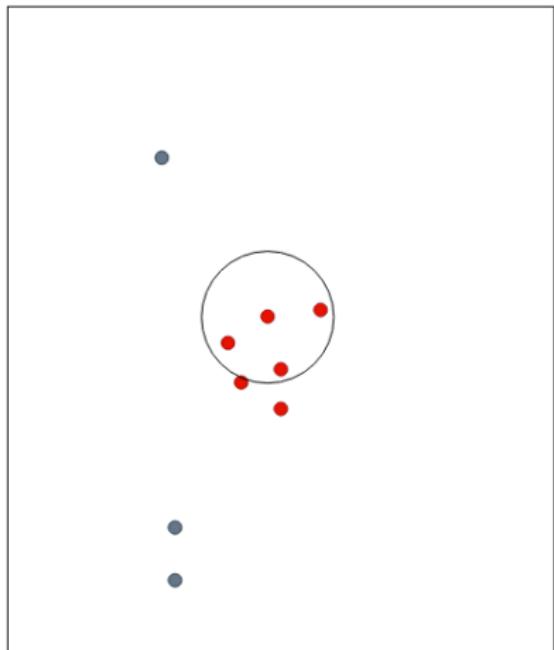
Type 3 : DBSCAN



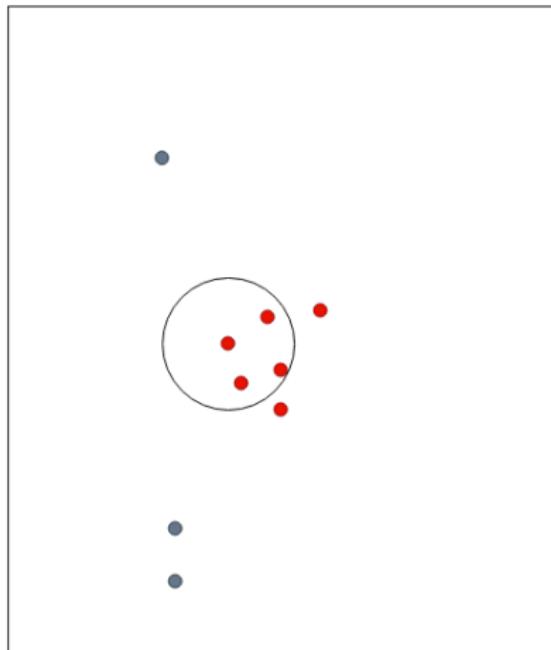
Type 3 : DBSCAN



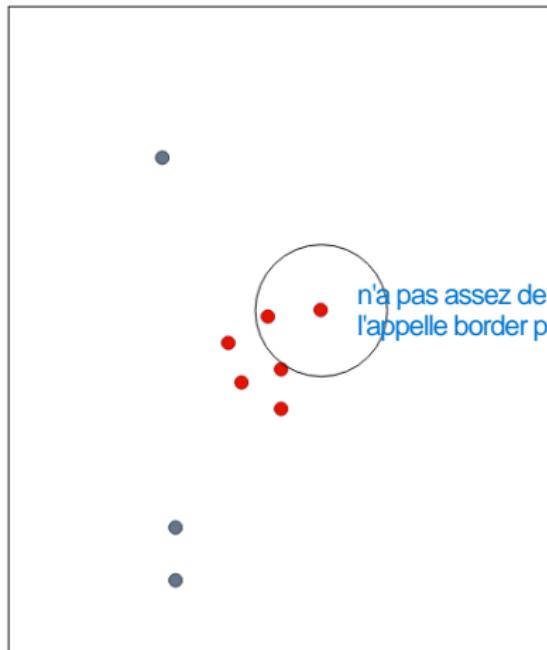
Type 3 : DBSCAN



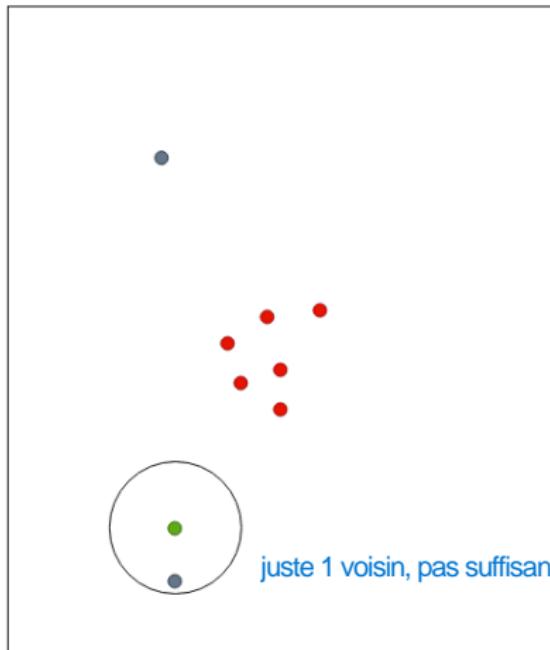
Type 3 : DBSCAN



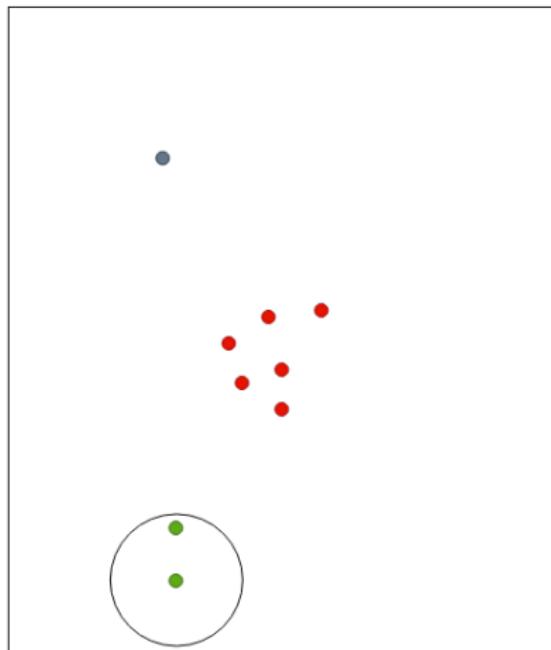
Type 3 : DBSCAN



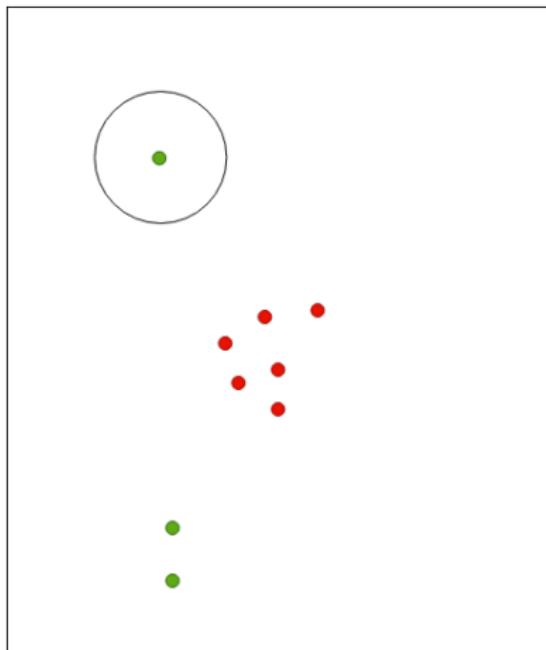
Type 3 : DBSCAN



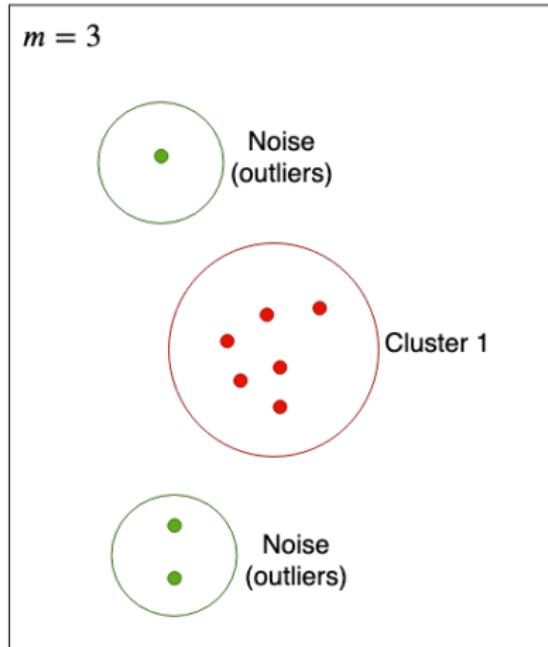
Type 3 : DBSCAN



Type 3 : DBSCAN



Type 3 : DBSCAN



Type 3 : DBSCAN

Avantages

- Plus robuste au bruit et aux outliers
- On n'a besoin de connaître le nombre de clusters a priori

Désavantages

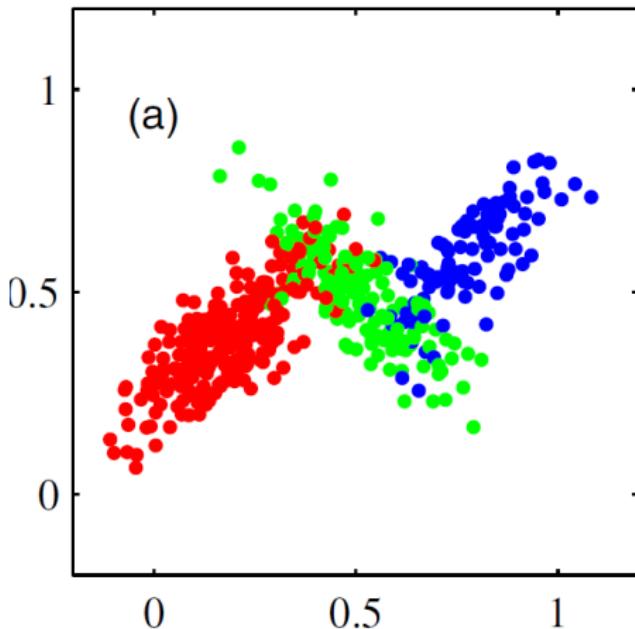
- La qualité du clustering dépend fortement des paramètres
- Mauvais pour classifier des jeux de données avec grandes différences de densité

Mélange de gaussiennes

- Modèle léger (*soft*) de clustering :
 - Les points peuvent être associés à plus d'un cluster.
- Un **mélange de gaussiennes** suppose que les données ont été générées comme suit :
- Pour $i = 1 \dots n$
 - un entier $j = 1, \dots, k$ est choisi selon les probabilités π_1, \dots, π_k
 - X_i est généré selon une loi de probabilité $\mathcal{N}(X_i | \mu_j, \Sigma_j)$
- Autrement dit, les données sont échantillonnées à partir de $j = 1, \dots, k$ différentes distributions gaussiennes, ayant chacune une moyenne μ_j et une covariance Σ_j

Mélange de gaussiennes

- Exemple de données générées d'un mélange de $k = 3$ gaussiennes



Mélange de gaussiennes

Probabilité a priori du choix de la gaussienne

- On va noter Z_i la variable aléatoire correspondant à l'identité de la gaussienne qui a généré une donnée X_i
- Format *one-hot* : $z_{ij} = 1$ si X_i a été générée par la j^{eme} gaussienne
- La probabilité de choisir la j^{eme} gaussienne pour générer X_i est donc :

$$p(z_{ij} = 1) = \pi_j$$

du même que :

$$p(Z_i) = \prod_{j=1}^k \pi_j^{z_{ij}}$$

Z_{ij} vaut 0 tt temps sauf pour 1

Mélange de gaussiennes

Fonction de densité conditionnelle de la donnée

- Sachant Z_i , la probabilité (fonction de densité) de X_i est

| -> veut dire sachant

$$p(X_i|z_{ij} = 1) \stackrel{\text{sachant}}{=} \mathcal{N}(X_i|\mu_j, \Sigma_j)$$

- Pour une gaussienne quelconque, ça vaut :

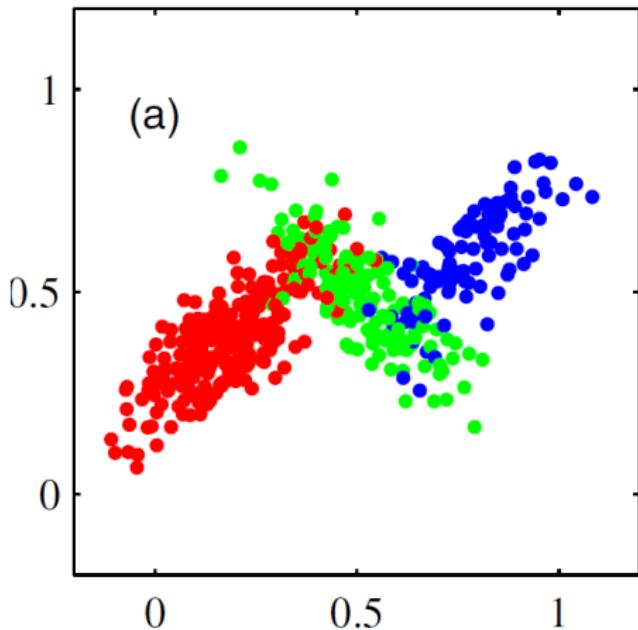
$$\mathcal{N}(X|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)}$$

- qu'on peut aussi écrire

$$p(X_i|Z_i) = \prod_{j=1}^k \mathcal{N}(X_i|\mu_j, \Sigma_j)^{z_{ij}}$$

Mélange de gaussiennes

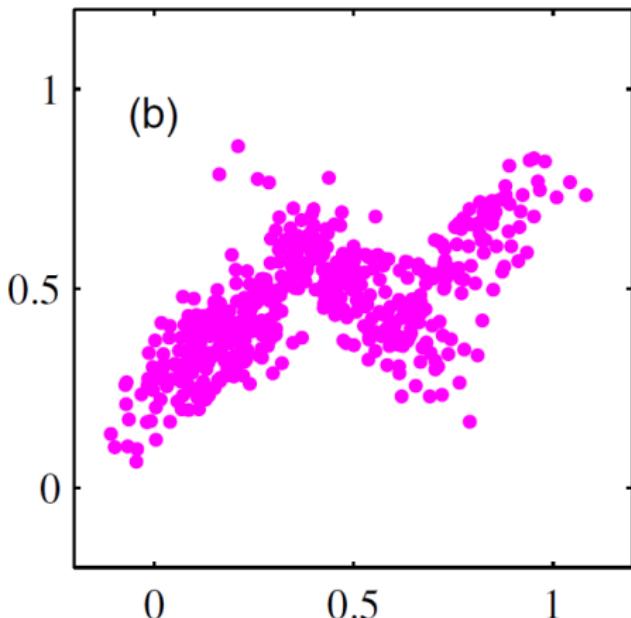
- Exemple de données générées d'un mélange de $k = 3$ gaussiennes



source : Laro

Mélange de gaussiennes

- Pourtant, pour le clustering, l'appartenance aux k gaussiennes («clusters») n'est pas connue



source : Larochelle, 2015

Mélange de gaussiennes

fonction de densité marginale des données

- Puisqu'on ne connaît pas l'appartenance aux gaussiennes Z_i , on va s'intéresser à la probabilité marginale

$$p(X_i) = \sum_{j=1}^k p(z_{ij})p(X_i|z_{ij}) = \sum_{j=1}^k \pi_j \mathcal{N}(X_i|\mu_j, \Sigma_j)$$

proba
selectionner la
jème pour choisir
le ième

- Nous voulons regrouper nos données en fonction des **probabilités d'appartenance** à chacune des gaussiennes

Mélange de gaussiennes

Probabilité d'appartenance

- À l'aide de la **règle de Bayes**, la probabilité d'appartenance à la j^* -ème gaussienne est calculée :

$$\gamma(z_{ij*}) \equiv p(z_{ij*} = 1 | X_i) = \frac{p(z_{ij*} = 1)p(X_i | z_{ij*} = 1)}{\sum_{j=1}^k p(z_{ij} = 1)p(X_i | z_{ij} = 1)}$$
$$= \frac{\pi_{j*} \mathcal{N}(X_i | \mu_{j*}, \Sigma_{j*})}{\sum_{j=1}^k \pi_j \mathcal{N}(X_i | \mu_j, \Sigma_j)}$$

Mélange de gaussiennes

- Mais, nous ne connaissons pas les gaussiennes (μ, Σ) !
- Il se peut même que les données de chaque cluster ne soient pas générées des gaussiennes
- Cela est une **hypothèse** de ce modèle de clustering
- ... qu'on peut changer d'ailleurs : modèles avec (d'autres) mélanges

Mélange de gaussiennes

Maximum de vraisemblance

- On maximise un mélange de gaussiennes par (log)-vraisemblance

$$\log p(X|\pi, \mu, \Sigma) = \sum_{i=1}^n \log \left\{ \sum_{j=1}^k \pi_j \mathcal{N}(X_i | \mu_j, \Sigma_j) \right\}$$

- Remarque qu'on a une somme pour tous les données $X_1 \dots X_n$

Mélange de gaussiennes

- En utilisant les conditions d'optimalité de premier ordre, on arrive aux expressions suivantes

$$\mu_j = \frac{1}{n_j} \sum_{i=1}^n \gamma(z_{ij}) X_i \quad \text{où} \quad n_j = \sum_{i=1}^n \gamma(z_{ij})$$

$$\pi_j = \frac{n_j}{n}$$

$$\Sigma_j = \frac{1}{n_j} \sum_{i=1}^n \gamma(z_{ij})(X_i - \mu_j)(X_i - \mu_j)^T$$

si on suppose que les $\gamma(z_{ij})$ sont fixes alors on est capables de calculer les paramètres des distributions

Mélange de gaussiennes

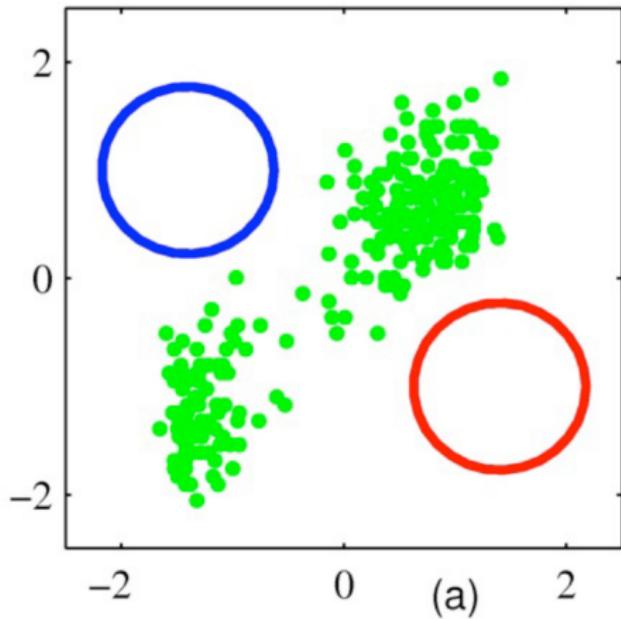
- Les solutions pour μ_j, π_j et Σ_j supposent que les $\gamma(z_{ij})$ sont fixes
- par contre, changer μ_j, π_j et Σ_j change aussi $\gamma(z_{ij})$

Mélange de gaussiennes

Algorithme EM (*Estimation Maximization*)

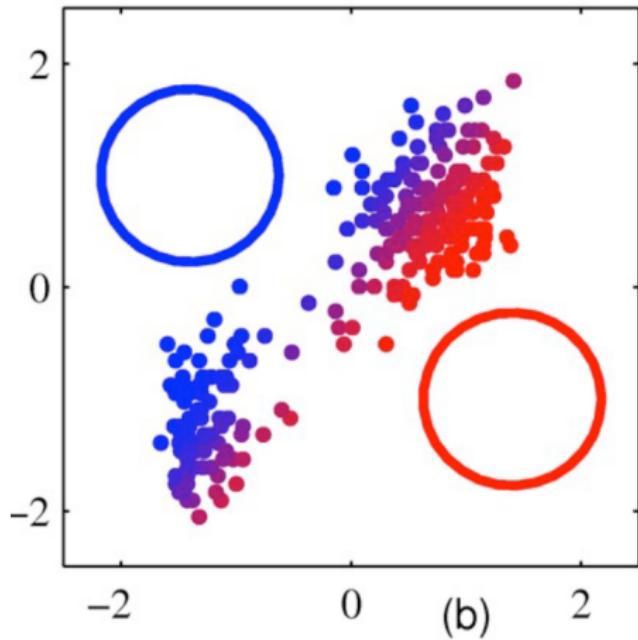
- ① On initialise les moyennes μ_j , les covariances Σ_j et les coefficients π_j . On évalue la valeur initial de la (log)-vraisemblance
- ② On calcule les $\gamma(z_{ij})$
- ③ On restime les moyennes μ_j , les covariances Σ_j et les coefficients π_j
- ④ On calcule la (log)-vraisemblance en vérifiant sa convergence. Si la convergence n'est pas encore atteinte, retourne au pas 2.

Exemple d'exécution



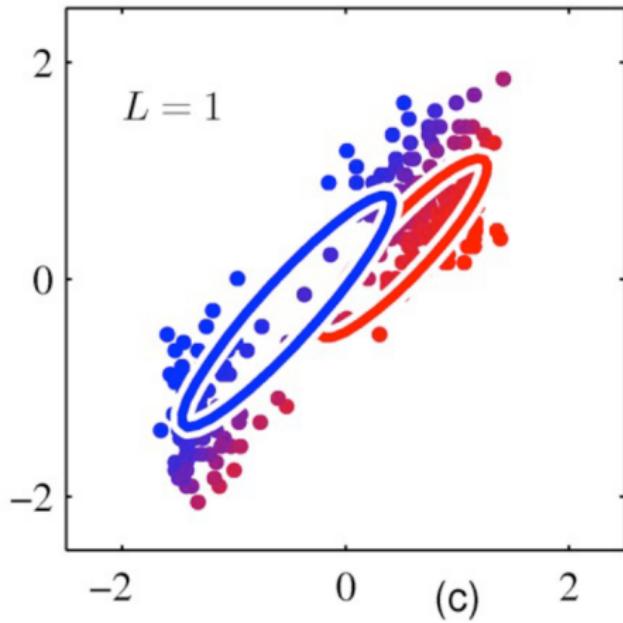
source : Larochelle, 2015

Exemple d'exécution



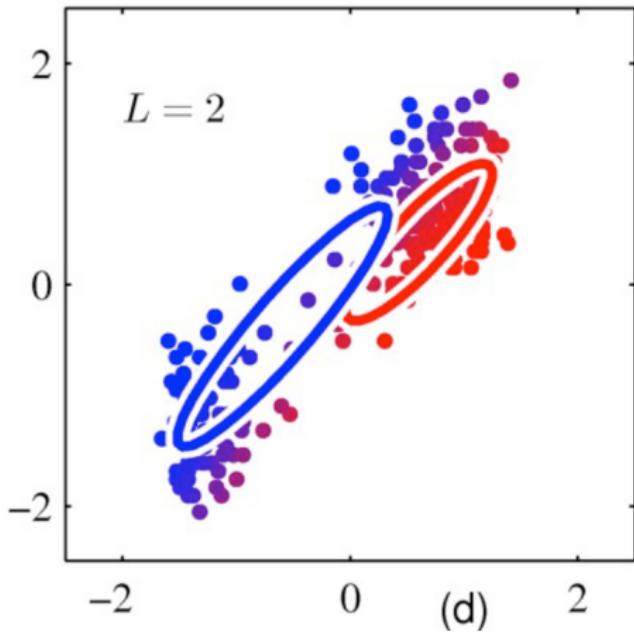
source : Larochelle, 2015

Exemple d'exécution



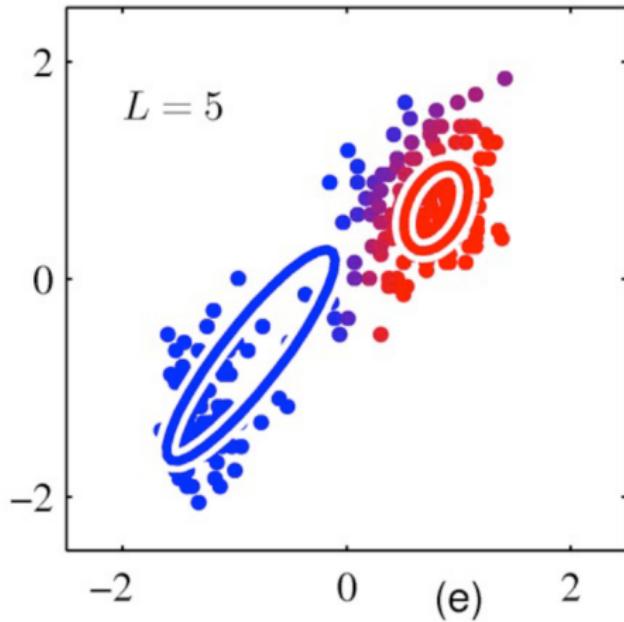
source : Larochelle, 2015

Exemple d'exécution



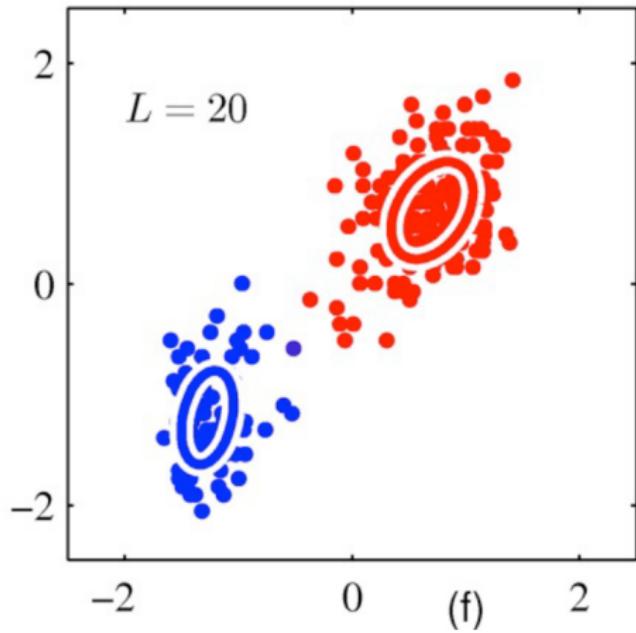
source : Larochelle, 2015

Exemple d'exécution



source : Larochelle, 2015

Exemple d'exécution



source : Larochelle, 2015

Choix de l'algorithme

- Si vous êtes intéressé à une hiérarchie, utilisez un **algorithme hiérarchique**
- Si vous êtes intéressés à une **partition** :
 - Choisissez parmi les options disponibles d'algorithmes
 - Analysez les résultats obtenus (e.g. visualisation)
- Si vous êtes intéressés à savoir les probabilités d'appartenance de vos données aux différents clusters, utilisez un **modèle léger**

Évaluation de clusters

- Mesures internes :
 - Liées au critère de clustering choisi
 - A maximiser pour les critères de séparation / minimiser pour les critères d'homogénéité
- Mesures externes
 - Utilisent une partition **ground-truth** pour calculer des mesures tel quelles le F-score, recall, etc (attention à la symétrie !)

Évaluation de clusters

- Mesures internes :
 - Liées au critère de clustering choisi
 - A maximiser pour les critères de séparation / minimiser pour les critères d'homogénéité
- Mesures externes
 - Utilisent une partition **ground-truth** pour calculer des mesures tel quelles le F-score, recall, etc (attention à la symétrie !)
 - Basée sur une connaissance *a priori*

Évaluation de clusters

- Mesures internes :
 - Liées au critère de clustering choisi
 - A maximiser pour les critères de séparation / minimiser pour les critères d'homogénéité
- Mesures externes
 - Utilisent une partition **ground-truth** pour calculer des mesures tel quelles le F-score, recall, etc (attention à la symétrie !)
 - Basée sur une connaissance *a priori*
 - Critique : pourquoi donc utiliser l'apprentissage non supervisé ?