

Reconnaissance des formes et méthodes neuronales (RCP208)

Classification automatique par densité

Nicolas Audebert

modifié par Michel Crucianu

(prenom.nom@cnam.fr)

<http://cedric.cnam.fr/vertigo/Cours/ml/>

EPN05 Informatique

Conservatoire National des Arts & Métiers, Paris, France

31 octobre 2024

Plan du cours

1 Motivation

2 DBSCAN

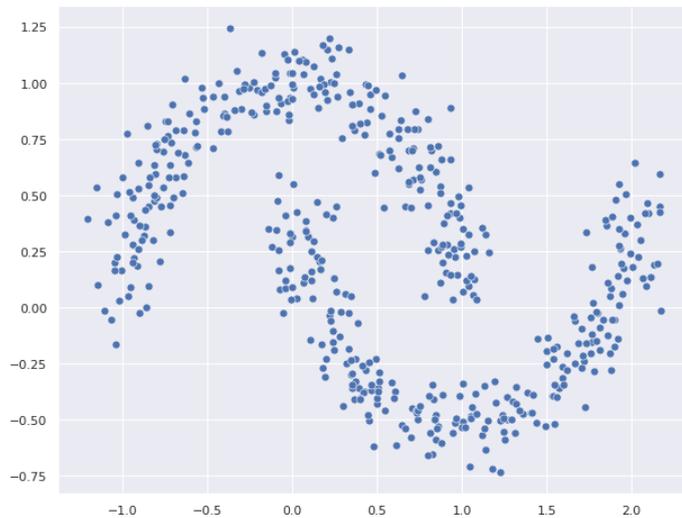
3 Paramétrage

4 Pour aller plus loin

Quand k -means ne suffit plus...

Considérons un jeu de données $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ de N observations bi-dimensionnelles

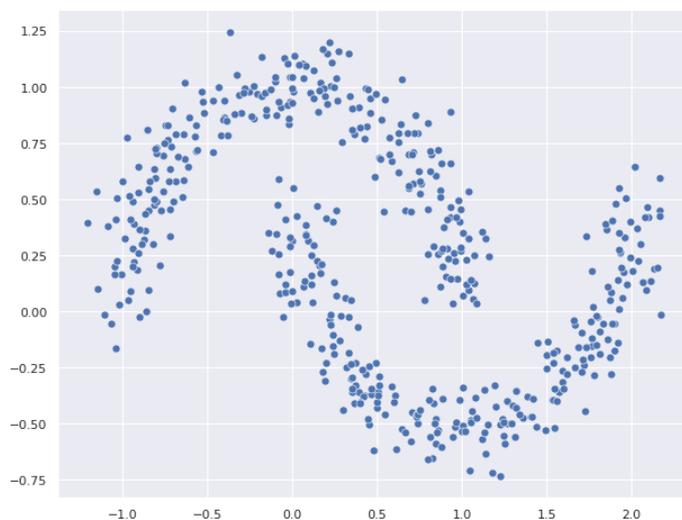
Comment partitionner ce jeu de données ?



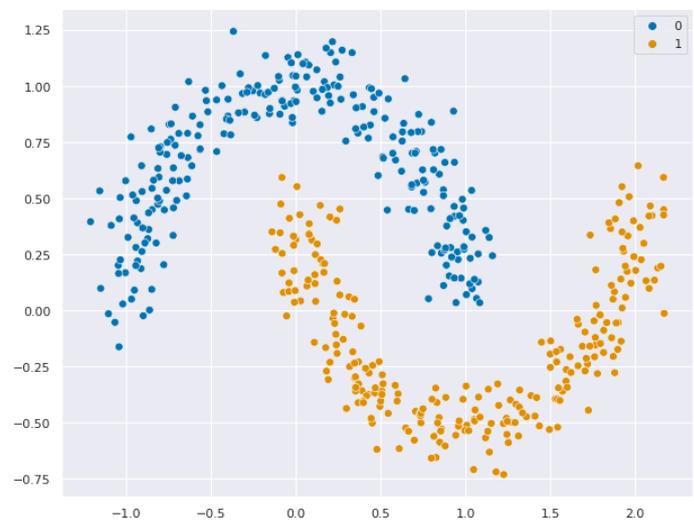
Quand k -means ne suffit plus...

Considérons un jeu de données $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ de N observations bi-dimensionnelles

Comment partitionner ce jeu de données ?



Une proposition « naturelle »

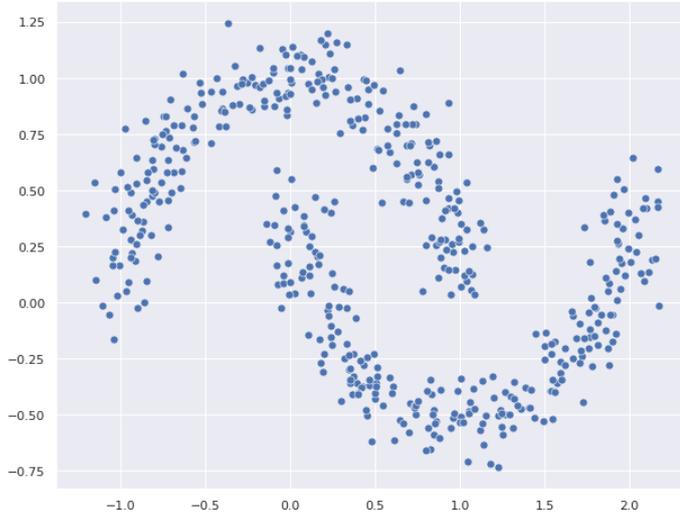


Quand k -means ne suffit plus...

Considérons un jeu de données $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ de N observations bi-dimensionnelles

Comment partitionner ce jeu de données ?

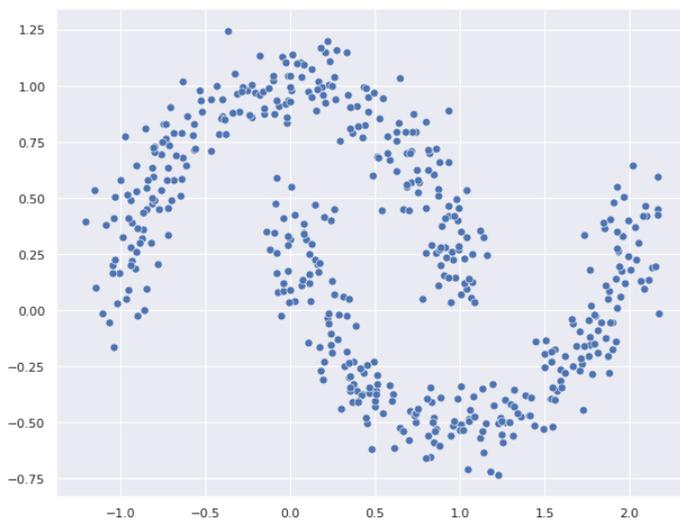
Une proposition « naturelle »



Le partitionnement proposé ci-dessus semble être évident mais peut-on l'obtenir automatiquement ?

Hypothèses de k -means

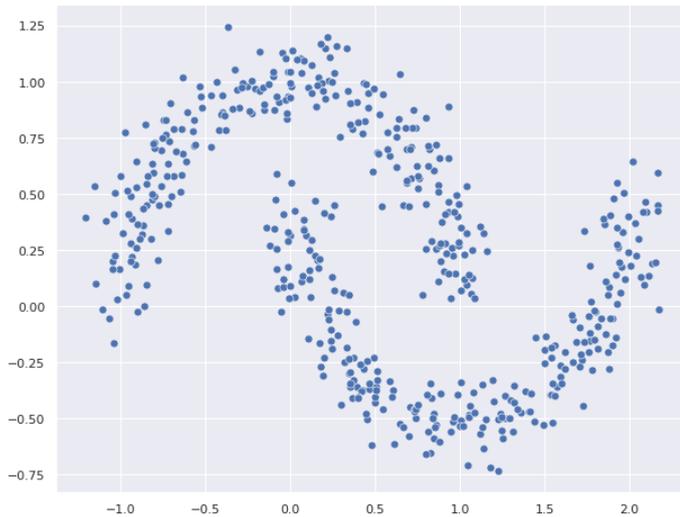
- Clusters symétriques = il n'y a pas de direction privilégiée dans un groupe



Jeu de données $\mathcal{X} \subset \mathbb{R}^2$

Hypothèses de k -means

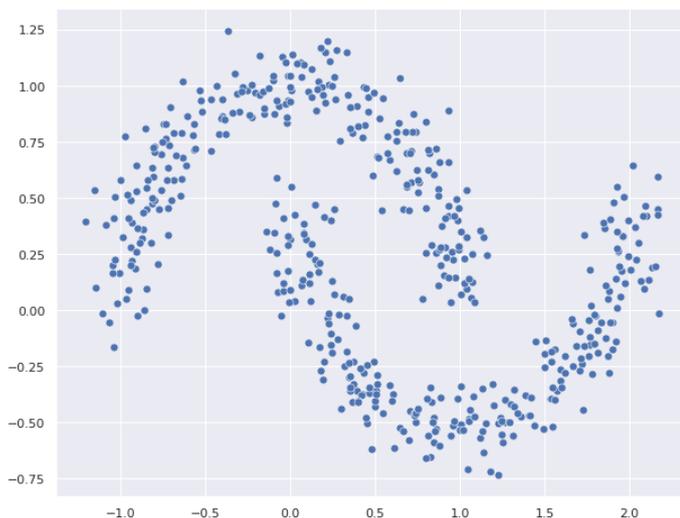
- Clusters symétriques = il n'y pas de direction privilégiée dans un groupe
- Compacts = les observations sont proches du centre de leur groupe



Jeu de données $\mathcal{X} \subset \mathbb{R}^2$

Hypothèses de k -means

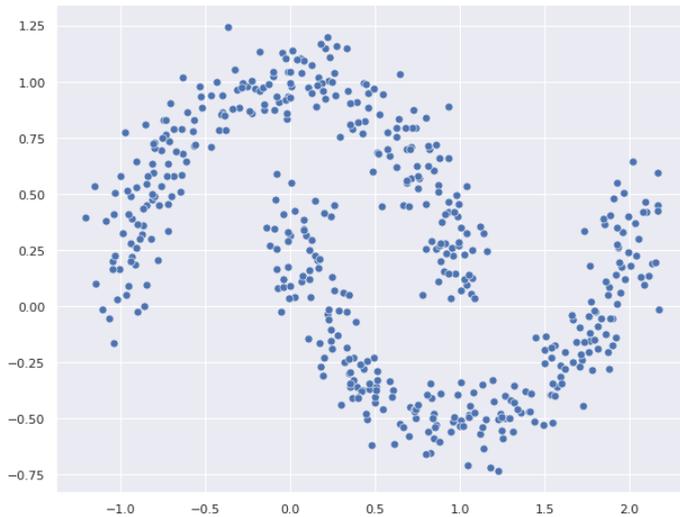
- Clusters symétriques = il n'y pas de direction privilégiée dans un groupe
- Compacts = les observations sont proches du centre de leur groupe
- Convexes = il n'y a pas de « trou » dans les groupes



Jeu de données $\mathcal{X} \subset \mathbb{R}^2$

Hypothèses de k -means

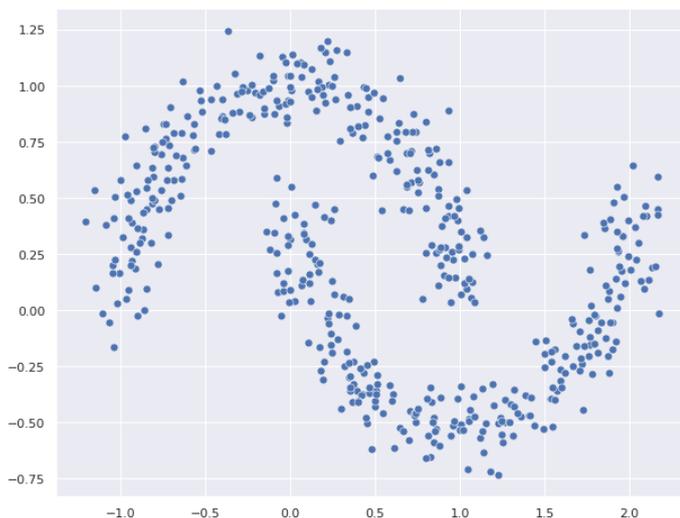
- Clusters symétriques = il n'y pas de direction privilégiée dans un groupe
- Compacts = les observations sont proches du centre de leur groupe
- Convexes = il n'y a pas de « trou » dans les groupes
- Linéairement séparables (deux à deux)



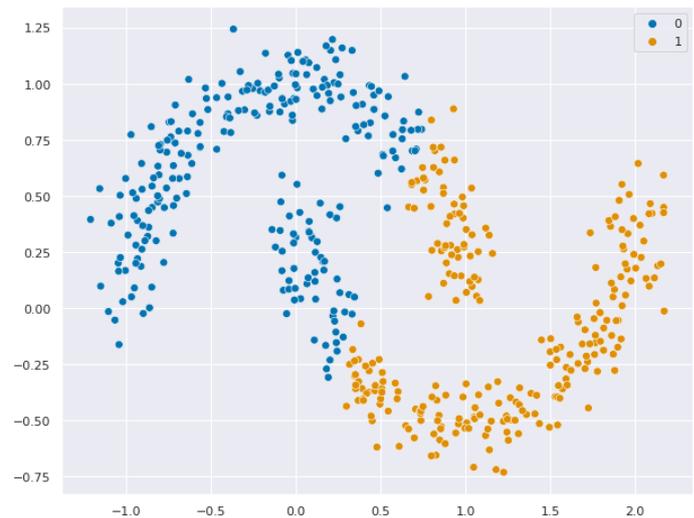
Jeu de données $\mathcal{X} \subset \mathbb{R}^2$

Hypothèses de k -means

- Clusters symétriques = il n'y pas de direction privilégiée dans un groupe
- Compacts = les observations sont proches du centre de leur groupe
- Convexes = il n'y a pas de « trou » dans les groupes
- Linéairement séparables (deux à deux)



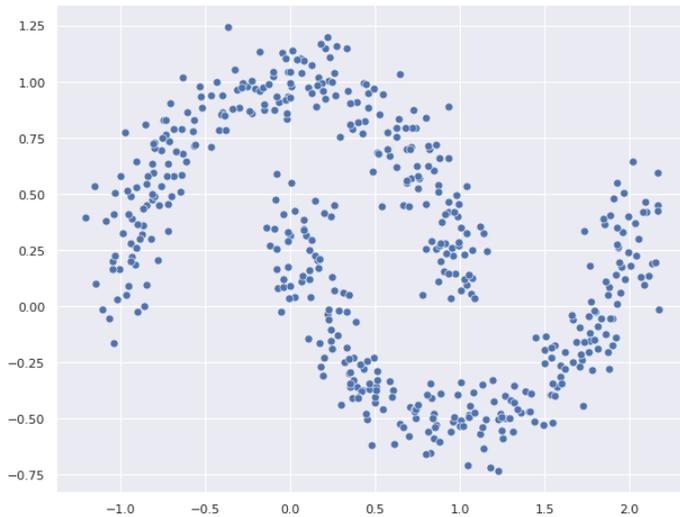
Jeu de données $\mathcal{X} \subset \mathbb{R}^2$



Application d'un k -means ($k = 2$)

Hypothèses de k -means

- Clusters symétriques = il n'y a pas de direction privilégiée dans un groupe
- Compacts = les observations sont proches du centre de leur groupe
- Convexes = il n'y a pas de « trou » dans les groupes
- Linéairement séparables (deux à deux)



Jeu de données $\mathcal{X} \subset \mathbb{R}^2$



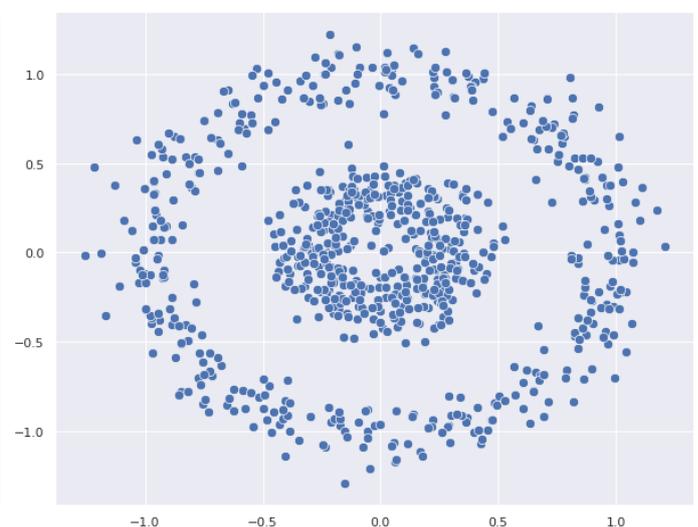
Application d'un k -means ($k = 2$)

Échec...

Comment faire mieux ?

Ce qui pose problème :

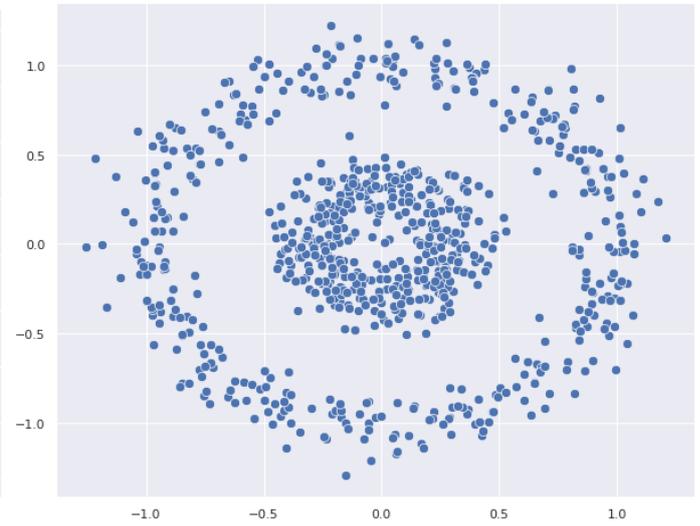
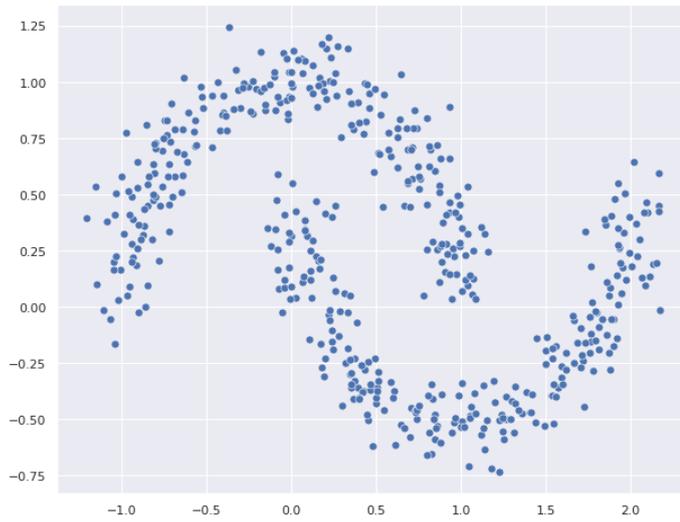
- la non-convexité



Comment faire mieux ?

Ce qui pose problème :

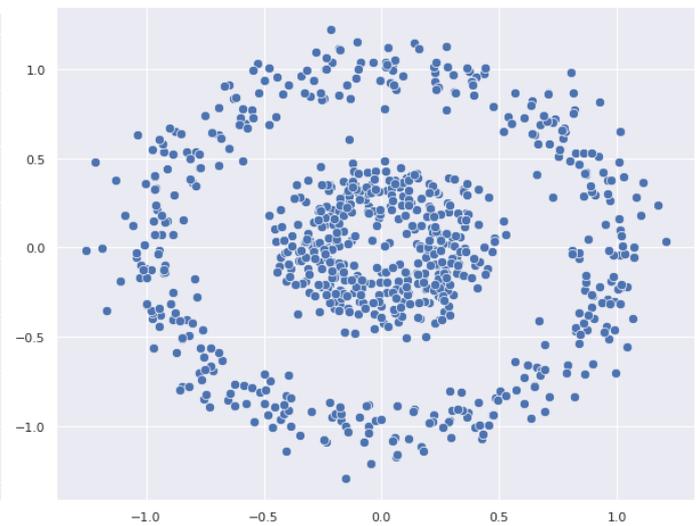
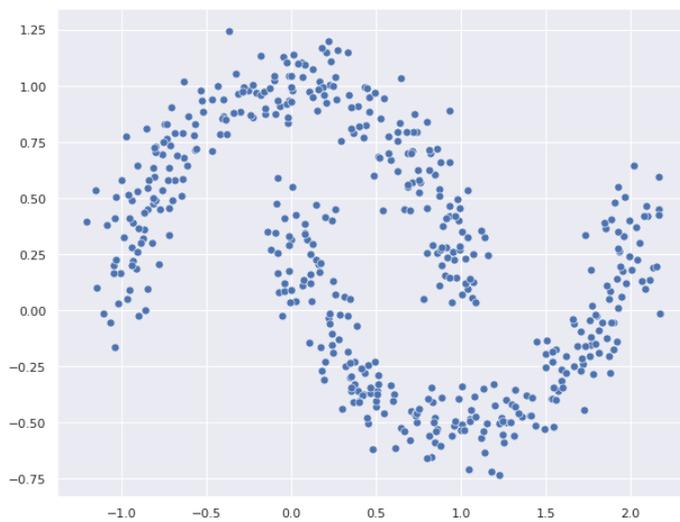
- la non-convexité
- la non-linéarité



Comment faire mieux ?

Ce qui pose problème :

- la non-convexité
- la non-linéarité



Comment sait-on que deux points appartiennent au même groupe dans ces exemples ?

Plan du cours

- 1 Motivation
- 2 DBSCAN
- 3 Paramétrage
- 4 Pour aller plus loin

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN est un algorithme de classification automatique (ou *partitionnement*) par densité introduit par [3]

Hypothèses de travail

- les groupes sont des îlots à forte densité
 - beaucoup d'observations dans un petit espace
- séparés par des océans à faible densité
 - peu de points dans un grand espace

Les données aberrantes

Les observations dans les zones à faible densité sont accidentelles (données aberrantes ou *outliers* ou « bruit »)

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN est un algorithme de classification automatique (ou *partitionnement*) par densité introduit par [3]

Hypothèses de travail

- les groupes sont des îlots à forte densité
 - beaucoup d'observations dans un petit espace
- séparés par des océans à faible densité
 - peu de points dans un grand espace

Les données aberrantes

Les observations dans les zones à faible densité sont accidentelles (données aberrantes ou *outliers* ou « bruit »)

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN est un algorithme de classification automatique (ou *partitionnement*) par densité introduit par [3]

Hypothèses de travail

- les groupes sont des îlots à forte densité
 - beaucoup d'observations dans un petit espace
- séparés par des océans à faible densité
 - peu de points dans un grand espace

Les données aberrantes

Les observations dans les zones à faible densité sont accidentelles (données aberrantes ou *outliers* ou « bruit »)

Taxonomie des observations selon DBSCAN

DBSCAN considère trois types de points selon les propriétés de leur voisinage :

Les points centraux (*core points*)

- ce sont les points qui se trouvent au cœur d'un groupe
- leur voisinage doit contenir plus que *minPts* points

Les points frontière (*border points*)

- ce sont les points qui bordent un groupe
- ils sont voisins d'un point central mais ne sont pas centraux eux-mêmes

Les points aberrants ou isolés ou « bruit » (*noise points*)

- ce sont les points isolés
- ils ne sont ni centraux, ni frontière

Taxonomie des observations selon DBSCAN

DBSCAN considère trois types de points selon les propriétés de leur voisinage :

Les points centraux (*core points*)

- ce sont les points qui se trouvent au cœur d'un groupe
- leur voisinage doit contenir plus que *minPts* points

Les points frontière (*border points*)

- ce sont les points qui bordent un groupe
- ils sont voisins d'un point central mais ne sont pas centraux eux-mêmes

Les points aberrants ou isolés ou « bruit » (*noise points*)

- ce sont les points isolés
- ils ne sont ni centraux, ni frontière

Taxonomie des observations selon DBSCAN

DBSCAN considère trois types de points selon les propriétés de leur voisinage :

Les points centraux (*core points*)

- ce sont les points qui se trouvent au cœur d'un groupe
- leur voisinage doit contenir plus que *minPts* points

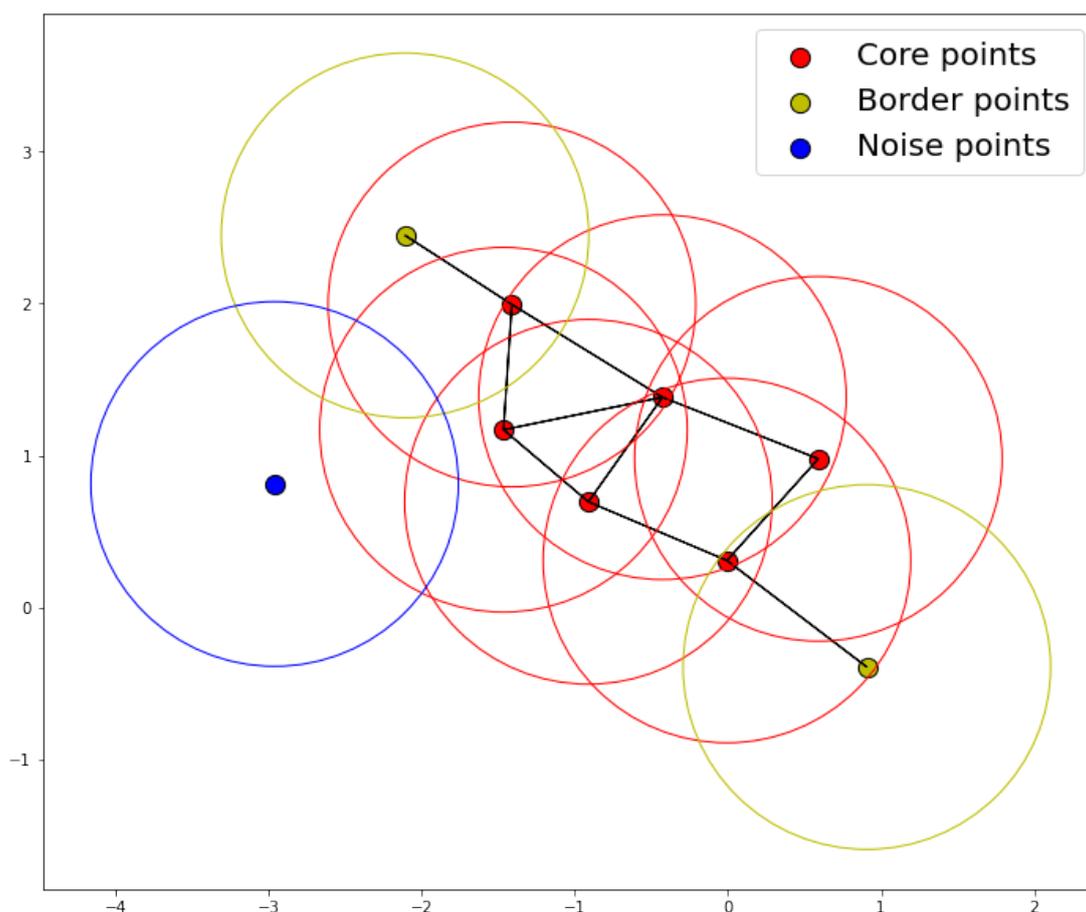
Les points frontière (*border points*)

- ce sont les points qui bordent un groupe
- ils sont voisins d'un point central mais ne sont pas centraux eux-mêmes

Les points aberrants ou isolés ou « bruit » (*noise points*)

- ce sont les points isolés
- ils ne sont ni centraux, ni frontière

Illustration des différents types de points



Algorithme

- 1 Choisir un point $\mathbf{x} \in \mathcal{D}$ pas encore visité
- 2 Marquer \mathbf{x} comme visité
- 3 Trouver le ε -voisinage de \mathbf{x} et mettre les voisins dans une liste
- 4 Si le voisinage est m -dense (si le nombre de voisins est supérieur à m) :
 - assigner \mathbf{x} à un nouveau cluster
 - pour chaque voisin \mathbf{x}' de \mathbf{x}
 - marquer \mathbf{x}' comme visité
 - si le voisinage de \mathbf{x}' est dense, ajouter ses voisins à la liste
 - si \mathbf{x}' n'a pas encore de groupe, l'ajouter au cluster
- 5 Sinon, marquer \mathbf{x} comme aberrant
- 6 Retour à 1 tant que tous les points n'ont pas été visités

Aspects théoriques

Quelles sont les propriétés qui permettent à cet algorithme de réaliser un partitionnement par densité ?

Soit un espace métrique (\mathcal{X}, d) , par exemple un espace vectoriel muni de la distance euclidienne, et $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathcal{X}$ un jeu de données de N d'observations

Définition

On appelle ε -voisinage de \mathbf{x} le sous-ensemble $V_\varepsilon(\mathbf{x}) \subset \mathcal{D}$ tel que :

$$V_\varepsilon(\mathbf{x}) = \{\mathbf{x}' \in \mathcal{D} \mid d(\mathbf{x}, \mathbf{x}') < \varepsilon\}$$

c'est-à-dire l'ensemble des observations qui sont à distance inférieure à ε de \mathbf{x} .

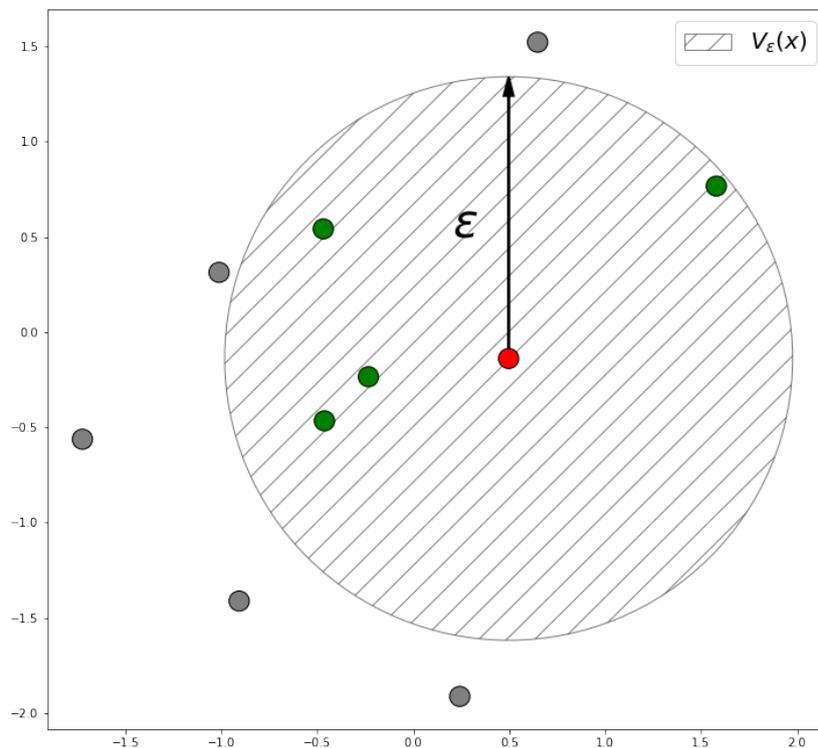
Autrement dit, ce sont les observations du jeu de données contenues dans la boule (ouverte) de rayon ε centrée sur \mathbf{x} .

m -densité

Ce voisinage est m -dense s'il contient au moins m points.

Exemple

L' ε -voisinage ci-dessous est 5-dense : l'observation x (en rouge) possède 4 voisins (en vert) à distance inférieure à ε



Accessibilité par densité

Accessibilité et accessibilité directe

Soient x et x' deux observations d'un jeu de données $\mathcal{D} \subset \mathbb{R}^p$

x' est dit **directement accessible** par ε -densité depuis x si :

- le voisinage de x est m -dense ($|V_\varepsilon(x)| \geq m$),
- x' est dans le voisinage de x ($x' \in V_\varepsilon(x)$).

En généralisant, x' est **accessible** par ε -densité depuis x si il existe une suite (y_1, \dots, y_n) telle que :

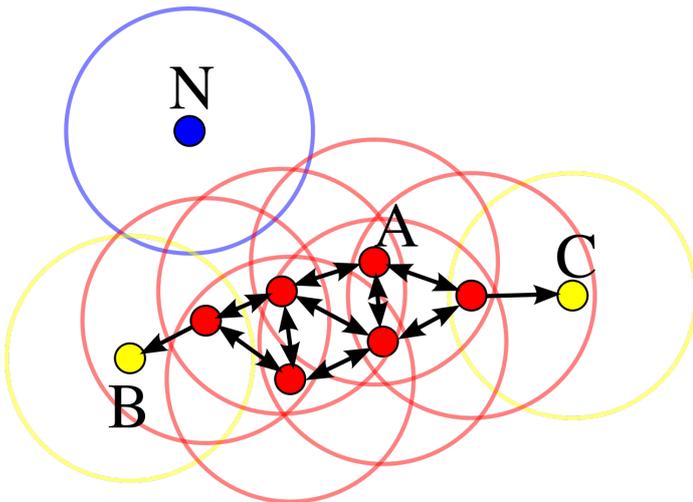
- $y_1 = x$ et $y_n = x'$,
- pour tout i , y_{i+1} est *directement accessible* depuis y_i .

DBSCAN et le graphe d'accessibilité

Graphe d'accessibilité

Construction du graphe d'accessibilité :

- 1 les nœuds du graphe sont les points $x \in \mathcal{D}$
- 2 x et x' sont reliés par une arête orientée si x' est directement accessible depuis x
 x' est accessible depuis x s'il existe un chemin permettant d'aller de x à x' dans le graphe d'accessibilité



Algorithme DBSCAN version graphe :

- 1 Identifier les points centraux
- 2 Calculer les composantes connectées du graphe réduit aux points centraux
- 3 Assigner chaque point frontière au cluster de son voisin le plus proche

Plan du cours

- 1 Motivation
- 2 DBSCAN
- 3 Paramétrage
- 4 Pour aller plus loin

Quels paramètres pour DBSCAN ?

L'exécution de l'algorithme DBSCAN nécessite de régler deux paramètres :

- ε , la taille du voisinage et donc le rayon de la boule dans laquelle on peut passer d'un point à un autre sans changer de groupe
- m , le nombre minimum de voisins pour qu'un voisinage soit qualifié de dense

ε

- Si ε est trop faible, alors aucune observation n'est voisine d'aucune autre \implies uniquement des points aberrants
- Si ε est trop grand, trop de points sont voisins entre eux \implies un seul groupe qui recouvre toutes les observations

m (ou minPts)

- Si m est faible, tous les voisinages sont denses : il suffit d'un seul voisin en commun pour relier deux groupes
- Si m est grand, peu de voisinages sont denses : beaucoup de points seront points frontière ou aberrants

m (ou minPts)

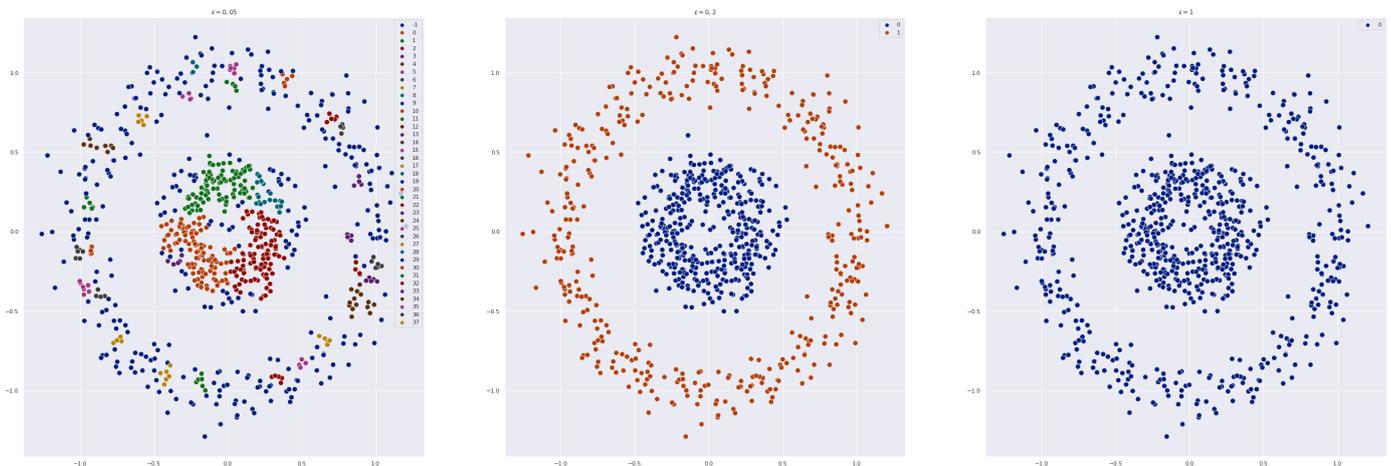
Quel nombre de voisins minimum pour définir un voisinage dense ?

Heuristiques pour le réglage de m

- $m = 2 \rightarrow$ classification ascendante hiérarchique
- A. par défaut, $m = 4$ ou $m = 5$ dans la plupart des implémentations...
- B. $[3, 5]$ recommandent $m = 2 \cdot p$ avec p la dimension des données
- C. si ε déjà choisi, choisir m comme le nombre moyen de points dans les ε -voisinages :

$$m = \frac{1}{N} \sum_{i=1}^N |V_{\varepsilon}(\mathbf{x}_i)|$$

ε : la taille du voisinage

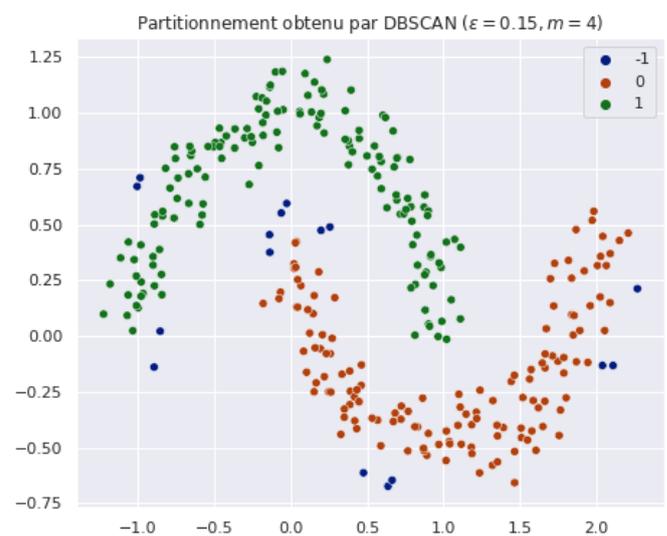
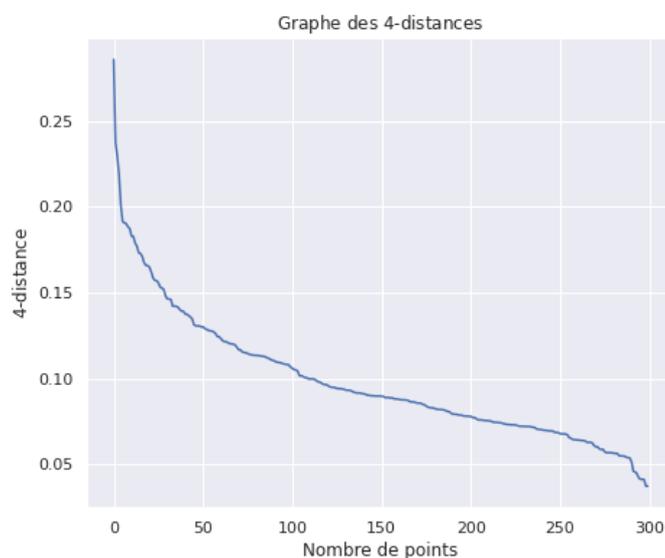


- ε trop faible : nombreux groupes de petite taille
 - ε trop élevé : un unique groupe
- ⇒ peut-on régler (semi-)automatiquement une valeur raisonnable pour ε ?

ε : choix heuristique

Graphe des k -distances [6, 4]

- pour chaque point, quelle la distance à son $k^{\text{ème}}$ voisin ?
- tracer le graphe des k -distances triées en ordre décroissant (avec $k = m - 1$)



Heuristique : choisir ε en considérant seulement le sous-ensemble des voisins réels

→ ordonnée du point de rupture de pente (« coude »)

Plan du cours

- 1 Motivation
- 2 DBSCAN
- 3 Paramétrage
- 4 Pour aller plus loin

Intérêts de DBSCAN

- Robustesse aux données aberrantes
 - DBSCAN identifie automatiquement et retire les données aberrantes durant le partitionnement. Cela permet de détecter les *outliers* mais également de ne pas contaminer la classification automatique (*k-means* est particulièrement sensible aux données aberrantes).
- Les groupes obtenus par DBSCAN ne sont pas nécessairement linéairement séparables
 - Réduit la contrainte sur la forme des groupes obtenus. Les groupes non convexes ne sont plus sur-partitionnés comme c'est le cas avec *k-means*.
- DBSCAN ne nécessite pas de préciser *a priori* le nombre de groupes recherchés.
 - Le nombre de groupes est estimé automatiquement à partir du nombre de composantes connectées par le graphe de l'accessibilité par densité.

Intérêts de DBSCAN

- Robustesse aux données aberrantes
 - DBSCAN identifie automatiquement et retire les données aberrantes durant le partitionnement. Cela permet de détecter les *outliers* mais également de ne pas contaminer la classification automatique (*k-means* est particulièrement sensible aux données aberrantes).
- Les groupes obtenus par DBSCAN ne sont pas nécessairement linéairement séparables
 - Réduit la contrainte sur la forme des groupes obtenus. Les groupes non convexes ne sont plus sur-partitionnés comme c'est le cas avec *k-means*.
- DBSCAN ne nécessite pas de préciser *a priori* le nombre de groupes cherchés.
 - Le nombre de groupes est estimé automatiquement à partir du nombre de composantes connectées par le graphe de l'accessibilité par densité.

Intérêts de DBSCAN

- Robustesse aux données aberrantes
 - DBSCAN identifie automatiquement et retire les données aberrantes durant le partitionnement. Cela permet de détecter les *outliers* mais également de ne pas contaminer la classification automatique (*k-means* est particulièrement sensible aux données aberrantes).
- Les groupes obtenus par DBSCAN ne sont pas nécessairement linéairement séparables
 - Réduit la contrainte sur la forme des groupes obtenus. Les groupes non convexes ne sont plus sur-partitionnés comme c'est le cas avec *k-means*.
- DBSCAN ne nécessite pas de préciser *a priori* le nombre de groupes cherchés.
 - Le nombre de groupes est estimé automatiquement à partir du nombre de composantes connectées par le graphe de l'accessibilité par densité.

Intérêts de DBSCAN

- Robustesse aux données aberrantes
 - DBSCAN identifie automatiquement et retire les données aberrantes durant le partitionnement. Cela permet de détecter les *outliers* mais également de ne pas contaminer la classification automatique (*k-means* est particulièrement sensible aux données aberrantes).
- Les groupes obtenus par DBSCAN ne sont pas nécessairement linéairement séparables
 - Réduit la contrainte sur la forme des groupes obtenus. Les groupes non convexes ne sont plus sur-partitionnés comme c'est le cas avec *k-means*.
- DBSCAN ne nécessite pas de préciser *a priori* le nombre de groupes cherchés.
 - Le nombre de groupes est estimé automatiquement à partir du nombre de composantes connectées par le graphe de l'accessibilité par densité.

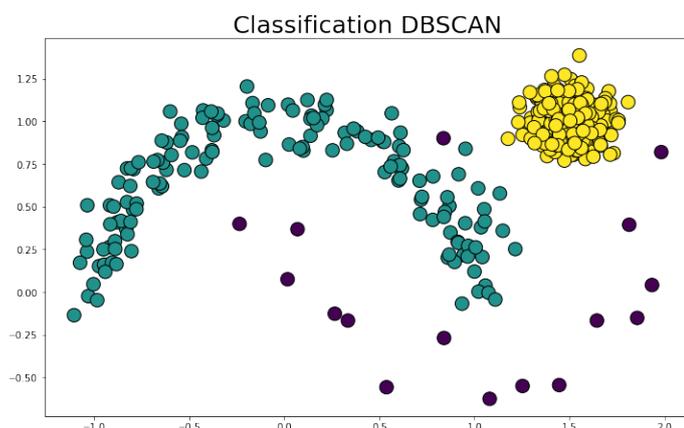
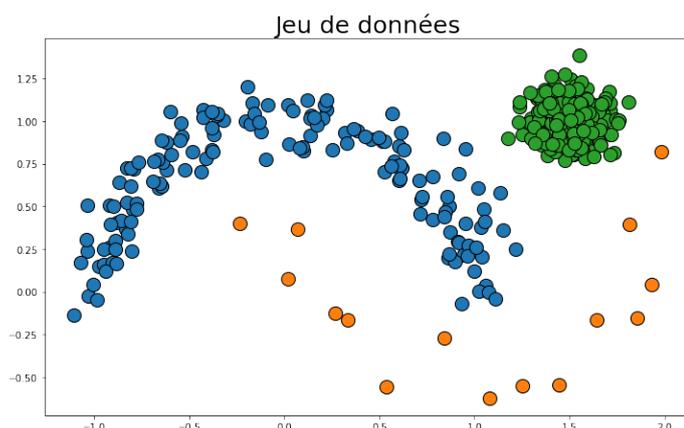
Intérêts de DBSCAN

- Robustesse aux données aberrantes
 - DBSCAN identifie automatiquement et retire les données aberrantes durant le partitionnement. Cela permet de détecter les *outliers* mais également de ne pas contaminer la classification automatique (*k-means* est particulièrement sensible aux données aberrantes).
- Les groupes obtenus par DBSCAN ne sont pas nécessairement linéairement séparables
 - Réduit la contrainte sur la forme des groupes obtenus. Les groupes non convexes ne sont plus sur-partitionnés comme c'est le cas avec *k-means*.
- DBSCAN ne nécessite pas de préciser *a priori* le nombre de groupes cherchés.
 - Le nombre de groupes est estimé automatiquement à partir du nombre de composantes connectées par le graphe de l'accessibilité par densité.

Intérêts de DBSCAN

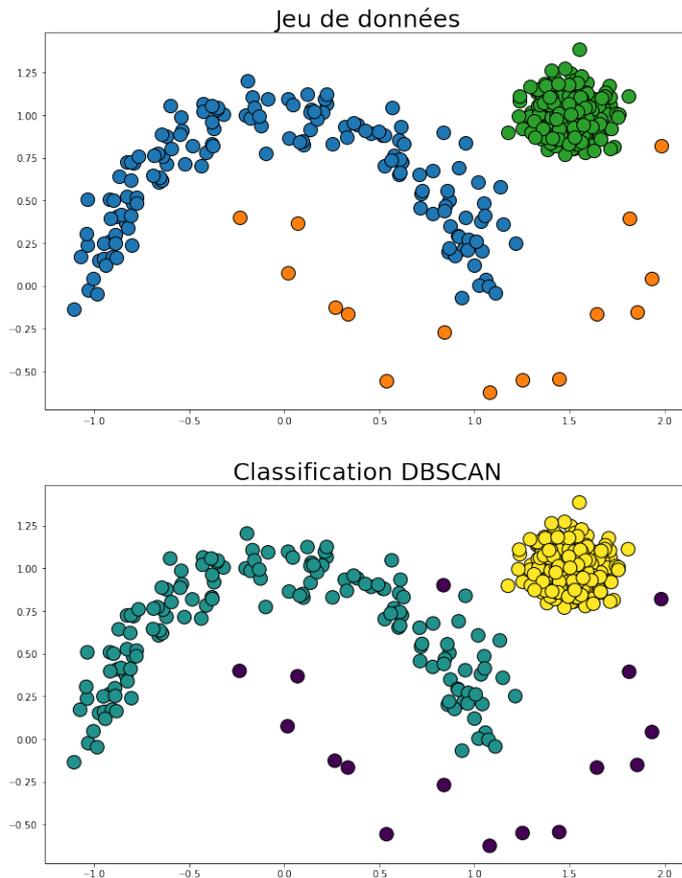
- Robustesse aux données aberrantes
 - DBSCAN identifie automatiquement et retire les données aberrantes durant le partitionnement. Cela permet de détecter les *outliers* mais également de ne pas contaminer la classification automatique (*k-means* est particulièrement sensible aux données aberrantes).
- Les groupes obtenus par DBSCAN ne sont pas nécessairement linéairement séparables
 - Réduit la contrainte sur la forme des groupes obtenus. Les groupes non convexes ne sont plus sur-partitionnés comme c'est le cas avec *k-means*.
- DBSCAN ne nécessite pas de préciser *a priori* le nombre de groupes cherchés.
 - Le nombre de groupes est estimé automatiquement à partir du nombre de composantes connectées par le graphe de l'accessibilité par densité.

Limites de DBSCAN



- DBSCAN considère que la densité est identique pour tous les groupes \implies impossible de trouver un unique seuil ϵ qui définit un voisinage adapté en cas de densité **variable**
- Les données dans des régions à faible densité sont automatiquement éliminées en tant que données aberrantes, or elles ne sont pas toujours aberrantes...

Limites de DBSCAN



- DBSCAN considère que la densité est identique pour tous les groupes \implies impossible de trouver un unique seuil ϵ qui définit un voisinage adapté en cas de densité **variable**
- Les données dans des régions à faible densité sont automatiquement éliminées en tant que données aberrantes, or elles ne sont pas toujours aberrantes...

Pour aller plus loin

- OPTICS [1]
 - Évolution de DBSCAN qui considère une plage de valeurs possibles pour ϵ
 - Permet de détecter des groupes de densités différentes
 - Partitionnement hiérarchique
 - Disponible dans `scikit-learn`
- HDBSCAN [2]
 - Variante proche d'OPTICS mais diffère dans son choix de sélection des groupes
 - Autorise la classification de points nouveaux
 - `pip install hdbscan`

Références I

-  M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander.
OPTICS : ordering points to identify the clustering structure.
28(2) :49–60.
-  R. J. G. B. Campello, D. Moulavi, and J. Sander.
Density-based clustering based on hierarchical density estimates.
In J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, pages 160–172. Springer.
-  M. Ester, H.-P. Kriegel, J. Sander, and X. Xu.
A density-based algorithm for discovering clusters in large spatial databases with noise.
In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press.
-  H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek.
Density-based clustering.
1(3) :231–240.
-  J. Sander, M. Ester, H.-P. Kriegel, and X. Xu.
Density-based clustering in spatial databases : The algorithm GDBSCAN and its applications.
2(2) :169–194.

Références II

-  E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu.
DBSCAN revisited, revisited : Why and how you should (still) use DBSCAN.
42(3) :19 :1–19 :21.