Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Licence en Informatique

Thème

Conception et implémentation d'un algorithme de Clustering à la base de DBSCAN

Réalisé par :

- M^elle^ *LOUATI Aicha*
- M^elle^ *MESROUA Ikram Fatima Zohra*

Présenté le 8 Juin 2014 devant la commission d'examination composée de MM.

- *M^me^ Berramdane.D* *(Examinatrice)*
- *M^r^ HADJILA.F* *(Encadreur)*
- *M^r^ LAHASAINI.M* *(Examinateur)*
- *M^r^ MOUAFEK.B* *(Examinateur)*

Année universitaire : 2013-2014

Remerciement :

Nous tenons tout d'abord à remercier Allah le tout puissant et miséricordieux, lequel ordonna à travers le premier verset coranique à l'humanité d'étudier, Il nous a donné force et patience afin d'accomplir ce Modeste travail.

En second lieu, nous tenons à remercier notre encadreur

Mr F .Hadjila pour l'orientation, la confiance, la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Dédicace :

Nous dédions ce modeste travail à celles qui nous ont donné la vie, le symbole de tendresse, qui se sont sacrifiées pour nos bonheurs et nos réussites, à nos mères,
A nos pères, école de notre enfance, qui ont été nos ombres durant toutes les années des études, et qui ont veillé tout au long de nos vies à nos encourager, à nous donner l'aide et à nous protéger.
Que dieu les gardes et les protège,

Aucune dédicace ne saurait exprimer nos grandes admirations, nos considérations et nos sincères affections pour vous.

LOUATI Aïcha :

A mes chers frères Mohammed, Ibrahim et ma très chère et douce sœur Zahira, son mari Ahmed et leur fille Rahaf Noor Elyakin, pour leur grande affection, compréhension, patience
et leur incontestable appui.
A qui je dois tout,
A tous ceux qui, par un mot, m'ont donné la force de continuer .....

MESROUA Ikram Fatéma Zohra :

A ma sœur unique Nadjlaa Amina avec laquelle je partage une immense complicité à travers laquelle une très forte affection est née entre nous et restera à jamais, rien au monde ne pourra nous séparer.

A mes chers amis Hichem et Ibtissem , ainsi qu'a la mémoire de sa mère qui vient de nous quitter récemment .

A Mon second Père Monsieur Boufatah.

A tous ceux qui nous sont chers,

Nous dédions ce travail.

Résumé : La classification automatique (Clustering), est la tâche qui consiste à diviser un jeu de données en sous-ensembles de données pour que tous les individus dans une même classe soient similaires et les individus de classes distinctes soient dissimilaires sans connaitre les classes à priori. Malgré le nombre important d'algorithmes de classification automatiques existants, plusieurs problématiques restent encore ouvertes dans le cadre de la classification. Les algorithmes de classification basée sur la densité sont particulièrement intéressants et répondent au mieux à ces problématiques et plus particulièrement l'algorithme DBSCAN qui a la capacité de découvrir des clusters de formes arbitraires et de détecter automatiquement le nombre de groupes dans les données tout en identifiant les bruits. L'objectif de ce travail est d'étudier les performances de cet algorithme qui est DBSCAN sur un jeu de données construits de façon aléatoire.
Mots-clés : Clustering, la classification basée sur la densité, DBSCAN.

Abstract : Clustering is the task of dividing large data sets into smaller subsets where all elements in the same subset share common parameters. Clustering should be done without knowing previous informations on the classes. Even if a large number of algorithms for data Clustering exist, several classification points still uncovered yet. The DBSCAN algorithm is one of the clustering approaches based on the data density criteria, which is particularly interesting and comes with the best results. This algorithm is able to determine automatically the data clusters, and the isolated points which leads to the identification of the noise (undesirable data) . The objective of this work is to study the performance of the DBSCAN algorithm by using a random data sets.
Keywords: Clustering, density based Clustering, DBSCAN.

ملخص: التصنيف التلقائي (المجموعات)، هو مهمة تعتمد على تقسيم مجموعة من البيانات إلى مجموعات فرعية من البيانات.جميع الأفراد الموجودة في نفس الفئة متشابهة، أما الأفراد من فئات مختلفة فهي فئات متباينة. بدون معرفة الفئات مسبقا على الرغم من وجود عدد كبير من الخوارزميات في التصنيف التلقائي، لا تزال هناك العديد من المشاكل المفتوحة في هذا المجال. خوارزميات التصنيف بالاعتماد على الكثافة هي مثيرة للاهتمام بصفة خاصة و تستجيب بشكل أفضل لهذه المشاكل وعلى وجه الخصوص الخوارزمية DBSCAN التي لديها القدرة على إيجاد مجموعات من مختلف الأشكال و الأحجام والكشف التلقائي على عدد المجموعات في البيانات مع تحديد الضوضاء و القيم الغير المطلوب تصنيفها.الهدف من هذا العمل هو دراسة أداء خوارزمية DBSCAN على مجموعة من البيانات التي تشكلت عشوائيا.
الكلمات المفتاحية : المجموعات، خوارزميات التصنيف بالاعتماد على الكثافة، DBSCAN.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1 :

Introduction Générale

“ La recherche procède par des moments distincts et durables, intuition, aveuglement, exaltation et fièvre. Elle aboutit un jour à cette joie, et connaît cette joie celui qui a vécu des moments singuliers "

Albert Einstein, " Comment je vois le monde"

Contexte :

Depuis des millénaires, la classification a suscité l'esprit de l'homme, elle lui permet de concevoir, comprendre et organiser sa vision du monde. Elle relie la science théorique à la pratique spontanée, par un ensemble de gestes précis et maîtrisés. L'homme a tenté d'organiser la nature en établissant des catégories, des groupes et des ordres.

Intronisée dans divers domaines, la classification apporte, cependant un éclairage sur la nature de la connaissance.

De l'astronomie à l'intelligence artificielle, son application demeure universelle :

- La classification de HARVARD est celle qui attribue un type spectral à une étoile, et correspond globalement à une échelle de température.
- La classification spectrale désigne une famille d'algorithmes de classification non supervisées.

Dans le cadre de l'Intelligence Artificielle(IA), la classification demeure l'un de ces champs d'étude et une discipline scientifique relevant du Datamining ,un domaine vaste et pluridisciplinaire qui englobe l'ensemble des techniques et méthodes permettant l'extraction, à partir d'un important volume de données brutes de connaissances originales auparavant inconnues, il s'agit de fouilles visant à découvrir l'information cachée que les données renferment et que l'on découvre à la recherche d'association de tendance ,de relation ou de régularité.

La classification non-supervisée « dénommée Clustering » peut être considérée comme l'étude la plus important du Datamining, et dont nous sommes concernées dans ce mémoire. Elle tente de trouver une structure dans un ensemble de données non étiquetées. Une définition vague de regroupement pourrait être « le processus d'organisation des objets dans des groupes dont les membres sont semblables d'une certaine façon ».

Un cluster est donc une collection d'objets qui sont « similaires » entre eux et sont « différents » pour les objets appartenant à d'autres groupes.

Il existe cependant une très large famille de méthodes dédiées à la classification non-supervisée qui différent par la stratégie mise en place pour construire les clusters. Parmi ces méthodes, il y a les méthodes hiérarchiques, les méthodes de partitionnement, les méthodes basées sur la densité et celles basées sur la grille.

Problématique :

On s'intéresse dans ce travail aux méthodes de Clustering permettant de repérer et d'isoler les bruits, lors de la classification, pour cela nous nous focalisons sur les méthodes à base de densité.

Dans ces types d'approches les classes sont considérées comme des régions en haute densité, qui sont séparées par des régions en faible densité.

La densité est représentée par le nombre d'individus de l'ensemble des données. C'est pourquoi ces méthodes sont capables de chercher des classes de tailles et formes arbitraires.

Contribution :

DBSCAN (Density Based Spatial Clustering of Applications with Noise), est l'un des algorithmes les plus populaires des méthodes à base de densité. Ce dernier présente l'intérêt de dégager les clusters ayant des formes variées et non prédéfinies à partir d'une base de données de grande dimension, tout en identifiant les groupes. Cet algorithme, en regroupant les points en fonction de leur densité, met ainsi en valeur des pôles d'équipement de même densité minimale.

L'objectif de notre mémoire est de concevoir et d'implémenter cet algorithme.

Plan de ce mémoire :

Le mémoire est organisé comme suit:

Le premier chapitre : débute par une introduction générale qui donne un aperçu globale de l'algorithme quand va traiter dans les chapitres qui vont suivre.

Le deuxième chapitre : nous nous intéresserons plus particulièrement à la classification non-supervisée « Clustering ». nous aborderons d'abord la définition d'un cluster et ses propriétés pour mieux apprivoiser par la suite les notions de base du Clustering, en examinant chacun des domaines qu'elle vise, ses étapes, les principales exigences, ses problèmes , …. , etc.) essentiels pour comprendre le fonctionnement de ce principe,

En second lieu nous présenterons les différentes approches de ce dernier comme suit :

Méthodes hiérarchiques :

- Ascendantes
- Descendantes

Méthodes non hiérarchique :

- Méthodes de partitionnement
- Méthodes basées sur la grille
- Méthodes basées sur la densité

En se focalisant plus particulièrement sur les méthodes basées sur la densité que compte l'algorithme DBSCAN, but de ce travail, qu'on abordera plus profondément dans le prochain chapitre.

On conclut ce chapitre par un tableau comparatif des différents algorithmes de ses méthodes.

Le troisième chapitre : Ce chapitre présente le cœur de ce mémoire : il est dédié à concevoir et implémenter DBSCAN**.** Nous commencerons par présenter quelques définitions formelles et essentielles pour comprendre DBSCAN. Ensuite, nous expliquerons le fonctionnement de DBSCAN.

Nous terminerons ce mémoire par différentes perspectives de recherche qui nous semblent intéressantes pour améliorer l'algorithme que nous avons traité dans notre travail.

Chapitre 2 :

Classification non-supervisée

« Classification automatique »

« Clustering »

« Groupement »

"Dessiner les phénomènes et ordonner en série les événements décisifs d'une expérience, voilà la tâche première où s'affirme l'esprit scientifique ; c'est en effet de cette manière qu'on arrive à la quantité figurée, à mi-chemin entre le concret et l'abstrait, dans une zone intermédiaire où l'esprit peut canaliser les mathématiques et l'expérience, les lois et les faits."

Gaston Bachelard.

Introduction :

Etant donné un ensemble d'observations, comment peut-on les regrouper, en un certain nombre de grappes ″ clusters ″, ″ groupes ″ de façon à ce que les clusters obtenus soient constitués d'observations semblables, et que ces clusters soient les plus différents possible entre eux.

C'est la réponse à cette question que veulent fournir les méthodes de classification automatique. La diversité de ses méthodes est déconcertante au premier abord. Des volumes entiers ont étés publiés sur ce sujet. Nous nous bornerons ici, à énoncer les grands principes qui sous-entendent toutes ces méthodes.

Dès le départ, il est nécessaire de différencier la classification non-supervisée de la classification supervisée ou analyse discriminante. La classification supervisée consiste à construire des règles de décision en se basant sur un ensemble de données pour lesquelles les étiquettes des classes sont connues a priori, par contre, le but de la classification non-supervisée (Clustering en anglais) est de trouver une organisation des données cohérentes et valides, qui puissent mettre en évidence les vraies structures dans un ensemble de données sans aucune connaissance a priori, sur les données traitées [3].

Dans beaucoup d'applications on ne dispose pas des connaissances a priori sur les données et donc, les méthodes de classification doivent faire le moins de suppositions. C'est grâce à ces restrictions que les méthodes de classification non supervisée sont particulièrement appropriées pour explorer les relations entre les données et pour offrir une vision cohérente sur leur vraie structure.

Avant de présenter les principes du processus du Clustering nous introduisons d'abord la notion de grappe (cluster).

2.1 Cluster :

2.1.1 Définition d'un cluster (une grappe) :

Il n'existe pas de définition universellement acceptée de ce qu'est exactement un cluster [2]. En conséquence, les méthodes de Clustering ne sont généralement pas des solutions identiques, voire similaires.

Un cluster est généralement considéré comme un groupe d'éléments (objets, points) où chacun est « proche » (dans un sens approprié) à un élément central (appartenant au même cluster) et que les membres des différents groupes sont « loins » les uns des autres. Dans un autre sens, les clusters peuvent être considérés comme des « régions à haute densité » de

certains espaces multidimensionnels (Hartigan, 1975 via [2]). Une telle notion est appropriée pour des clusters de forme convexe.

Toutefois, il n'est pas difficile de concevoir des situations dans lesquelles le regroupement des éléments dans la nature ne suit pas le modèle de Clustering. Par exemple: lorsque la dimension de l'espace traité est assez grande, les éléments multidimensionnels, considérés comme points de cet espace, peuvent se rassembler dans des clusters qui se courbent les uns autour des autres. Même, si ces essaims ne sont pas empiétants (ce qui est peu probable dans la nature), les configurations de forme étrange sont presque impossibles à détecter et à identifier à l'aide des techniques actuelles [2], cette question a été largement étudiée par Brucker [6] et Cleuziou [9].

2.1.2 Propriétés d'un cluster :

Les deux propriétés importantes définissant un cluster pertinent sont :

– Sa cohésion interne (que les objets appartenant à ce cluster soient les plus similaires possibles)

– Son isolation externe (que les objets appartenant aux autres clusters soient les plus éloignés possible).

Pour observer cela, plusieurs mesures sont associées à un cluster :

- Sa densité (la masse d'objets par unité volumique)
- Sa variance (le degré de dispersion des objets dans l'espace depuis le centre du cluster)
- Sa dimension (typiquement son radius ou son diamètre)
- Sa forme (hyper sphérique/allongée/concave/convexe,...)
- Sa séparation (par rapport aux autres clusters).

2.2 Classification non-supervisée (Clustering) :

2.2.1 Définition :

La classification est l'opération statistique qui consiste à regrouper des objets (individus ou variables) en un nombre limité de groupes, (classes, segments, ou clusters), qui ont deux propriétés.

- D'une part, ils ne sont pas prédéfinis par l'analyste mais découverts au cours de l'opération, contrairement aux classes du classement.
- D'autre part, les classes de la classification regroupent les objets dans des classes.

Les objets avec une similarité maximisée sont dans une même classe (forte similarité intra-classe), et les objets avec une similarité minimisée sont dans des classes différentes (faible

similarité inter-classe), ce qui peut être mesuré par des critères telle que l'inertie interclasse, Comme le montre la (**Figure 2.1)** Dans notre cas, la similarité est associée à la distance des objets entre eux et à leur concentration.

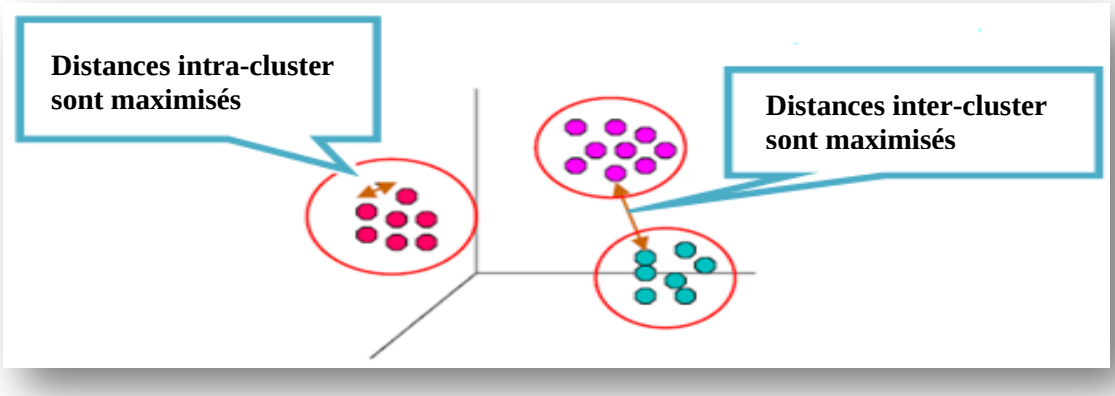


Figure 2.1 : Forte similarité intra-classe et faible similarité inter-classe

Donc la classification consiste à répartir les objets en clusters, dont on ne connaît pas à l'avance la classe à laquelle chaque objet appartient et même le nombre de ses classes n'est pas fixé à l'avance [14].

Exemple :

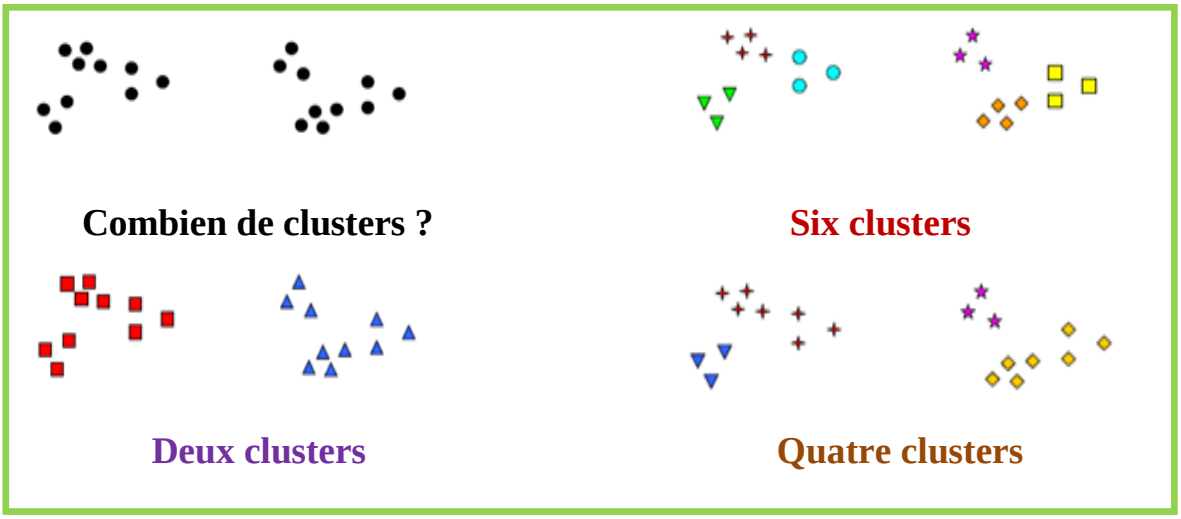


Figure 2.2 : Différentes façons de diviser les points en clusters

2.2.2 Domaines d'applications du Clustering:

Voici certains domaines et sans doute les plus populaires présentés dans le **(Tableau 2.1):**

Domaine	Formes des données	Clusters
Text mining	Textes Mails	Textes proches Dossiers automatiques
Web mining	Textes et images	Pages Web proches
Bioinformatique	Gènes	Gènes ressemblants
Marketing	Infos clients, produits achetés	Segmentation de clientèle
Segmentation d'images	Images	Zones homogènes dans l'image
Web log analysis	Clickstream	Profils utilisateurs

Tableau 2.1 : Domaines d'applications du Clustering

2.2.3 Les trois principales étapes du Clustering :

Le terme "Clustering" est utilisé dans beaucoup de communautés de chercheurs pour désigner les méthodes de regroupement des données non étiquetées. Le modèle typique de ce processus peut être résumé par les trois parties de la (**Figure 2.3)**.

Les étapes principales de ce processus sont donc [27] :

1. La préparation des données (prétraitement) :
2. L'algorithme de Clustering (structuration) :
3. L'exploitation des résultats de l'algorithme (interprétation).

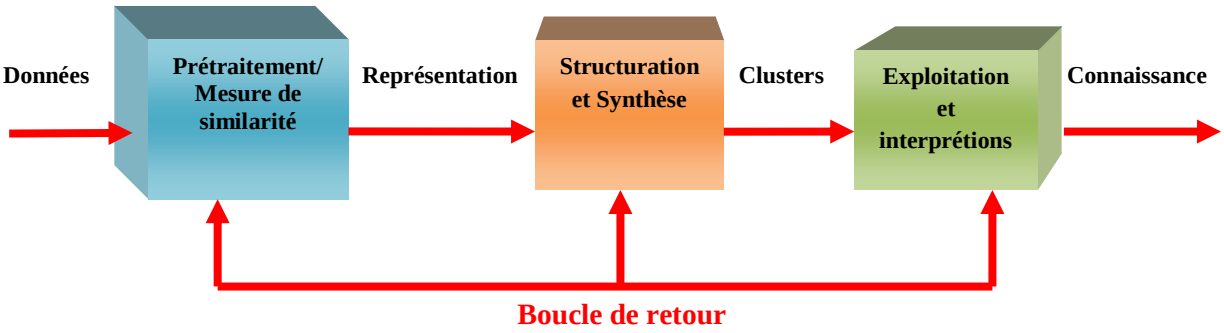


Figure 2.3 : Principales étapes du Clustering

2.2.4 Exigences du Clustering :

Les principales exigences qu'un algorithme de Clustering doit satisfaire sont:

- Evolutivité.
- Traitement des différents types d'attributs.
- Découverte les clusters de forme arbitraire.
- Exigences minimales en matière de connaissance du domaine pour déterminer les paramètres d'entrée.
- Capacité à traiter avec le bruit et les valeurs aberrantes.
- Insensibilité à l'ordre des enregistrements d'entrée.
- Grande dimension.
- L'intelligibilité et la convivialité.

2.2.5 Problèmes du Clustering :

Il y a un certain nombre de problèmes relatifs au regroupement (Clustering), Parmi eux :

- Les techniques actuelles du Clustering ne couvrent pas tous les besoins de manière adéquate.
- Le traitement d'un grand nombre d'éléments de données et de dimensions peut être problématique en raison de la complexité de temps.
- L'efficacité de la méthode dépend de la définition de la "distance" (regroupement fondé sur la distance).
- Le résultat de l'algorithme de Clustering peut être interprété de différentes manières.

2.2.6 Les mesures de ressemblance :

Afin de définir l'homogénéité d'un groupe d'observations, il est nécessaire de mesurer une ressemblance entre deux observations. En introduisant ainsi la notion de dissimilarité et de similarité ou bien de la distance qui peut être utilisée, qui sont la base d'une classification non-supervisée [7] :

2.2.6.1 Mesure de dissimilarité :

Une dissimilarité est une fonction d qui à tout couple (x_1, x_2) associe une valeur dans R_+ telle que : $d(x_1, x_2) = d(x_2, x_1) \geq 0$

$$d(x_1, x_2) = 0 \Rightarrow x_1 = x_2$$

Autrement dit, moins les unités x_1 et x_2 se ressemblent, plus le score est élevé. Remarquons qu'une distance est une dissimilarité, puisque toute distance possède les deux propriétés précédentes ainsi que l'inégalité triangulaire. Toutes les distances connues, en particulier la distance euclidienne, sont donc des exemples de dissimilarité [8].

À l'inverse, une autre possibilité consiste à mesurer la ressemblance entre observations à l'aide d'une similarité.

2.2.6.2 Mesure de similarité :

Une similarité est une fonction S qui à tout couple (x_1, x_2) associe une valeur dans R_+ tel que : $S(x_1, x_2) = S(x_2, x_1) \geq 0$

$$S(x_1, x_1) \geq S(x_2, x_2)$$

Contrairement à la dissimilarité, plus les unités x_1 et x_2 se ressemblent, plus le score est élevé. On peut citer comme exemple de similarité la valeur absolue du coefficient de corrélation :

$$|\rho(x_1, x_2)| = \left| \frac{\sum_{j=1}^{p} (x_{1j} - x_{1\blacksquare})(x_{2j} - x_{2\blacksquare})}{\sqrt{\sum_{j=1}^{p} (x_{1j} - x_{1\blacksquare})^2 \sum_{j=1}^{p} (x_{2j} - x_{2\blacksquare})}} \right|$$

Le choix de la distance est une question primordiale pour les méthodes exploratoires multi-variées. En effet, c'est à cet étape qu'il est possible de l'utiliser au mieux [8].

2.2.6.3 Les distances :

Les distances les plus utilisées pour les données de type quantitatives *continues* ou *discrètes* sont :

- Distance Eeuclidienne : $dist(x_1, x_2) = \sqrt{\sum_{j=1}^{p} (x_{1j} - x_{2j})^2}$
- Distance de Manhattan [13] : $\sum_{j=1}^{p} |x_{1j} - x_{2j}|$

❖ Distance de Jaccard (pour les ensembles) [4] : $dist(A,B)^2 = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$

Remarque :

Il est bien plus important de choisir la distance entre individu que l'algorithme de classification, c'est l'étape cruciale.

2.2.7 Evaluation de la qualité d'un Clustering :

Comme il existe un nombre important de résultats de Clustering possibles pour un même jeu de données, l'objectif est d'évaluer si un de ces résultats est meilleur qu'un autre. Cette notion de *meilleur* est à définir, et, est souvent dépendante de la méthode utilisée. L'évaluation d'un Clustering contient toujours une part de subjectivité et qu'il est impossible de définir un critère universel qui permettrait une évaluation sans biais de tous les résultats produits par toutes les méthodes de Clustering existantes. Cependant, un certain nombre de critères existent et sont utilisés de manière récurrente par de nombreux chercheurs pour comparer les résultats obtenus. Les inerties inter-classes et intra-classes sont largement utilisées [14].

2.2.7.1 Inerties inter-classes et intra-classes :

L'inertie totale I de la population, est la moyenne pondérée des carrés des distances individus au barycentre (centre de gravité) de la population (x_G *est la moyenne des* x_i)

$$\boldsymbol{x_G = \frac{1}{n}\sum_{i=1}^{n} x_i}$$

Elle peut s'écrire : $\sum_{i \in I} \boldsymbol{p_i}(\boldsymbol{x_i - x_G})^2$

L'inertie d'une classe est calculée de la même façon, par rapport à son barycentre, et elle peut s'écrire, (si la population $\sum_{i \in I} \boldsymbol{p_i}(\boldsymbol{x_i - x_G})^2$ est $\boldsymbol{I_1, \cdots, I_K}$ segmentée en k classes, d'inertie inter-classe):

$$\boldsymbol{I_A = \sum_{j=1}^{K} I_j}$$

Une classe sera d'autant plus homogène que son inertie sera plus faible, et la classification de la population sera d'autant meilleure que $\boldsymbol{I_A}$ sera petite. Enfin, l'Inertie inter-classe de la classification est définie comme étant la moyenne (pondérée par la somme des poids $\boldsymbol{I_R}$ de chaque $\boldsymbol{p_j = \sum_{i \in I_j} p_i}$ classe) des carrées des distances des barycentres de chaque classe au barycentre global.

Elle peut s'écrire : $\sum_{j \in classes} (\sum_{i \in I_j} p_i) \left(f x_{G_j} - x_i\right)^2$

Plus, elle est grande, et plus les classes sont séparées les uns des autres, ce qui indique une bonne classification (**Figure 2.4**).

Il est important de noter qu'une classification en K+1 classes aura une inertie inter-classe plus élevée qu'une classification en K classes et lui sera donc « supérieure», on ne peut donc comparer selon le seul critère inertiel deux classifications ayant des nombres de classes différents. Là réside le défaut incohérent aux méthodes de classifications selon un critère purement inertiel : l'optimisation de ce critère conduit, si l'on ne fixe pas a priori le nombre de classes, à isoler tous les individus en autant de classes distinctes réduites à un individu.

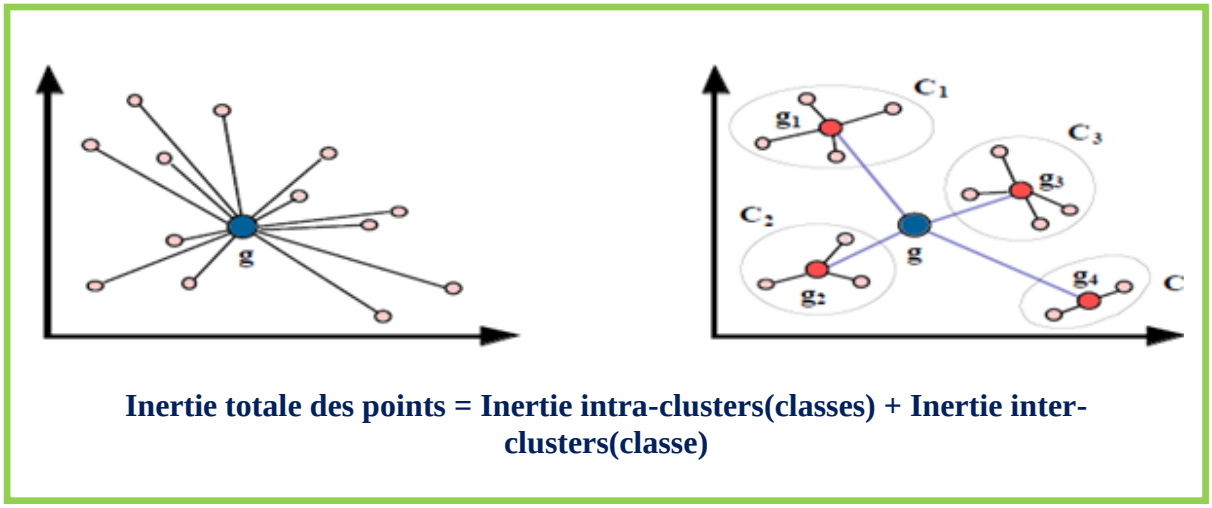


Figure 2.4 : Inertie inter-classes, intra-classes et Totale

2.2.8 Les différentes approches de la classification non-supervisée (Clustering) :

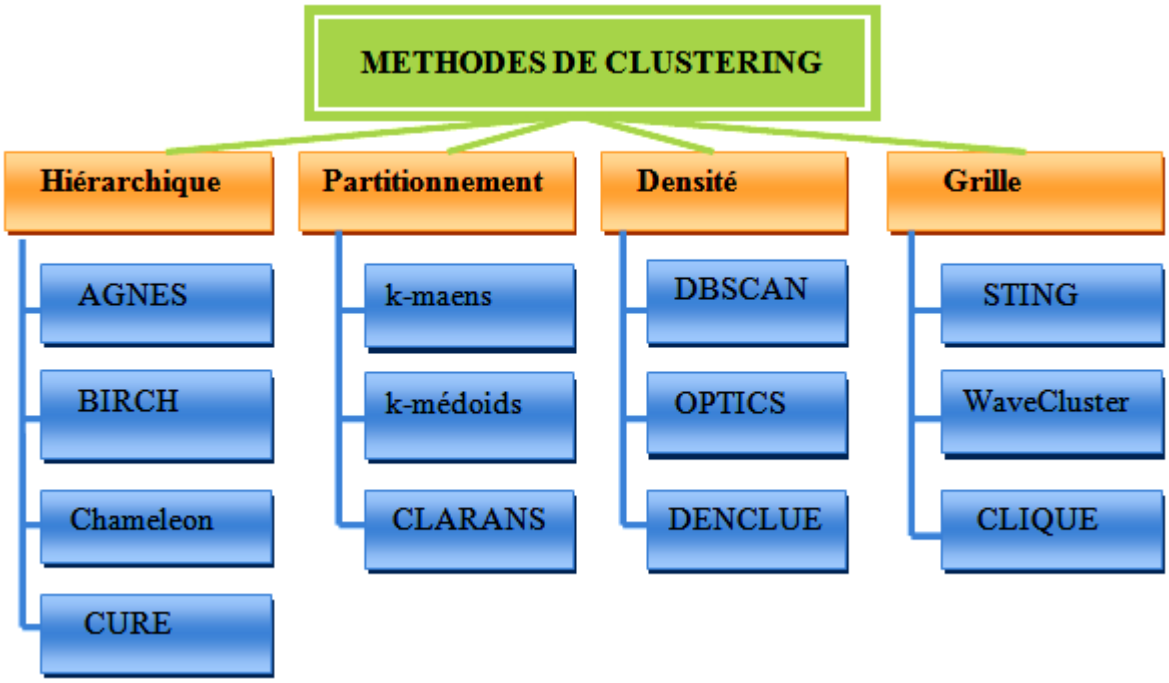


Figure 2.5 : Les différentes approches du Clustering

2.2.8.1 Les méthodes hiérarchiques :

Il y a deux méthodes dans la classification hiérarchique : la **C**lassification **A**scendante **H**iérarchique **CAH** et la **C**lassification **D**escendante **H**iérarchique **CDH :**

2.2.8.1.1 Classification Ascendante Hiérarchique (CAH):

Permet de construire une hiérarchie entière des objets sous forme d'un ″arbre″ dans un ordre ascendant. On commence par considérer chaque individu comme une classe et on essaye de fusionner deux ou plusieurs classes appropriées (selon une similarité) pour former une nouvelle classe. Le processus est réitéré jusqu'à ce que tous les individus se trouvent dans une même classe. Cette classification génère un arbre que l'on peut couper à différents niveaux pour obtenir un nombre de classes plus ou moins grand.

Différentes mesures de distance interclasses peuvent être utilisées citons : la distance euclidienne, la distance inférieure (qui favorise la création de classes de faible inertie) ou la distance supérieure (qui favorise la création de classes d'inertie plus importante).

Le schéma d'un algorithme CAH [13], [23], [11] est le suivant :

1. Les classes initiales sont les individus eux-mêmes.
2. On calcule les distances entre les classes.

3. Les deux classes les plus proches sont fusionnées et remplacées par une seule, comprenant que des individus très semblables.

4. Puis à partir de celles-ci, on construit des classes de moins en moins homogènes, jusqu'à aboutir à une seule classe, qui contient toutes les observations.

Cette procédure est basée sur deux choix :

1. La détermination d'un critère de ressemblance entre les individus. Cette méthode laisse à l'utilisateur le choix de la dissimilarité.
2. La détermination d'une dissimilarité entre classes : ce procédé est appelé critère d'agrégation. La méthode laisse à l'utilisateur le choix de ce critère.

Parmi les algorithmes basés sur le principe de **CAH** citons: **Algorithme CURE**[28], **Algorithme de BIRCH** [29,30].

2.2.8.1.2 Classification Descendante Hiérarchique (CDH) :

Dans la **CDH,** on considère tous les individus comme une seule classe au début, on divise successivement les classes en classes plus raffinées. Le processus continue jusqu'à ce que chaque classe contienne un seul point, ou bien, on atteint un nombre de classe désiré.

A l'inverse de la classification ascendante hiérarchique, à chaque étape de l'algorithme il y a deux processus à faire :

1. Chercher une classe à scinder.
2. Choisir un mode d'affectation des objets aux sous-classes

Parmi les algorithmes basés sur le principe de **CDH** citons: **Algorithme Williams et Lambert** [32], **Algorithme TSVQ** [1].

2.2.8.1.3 Les avantages et les inconvénients des méthodes hiérarchiques:

Les constructions hiérarchiques par divisions successives ont un aspect séduisant[20] :

Elles commencent par le haut de l'arbre, c'est-à-dire par la partie sur laquelle repose essentiellement l'interprétation. Malheureusement les simplifications drastiques qu'elles exigent, pour maintenir des temps de calcul raisonnables, font que les résultats obtenus sont souvent décevants. Cependant les dichotomies basées sur des variables bien choisies ont l'avantage d'être rapides et de fournir des interprétations aisées. Elles permettent donc de traiter facilement de très grands jeux de données avec peu de variables.

Les Avantages des méthodes hiérarchiques :

- Flexibilité concernant le niveau de granularité.

- Facilité de manipuler toute forme de similarité ou de distance.
- Applicabilité à tout type d'attribut.

Les inconvénients des méthodes hiérarchiques :

- La difficulté de choisir la droite arrêtant des critères.
- La plupart des algorithmes hiérarchiques ne révisent pas des classes (d'intermédiaire) une fois ils sont construits.

2.2.8.2 Les méthodes non-hiérarchiques :

2.2.8.2.1 Les approches par partitionnement :

Contrairement aux approches hiérarchiques, les approches par partitionnement cherchent la meilleure partition en k classes disjointes des données, le nombre de classes (clusters ou groupes) k étant fixé a priori. Les approches par partitionnement utilisent un processus itératif fonction du nombre k qui consiste à affecter chaque individu à la classe la plus proche au sens d'une distance (ou d'un indice de similarité) en optimisant une certaine fonction objective traduisant le fait que les individus doivent être similaires au sein d'une même classe, et dissimilaires d'une classe à une autre. Les classes de la partition finale, prises deux à deux, sont d'intersection vide et chacune est représentée par un noyau (un ou des individus de la population, ou un point de l'espace). La plupart des approches supposent un nombre prédéfini des classes. Les algorithmes de partitionnement sont divisés en trois grandes sous-familles : les méthodes des k-moyennes (k-means) [5], les méthodes des k-médianes (k-medoids), (Il y'a plusieurs versions des méthodes de k-médianes : PAM [18], CLARA [18], CLARANS [24] [25]), et les méthodes des nuées dynamiques[5], selon la définition des représentants des classes.

2.2.8.2.2 Les approches fondées sur la grille :

L'idée de ces méthodes, est de diviser l'espace de données en un nombre fini de cellules formant une grille. Ce type d'algorithme est conçu pour des données spaciales. Une cellule peur être un cube, une région, ou un hyper rectangle.

En fait, avec une telle représentation des données, au lieu de faire la classification dans l'espace de données, on la fait dans l'espace spacial en utilisant des informations statiques des points dans la cellule. Les méthodes de ce type sont hiérarchiques ou de partitionnement. Les algorithmes les plus connus sont : **STING**[33], **CLIQUE**[26], **WaveCluster**[12].

2.2.8.2.3 Les approches fondées sur la notion de densité :

L'idée des méthodes basées sur la densité est de définir une classe comme étant un ensemble d'individus de forme quelconque, mais "dense" selon un critère de voisinage et de connectivité. Ces méthodes sont fondées sur les concepts de densité, noyau, point limite, accessibilité et connectivité détaillés dans la référence [15].

Un des algorithmes les plus utilisés est DBSCAN [21], et ses dérivées tels que DBCLASD [34] ou OPTICS [19]. L'idée principale est de définir la notion de voisinage de rayon $\boldsymbol{\varepsilon}$ d'un point : tous les points situés à une distance de ce point inférieure à $\boldsymbol{\varepsilon}$ appartiennent alors à son voisinage. A partir de cette notion de **$\boldsymbol{\varepsilon}$-voisinage**, les auteurs définissent les concepts de point central (un point contenant au moins $\boldsymbol{MinPts}$ points dans son voisinage) et de densité-accessible entre deux points (s'il existe une chaîne de points centraux reliés par leur voisinage). Chaque classe est identifiée par des ensembles compacts de points centraux (appelés aussi points noyaux), les points situés à leur périphérie constituent sa bordure (appelés aussi points limites). L'algorithme a le déroulement suivant :

1. Choisir aléatoirement un point $\boldsymbol{x}$ de l'ensemble d'individus $\boldsymbol{X}$.
2. Rassembler dans un groupe $\boldsymbol{c}$, tous les points densité-accessible à partir de $\boldsymbol{x}$.
3. Si $\boldsymbol{x}$ est un point central, alors $\boldsymbol{c}$ est une classe.
4. Sinon, $\boldsymbol{x}$ est un point limite alors aucun point n'est atteignable à partir de $\boldsymbol{x}$.

L'algorithme sélectionne un autre point $\boldsymbol{x} \in \boldsymbol{X}$ et reprend en 2 jusqu'à avoir balayer tous les points de $\boldsymbol{X}$.

Ce type d'algorithme présente l'intérêt de trouver lui-même une évaluation du nombre de classes, et celles-ci peuvent avoir des formes arbitraires. Il permet également de gérer tout type de données et de bien tenir compte des données aberrantes, qui ne sont pas affectées aux clusters identifiés. Cependant comme pour BIRCH [29] et CHAMELEON [10], l'algorithme requiert l'entrée des paramètres $\boldsymbol{\varepsilon}$ et $\boldsymbol{MinPts}$, et l'expérience montre que les résultats obtenus sont très sensibles aux choix de ces paramètres [17].

2.2.8.2.3.1 DBSCAN :

***D**ensity **B**ased **S**patial **C**lustering of **A**pplications with **N**oise*

(***Ester, et al. KDD '1996***) [22]. Avec DBSCAN, la découverte d'un groupe se fait en deux étapes. Tout d'abord, un point dense est choisi aléatoirement, puis tous les points qui sont accessibles à partir de ce point, selon le critère de densité, forment le groupe. Deux paramètres

indiqués par l'utilisateur entrent en jeu : Epsilon, (le rayon de voisinage), et MinPts, le seuil de densité qui correspond au nombre minimal d'objets dans le voisinage d'un point.

2.2.8.2.3.2 OPTICS :

***O**rdering **P**oints **T**o **I**dentify the **C**lustering **S**tructure*

(***Ankerst, Breunig, Kriegel, and Sander '1999***)[19]. Dans l'utilisation d'OPTICS, seul un Epsilon limite est à définir, le Clustering est effectué pour plusieurs valeurs du rayon de voisinage simultanément. Cette solution est aussi appropriée pour retrouver la hiérarchie de clusters. OPTICS calcule ainsi un ordre de Clustering représentant la structure de densité des données.

2.2.8.2.3.3 DENCLUE:

DEN**sity − based **CLU**st**E**ring* (Hinneburg and Keim* `1998**) .

DENCLUE modélise la densité avec la notion d'influence, c'est-à-dire que chaque donnée exerce une influence sur ses voisins. Cette influence décroit avec la distance. Elle est représentée par une fonction mathématique. DENCLUE est capable de travailler dans des ensembles de points multidimensionnels. Cette méthode est appropriée pour des ensembles de données très bruitées. Les paramètres d'entrée sont le rayon des classes et le nombre minimum d'objet. Elle est plus rapide que DBSCAN (jusqu'à 45 fois plus rapide que DBSCAN) [20].

2.2.8.2.3.3 GDBSCAN :

C'est une généralisation de DBSCAN. Cette solution permet une gestion plus complète dans l'espace. Les conditions sont plus souples au niveau de la définition de voisinage et de la densité.

2.2.8.2.3.4 PDBSCAN :

C'est une adaptation parallèle réalisée pour les bases de données plus conséquentes. PDBSCAN permet d'obtenir un DBSCAN distribué et performant pour les données très volumineuse

	DBSCAN	OPTICS	DENCLUE
Paramètres	-Epsilon : rayon de voisinage (distance maximale) -MinPts : seuil de densité (nombre minimal de points dans le voisinage)	-Un Epsilon limite est à choisir, le Clustering est effectué pour plusieurs simultanément (inférieurs à cet Epsilon limite). -MinPts	-Rayon des classes. -Nombre minimum d'objets
Performances	-Tient compte des points isolés (outliers). -Traite des données numériques de grandes tailles.	-Chaque cluster a son propre Epsilon. -L'ordre hiérarchique est conservé. -1.6 fois plus rapide que DBSCA N.	- Caractère multidimensionnel
Inconvénients	Fonctionne mal en trois dimensions	On ne peut pas fixer une densité unique. Les points ne sont pas effectués à des classes comme dans DBSCAN	Complexité de la résolution mathématique.

Tableau 2.2 : Comparaison entre les différents algorithmes fondés sur la densité

2.2.9. Comparaison des algorithmes de classification :

Plusieurs méthodes sont proposées pour le problème général de la classification comme mentionné dans le (**Tableau 2.3**) Ils diffèrent par les mesures de proximité qu'ils utilisent, la nature des données qu'ils traitent et les objectifs finaux de la classification. Chacune de ces méthodes possède ses points forts et ses points faibles. Les méthodes hiérarchiques ascendantes sont utilisées en cas de données de petite taille car la complexité est très élevée.

Si au contraire, des problèmes de temps d'exécution se posent, alors c'est les méthodes de k-means qui sont utilisées. En fin, si l'objectif est de fournir des classes de formes quelconques, alors ce sont les méthodes basées sur la densité ou sur des grilles qui sont utilisées.

Nom	Type d'algorithme	Paramètres d'entrée	Caractéristiques de l'algorithme	Types de données	Ordre de données
k-means	Partition	Nombre de classes	Ne traite pas les aberrants. Complexité : O(I kn) . Il existe certaines versions qui différent par la mise à jour des céntroids	Numérique	
PAM	Partition	Nombre de classes	Ne traite pas les aberrants. Complexité : $O(Ik(n - k)^2)$		
CLARA	Partition	Nombre de classes	Ne traite pas les aberrants. Complexité $O(ks^2 + k(nk))$		
CLARANS	Partition	Nombre de classes. Maximum nombre de voisins	Ne traite pas les aberrants. Complexité $O(kn^2)$. Combinaison entre PAM et CLARA donc donner une meilleure qualité de classes.		
CURE	Hiérarchique	Nombre d'embranchement, Seuil de compacité	Traite les aberrants. Complexité : $O(n^2 \log n)$. Utiliser le résumé de données. Trois versions pour données numériques		
BIRCH	Hiérarchique	Paramètres de densité Esp et MinPts .	Traite les aberrants. Complexité :O(n). Utiliser le résumé de données. Trois versions pour données numériques	Numérique	Sensible
DBSCAN	Basé sur densité	Rayon d'une classe , Nombre minimum d'objets	Traite les aberrants. Complexité : O(n log n).	Numérique	Sensible
DENCLUE	Basé sur densité	Rayon d'une classe , Nombre minimum , maximum d'une d'objets	Traite les aberrants. Complexité : O(n log n).		
OPTICS	Basé sur densité	Rayon d'une classe , Nombre minimum , maximum d'une classe, Nombre minimum d'objets	Traite les aberrants, Complexité : O(n log n)		
STING	Basé sur grille	Nombre de cellules au niveau le plus bas, Nombre d'objets dans une cellule .	Traite les aberrants, Complexité O(n)		
WavaCluster	Basé sur grille	Pour chaque dimension, Wavelet, Nombre d'applications de transformation	Traite les aberrants, Complexité O(n)		
CLIQUE	Basé sur grille	Taille de grille, Nombre minimum de points dans une cellule	Traite les aberrants, Complexité $O(c^k+k_n)$ Classification faite dans sous-espaces.	Numérique	Résistant

Tableau 2.3 : Comparaison entre les algorithmes du Clustering

Conclusion :

Dans ce chapitre qui traite la classification non-supervisée, nous avons présenté, en général, sa définition, parmi celles proposées dans la littérature, ainsi que ses exigences, et ses principales étapes, problèmes.

Dans un second temps, nous avons donné un aperçu sur les approches les plus connues du Clustering, en citant pour chacune les algorithmes les plus utilisés, en se focalisant particulièrement sur la méthode à base de densité dont nous allons interpréter l'un de ses algorithmes les plus performant DBSCAN dans le chapitre qui suit, pour conclure, un tableau comparatif entre tous les algorithmes de ses méthodes est proposé.

Chapitre 3 :

DBSCAN

Density Based Spatial Clustering of Applications With Noise

"Le modèle doit suivre les donnés et non l'inverse !"

Jean-Paul Benzécri.

Introduction :

Basé sur la densité, le regroupement spatial des applications avec bruits (DBSCAN) est l'un des algorithmes de la classification de données le plus répandu et le plus également cité dans la littérature scientifique [3], Les auteurs ont mis au point cet algorithme pour l'indication de classes en grande bases de données spatiales. Le but de ce chapitre est de présenter et d'expliquer au mieux le fonctionnement de ce dit algorithme.

3.1 Definitions:

L'algorithme DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est un algorithme de partitionnement de données proposé en 1996 par Martin Ester, Hans-Peter Kriegel, Jörg Sander et Xiaowei Xu [21].

Il s'agit d'un algorithme basé sur la densité dans la mesure où il s'appuie sur la densité estimée des clusters pour effectuer le partitionnement. Il peut trouver des clusters de forme arbitraire [21]. Cependant, les clusters qui se trouvent proches les uns des autres appartiennent généralement à la même classe.

L'algorithme DBSCAN utilise deux paramètres : le rayon du voisinage $\boldsymbol{Eps}$ et le nombre minimum de points $\boldsymbol{MinPts}$ qui sont deux paramètres de densité.

Pour comprendre le fonctionnement de l'algorithme DBSCAN il faut commencer par définir quelques concepts de base :

Définition 1:

Le **ε-voisinage** d'un point **p**, noté par $\mathbf{N_\varepsilon(q)}$, est définie par :

$\mathbf{N_\varepsilon(q) = \{p \in D \mid dist\ (p, q) \leq \varepsilon\}}$ où **D** est un ensemble de points et **dist (p, q)** est une fonction de la distance par exemple La distance euclidienne entre **p** et **q** [9].

Définition 2 :

Le nombre requis afin qu'un voisinage d'un objet donné soit considéré comme un cluster est appelé ici $\boldsymbol{MinPts}$ [27].

Définition 3 : noyau / core point

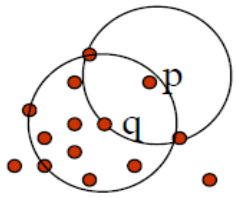
Un point **q** est un point de base (noyau) si $\mathbf{\mid N_\varepsilon(q) \geq MinPts \mid}$ [9] (les objets ne possédant pas ce type de voisinage sont nommés des objets non-noyaux) [27].

Définition 4 : accessible directement par densité / Directly density-reachable

Un point **p** est directement densité accessible d'un point **q** ***Eps***, ***Minpts*** si et seulement si :

1) **p** ϵ $N_\varepsilon(q)$ (**p** appartient au voisinage de **q** à un rayon ***Eps***).

2) $\left| N_\varepsilon(q) \geq \text{MinPts} \right|$ (condition : point de base).



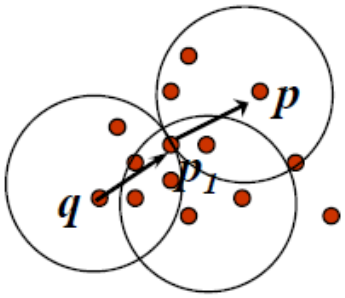
p est directement accessible par densité à partir **q**

q n'est directement densité accessible **p**

Figure 3.1 : Points accessible directement par densité

Définition 5: densité-accessible / Density-reachable

Un point **p** est dit joignable par densité à partir d'un point **q** par rapport à ***Eps*** et ***MinPts*** si et seulement s'il existe une suite de points (chaine) $\mathbf{p_1,\ldots,p_n}$ tels que :



i. $\mathbf{p_1} = q, \mathbf{p_n} = p$

ii. $\mathbf{p_{i+1}}$est joignable directement par densité à partir de $\mathbf{p_i}$ [2].

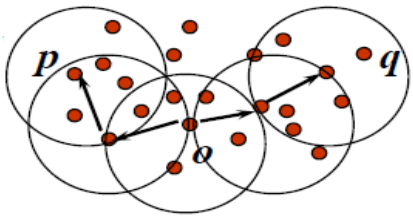
p est accessible par densité à partir **q**

q n'est accessible par densité à partir **p**

Figure 3.2 : Points densité-accessible

Définition 6: Connecté par densité / Density-connected

Un point **p** est dit connecté par densité à partir d'un point **q** par rapport à ***Eps*** et ***MinPts*** si et seulement s'il existe un point **o** tels que **p** et **q** sont accéssible par densité à partir de **o** par rapport à ***Eps*** et ***MinPts*** [21].



p et **q** sont connectés par densité par rapport à **o**

Figure 3.3 : Points connectés par densité

Définition 7: Groupe / Cluster

Un cluster **C** par rapport à ***Eps*** et ***MinPts*** dans **D** est un ensemble non-vide satisfaisant les conditions suivantes [21] :

1) **∀ p, q: si p ∈ C** et **q** est accessible par densité depuis **p** par rapport à ***Eps*** et ***MinPts***, alors **q ∈ C** (Maximalité)

2) **∀ p, q ∈ C: p** est connecté par densité à partir de **q** par rapport à ***Eps*** et ***MinPts*** dans **D**. (Connectivité)

Définition 8: Bruit / Noise

Soient $\mathbf{C_1}$**. . .,** $\mathbf{C_K}$ des clusters par rapport à ***Eps*** et ***MinPts*** dans la base de données **D** .

On définit le bruit comme étant un ensemble de points dans **D** qui n'appartient à aucun cluster $\mathbf{C_i}$**, i = 1,…,k** c'est à dire le **bruit = {p ∈ D | ∀ i: p ∉** $\mathbf{C_i}$**}** [21].

Définition 9: Point frontière / Border point

Un point **p** est dit point frontière s'il admet une densité inférieure à ***MinPts,*** mais se trouve dans le voisinage d'un point noyau.

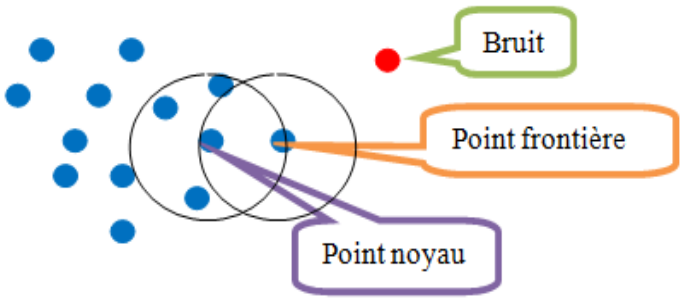


Figure 3.4 : Point noyau, point frontière et bruit

Lemme 1:

Une grappe peut être formé à partir d'un quelconque de ses points de base et aura toujours la même forme [21].

Lemme 2:

Soit **p** un point dans le groupe **C** de base à une distance donnée de minimum ***Eps*** et un nombre minimum de points dans cette distance ***MinPts***. Si l'ensemble **O** est densité accessible de **p** par rapport au même ***Eps*** et ***MinPts***, alors C est égal à l'ensemble **O**.

"Pour trouver un cluster, DBSCAN commence par un point arbitraire **p** et récupère tous les points accessibles par densité de **p** par rapport à ***Eps*** et ***MinPts***. Si **p** est un point de base, cette

procédure donne un cluster par rapport à ***Eps*** et ***MinPts*** (voir lemme 2). Si **p** est un point de la frontière alors il n'y a pas de points qui sont accessibles par densité à partir de **p** et DBSCAN visite le point de la base de données suivante " [21].

3.2 Présentation approfondie de l'algorithme DBSCAN :

Deux paramètres doivent être définis préalablement: le rayon de voisinage **Epsilon (*Eps*)** et le nombre de points minimal ***MinPts*** à l'intérieur de ce voisinage, ce qui permet d'établir un seuil de densité.

L'algorithme opère de la façon suivante pour chaque point du nuage :

- Prenant un point au hasard, l'algorithme crée un cluster à partir de ce point (on dit que c'est le noyau) s'il réalise ces deux conditions : il faut qu'il y ait au moins un certain nombre de points, déterminés par ***MinPts***, à une distance de voisinage inférieure à ***Eps***.
- Le cluster formé contient alors l'ensemble des points (l'intérieur du voisinage du noyau).
- Il opère de la même manière pour tous les autres points appartenant au cluster, augmentant ainsi le nombre de points du cluster. On dit que tous les points de ce cluster sont densité-accessibles.
- Il s'arrête lorsque les points ne respectent plus les critères ***Eps*** ou ***MinPts***, ce sont des points de bordure. Deux points de bordure sont densité-connectés.
- En prenant alors un autre point, un autre cluster peut être formé.

Remarque :

- Deux clusters de densité différente peuvent fusionner si la distance qui les sépare est inférieure à ***Eps***. C'est pour cela que les paramètres ***Eps*** et ***MinPts*** sont déterminants pour le nombre de clusters trouvé. On peut aussi noter que le choix de ces deux paramètres permet de supprimer le bruit plus ou moins efficacement.
- L'algorithme commence par un point au hasard, mais le résultat trouvé sera toujours le même.
- Pour définir un voisinage, il faut introduire une fonction de distance.

Exemple :

Regardons sur un exemple rapide, (**Figure 3.5**), le fonctionnement de cet algorithme en prenant ***MinPts*=2**, ce qui signifie qu'un point doit avoir au minimum **2** voisins dont le centre est à une distance inférieure à ***Eps*** pour ne pas être considéré comme étant du bruit. On voit

apparaître deux clusters, **C1** et **C2**, contenant respectivement **3** et **4** points. Les points bleus sont des points de type BRUIT, qui ne font partie d'aucun cluster.

Notons que l'algorithme est déterministe, puisque les résultats ne dépendent pas du choix du point initial. Ce n'est pas le cas de tous les algorithmes de regroupement par densité.

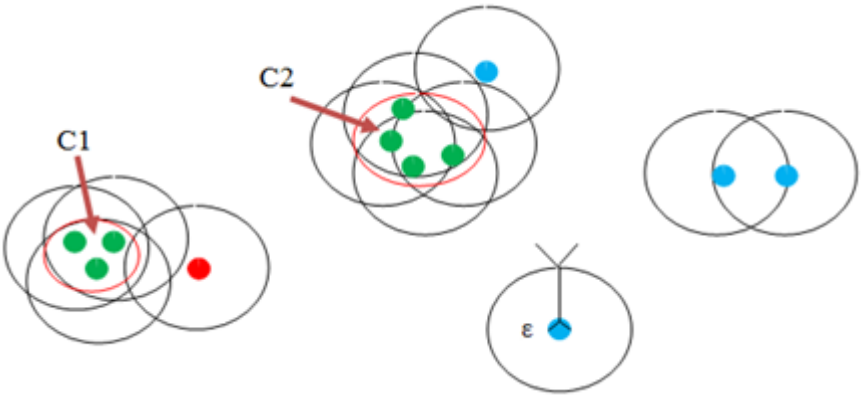


Figure 3.5 : Exemple du fonctionnement de DBSCAN

3.3 Détermination des paramètres *Eps* et *MinPts* :

Dans cette partie, nous allons présenter une heuristique simple et efficace pour déterminer les paramètres ***Eps*** et ***MinPts*** du plus petit cluster de la base de données.

Cette heuristique est basée sur les observations suivantes :

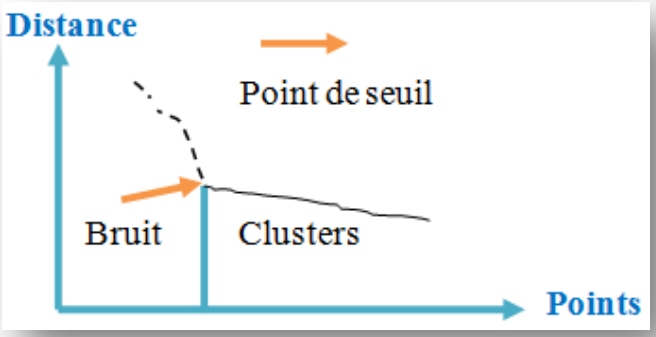


Figure 3.6 : Heuristique de fixation des paramètres

En général, il peut être délicat de détecter la première vallée automatiquement, mais il est relativement simple pour l'utilisateur de voir cette vallée sur une représentation graphique. C'est pourquoi une approche interactive pour déterminer ce seuil est intéressante.

DBSCAN a besoin des paramètres ***Eps*** et ***MinPts***. Les expériences ont montré que les graphes de distance **k (k > 4)** ne diffèrent pas vraiment des graphes de distance **4** mais

nécessitent des calculs bien plus important. Ainsi, nous éliminons le paramètre ***MinPts*** en le fixant à **4** pour toutes les bases de données **(d'espace 2D)**.

Il s'agit ensuite d'utiliser une approche interactive pour déterminer le paramètre ***Eps*** de DBSCAN:

- Le système calcul et affiche le graphe de distance **4** pour la base de donnée.
- Si l'utilisateur peut estimer le pourcentage de bruit, ce pourcentage est entré et le système en déduit une proposition pour le seuil de point.
- L'utilisateur peut accepter ou non le seuil proposé ou sélectionner un autre seuil qui est alors utilisé dans DBSCAN [21].

3.4 La complexité en temps et en espace :

L'encombrement de DBSCAN est O (m), seulement une petite quantité de données doit être stocké pour un point. Seul le nombre de clusters peut-être la qualification comme base, frontière ou un point de bruit. Seulement le nombre de cluster peut-être la classification comme base, frontière ou un point de bruit. Lors de l'exécution de l'algorithme, la complexité de temps est en O (m × temps pour trouver des points dans l Eps-voisinage), Où m est le nombre de données des points en l'ensemble de données D. Lors l'exécution d'une recherche linéaire pour trouver les points dans le voisinage, la complexité est O (m2), c'est le pire des cas. En utilisant une structure d'index comme le R-arbre, la recherche de voisinage peut être effectuée de manière plus efficace. La complexité dans ce cas est O (n log n) [31].

3.5 Les avantages et les inconvénients de DBSCAN :

3.5.1 Les avantages de DBSCAN :

- Cet algorithme présente l'intérêt de trouver lui-même une évaluation du nombre de classes. Celles-ci peuvent avoir des formes arbitraires.
- L'algorithme permet également de bien gérer les données aberrantes, qui ne sont pas affectées aux clusters détectés.

3.5.2 Les inconvénients de DBSCAN :

- Il requiert des paramètres Eps et MinPts, et l'expérience montre que les résultats obtenus sont très sensibles aux choix de ces paramètres.
- Il incapable de caractériser des clusters de densités différentes.

3.6 Les exigences DBSCAN [31] :

- Les exigences minimales de connaissances du domaine pour déterminer l'entrée des paramètres, car les valeurs appropriées ne sont pas souvent connues à l'avance lorsqu'il s'agit de grandes bases de données.
- Découverte des clusters avec la forme arbitraire, parce que la forme de regroupement dans la base spatiale peut être sphérique, étirée, linéaire, allongée, etc …
- Bonne efficacité sur de grandes bases de données, c'est à dire sur des bases de données beaucoup plus que quelques objets, des milliers.

3.7 Limites de la méthode :

Cela nécessite une bonne définition de la distance utilisée pour le voisinage. On verra aussi que les deux paramètres Epsilon et MinPts sont cependant difficiles à choisir et impactent largement les résultats. La méthode est moins efficace pour les problèmes à plusieurs dimensions.

3.8 Applications :

DBSCAN est utilisé dans de nombreux domaines :

- Médecine : regroupe différents types de cellules.
- Informatique : détecte les anomalies (alors considérées comme bruit).
- Imagerie : améliore la qualité des images satellites et télévisuelles.
- Statistique : crée des clusters dans les graphes.

3.9 Les outils utilisés :

3.9.1 L'environnement de développement (NetBeans) :

Les environnements de développement intégrés (EDI), sont des logiciels regroupant un ensemble d'outils nécessaires au développement logiciel dans un (ou plusieurs) langage(s) de programmation.

Parmi tous les environnements de développement existant notre choix a porté sur NetBeans pour simplifier le travail afin de développer notre application suite, à sa facilité d'utilisation. L'environnement NetBeans est un environnement de développement intégré IDE (*Integrated Développent Environnent*) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Développement and Distribution License), utilisé pour exécuter et compiler. On peut également trouver dans cet IDE un système de gestion de versions et différents outils pour faciliter la création graphique GUI. Par rapport aux autres environnements comme Eclipse, Wireless …, notre choix a été adopté non seulement pour faciliter l'interface de développement mais aussi pour sa flexibilité, en plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web), ainsi que des outils de débogage et de test des programmes. C'est un outil qui facilite énormément la phase de développement et des tests.

3.9.2 Langage de programmation (Java) :

Notre choix a porté sur Java qui est un langage de programmation à usage général, évolué et orienté objet, inventé en 1995 dont la syntaxe est proche du C. Il possède un certain nombre de caractéristiques : interprété, portable (il est indépendant de toute plate-forme) sur plusieurs système d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications, simple, fortement typé, assure la gestion de la mémoire, …etc. On a également choisi ce langage car il possède une bibliothèque graphique en standard et des outils facilitant le développement d'interfaces.

3.10 Conception, Implémentation et Expérimentation du prototype :

3.10.1 Conception :

3.10.1.1 Organigramme de l'algorithme de DBSCAN :

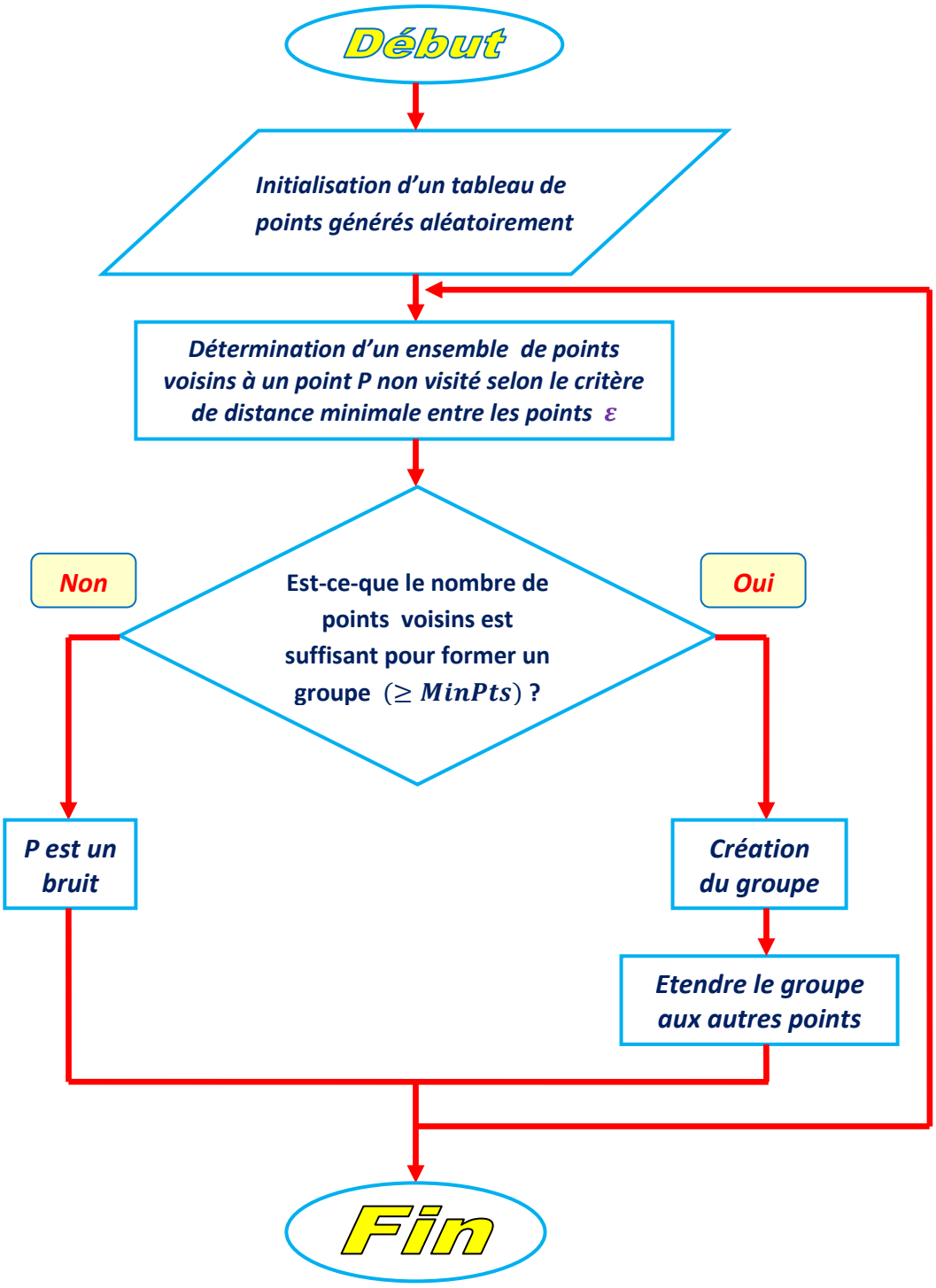


Figure 3.7 : Organigramme de l'algorithme DBSCAN

3.10.2 Implémentation :

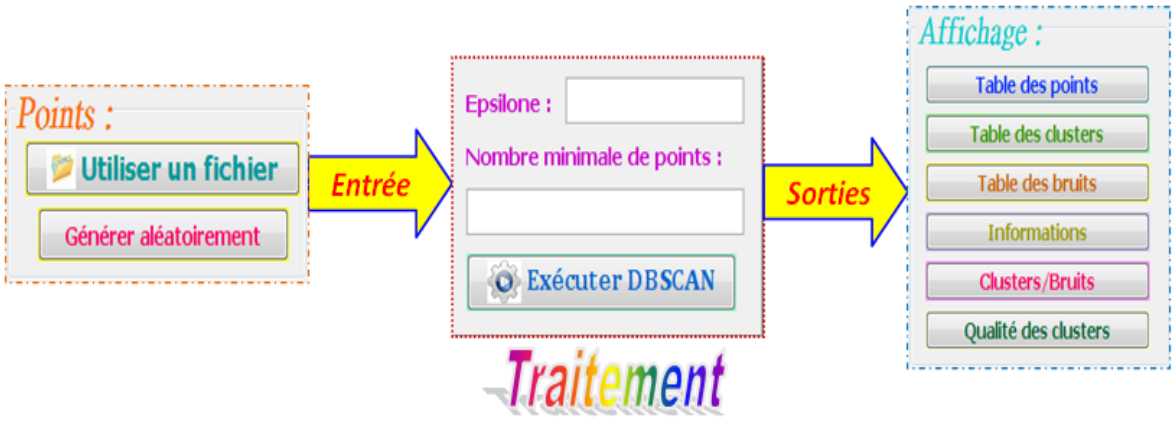


Figure 3.8 : Les principales étapes du déroulement de notre application

3.10.2.1 Entrées (Points) :

- Le bouton 

nous permet de générer un ensemble de points de manière aléatoire selon une loi uniforme (**Figure 3.9)**.

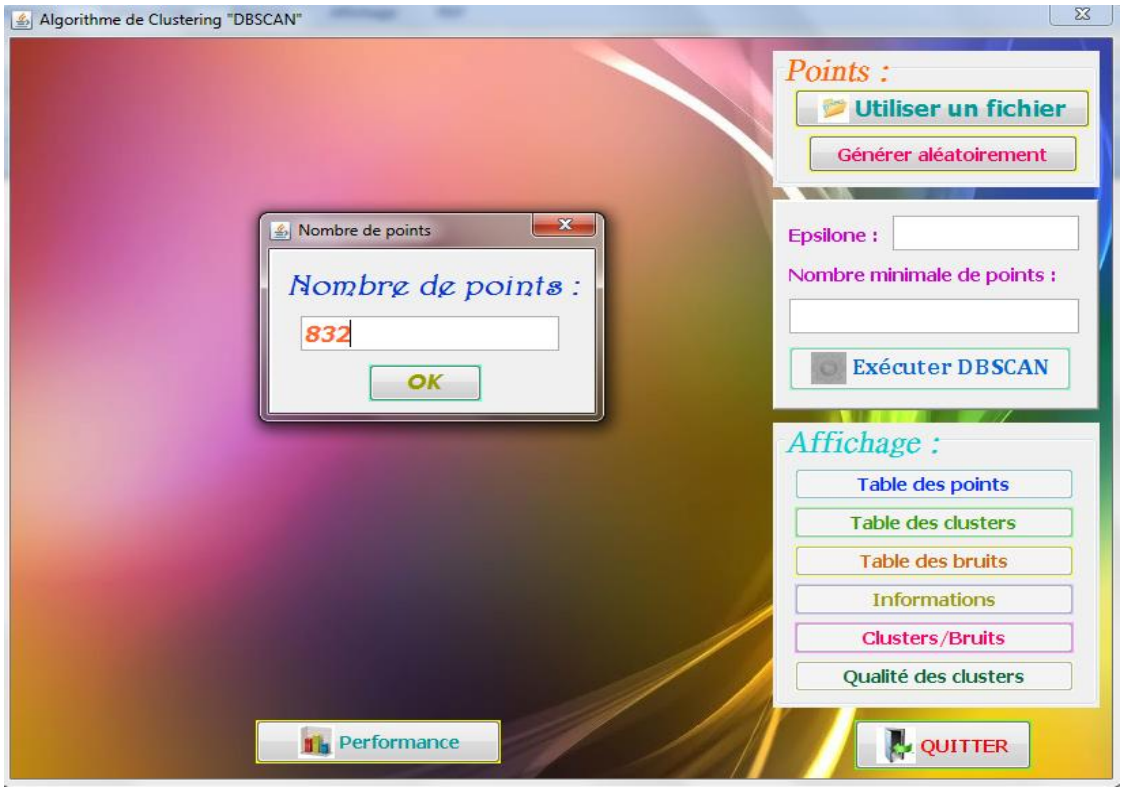


Figure 3.9 : Détermination du nombre de points

3.10.2.2 Traitement :

- Le bouton 

nous permet d'exécuter l'algorithme DBSCAN sur l'ensemble des points générés précédemment en distinguant les groupes « clusters » entre eux par des couleurs définies différentes du noir qui identifie les points bruits.

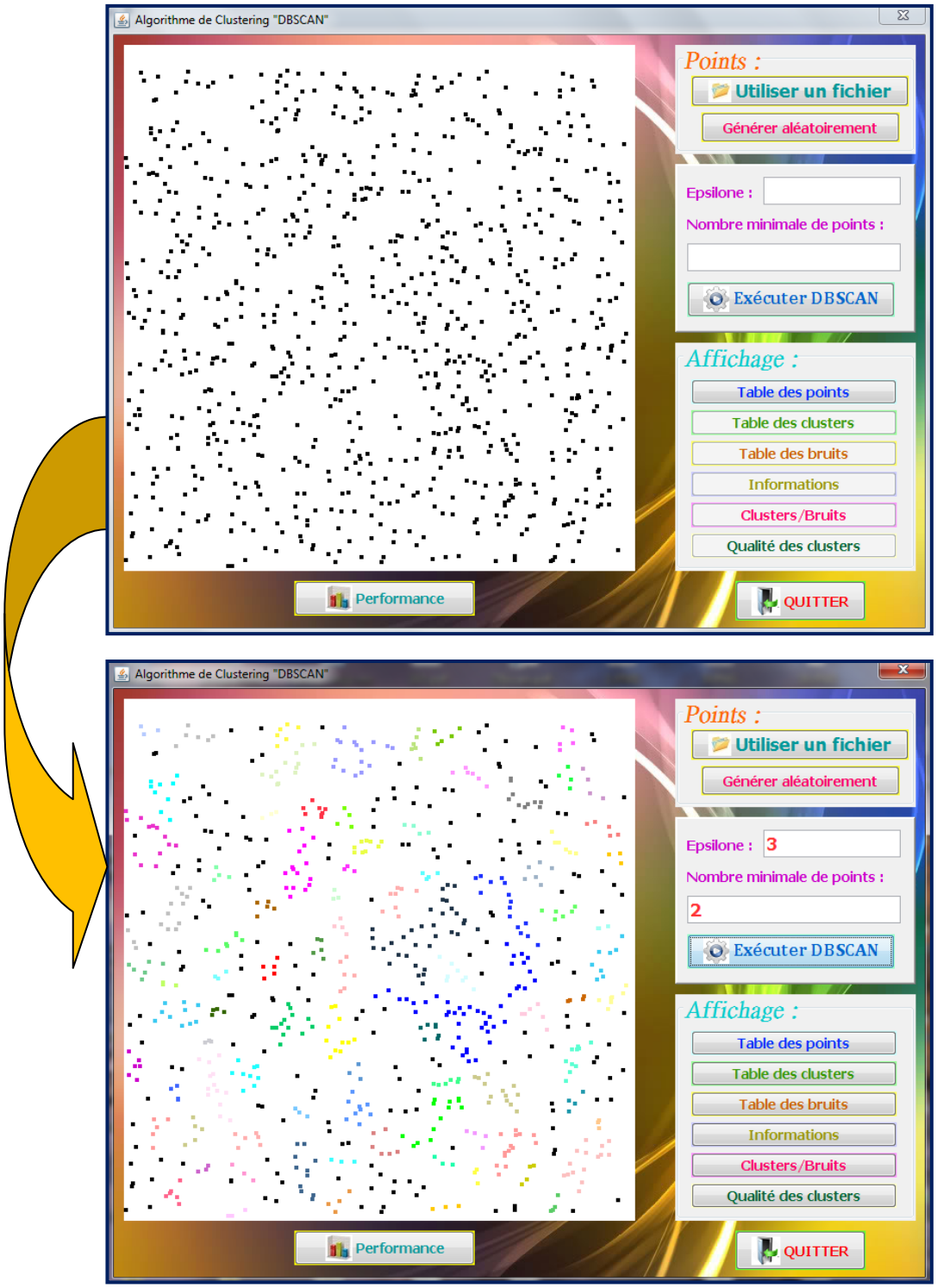
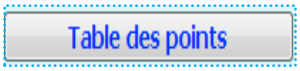
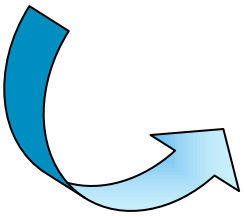


Figure 3.10 : L'état des points avant et après l'exécution de DBSCAN

3.10.2.3 Sorties (Affichage) :

- Le bouton 

nous permet d'afficher le tableau de tout les points générés, en fonction de leurs numéros et coordonnées (X, Y).



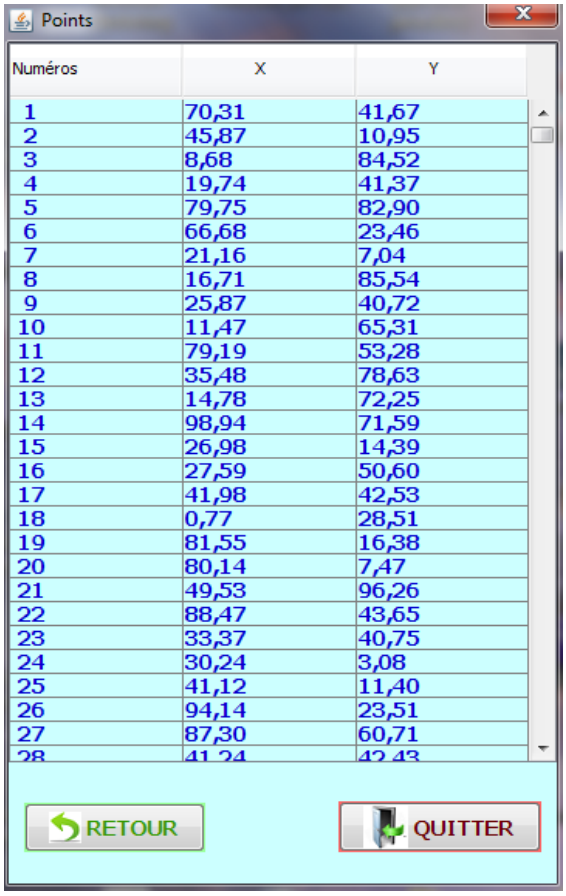


Tableau 3.1 : Tableau des points

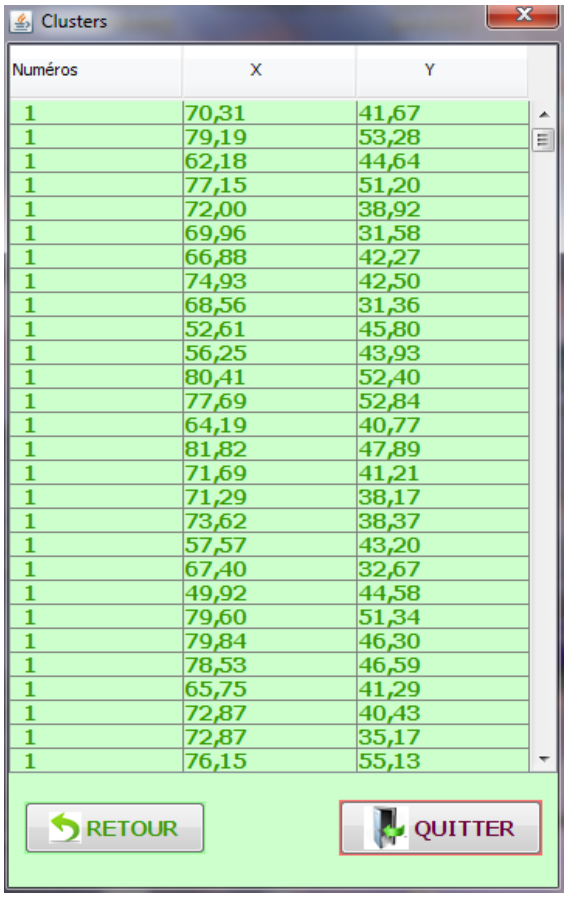


Tableau 3.2 : Tableau des groupes

- Le bouton 

nous permet d'afficher le tableau des groupes « clusters » formés en fonction du numéro de chaque groupe ainsi que l'ensemble des points le forment.

- Le bouton [Table des bruits] nous permet d'afficher le tableau des points bruits, en fonction de leurs numéros et coordonnées (X, Y).

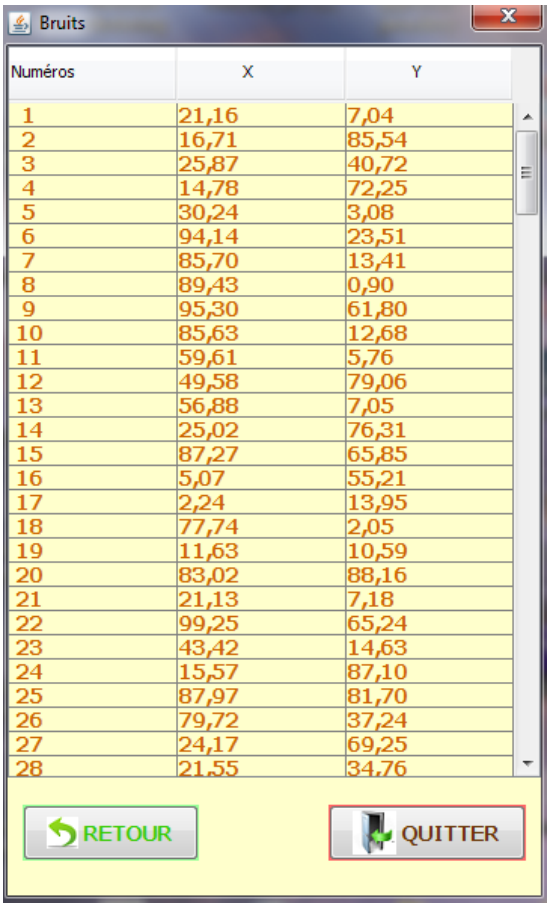


Tableau 3.3 : Tableau des bruits

- Le bouton [Qualité des clusters] nous permet d'afficher le tableau contenant l'inertie et l'inertie normalisée de chaque groupe « clusters », ainsi que l'inertie globale (de tous les groupes) et l'inertie globale normalisée.

Tableau 3.4 : Tableau de la qualité des groupes

- Le bouton Informations nous permet d'afficher quelques informations importantes : le nombre de points,le nombre de groupes « clusters », le nombre de bruits comme le montre **(Figure 3.11)**.

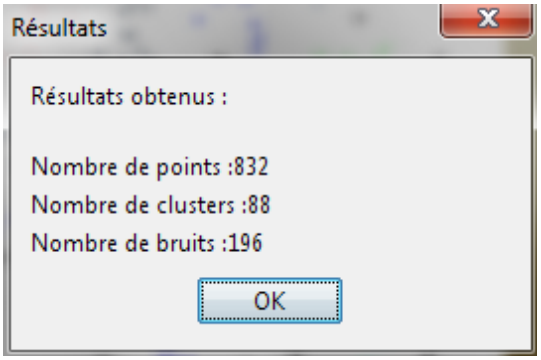


Figure 3.11 : Fenêtre d'information

- Le bouton Clusters/Bruits nous permet d'afficher les bruits, les groupes, de notre choix ou bien les deux en fonction de leurs couleurs respectives (par sélection).

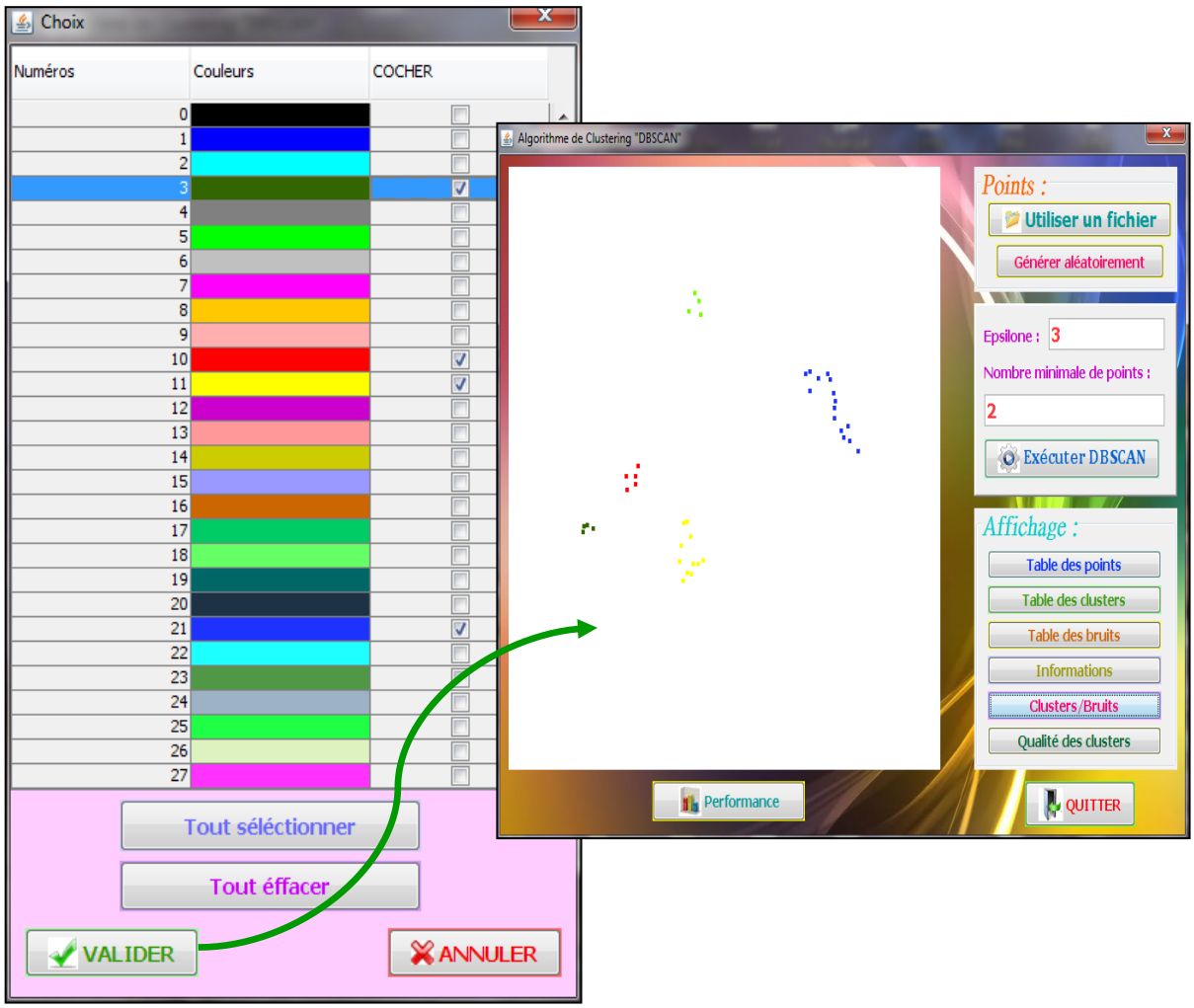


Figure 3.12 : Affichage des groupes/bruits par sélection

- Le bouton Performance nous permet d'afficher l'histogramme d'inerties des vingt premiers groupes.

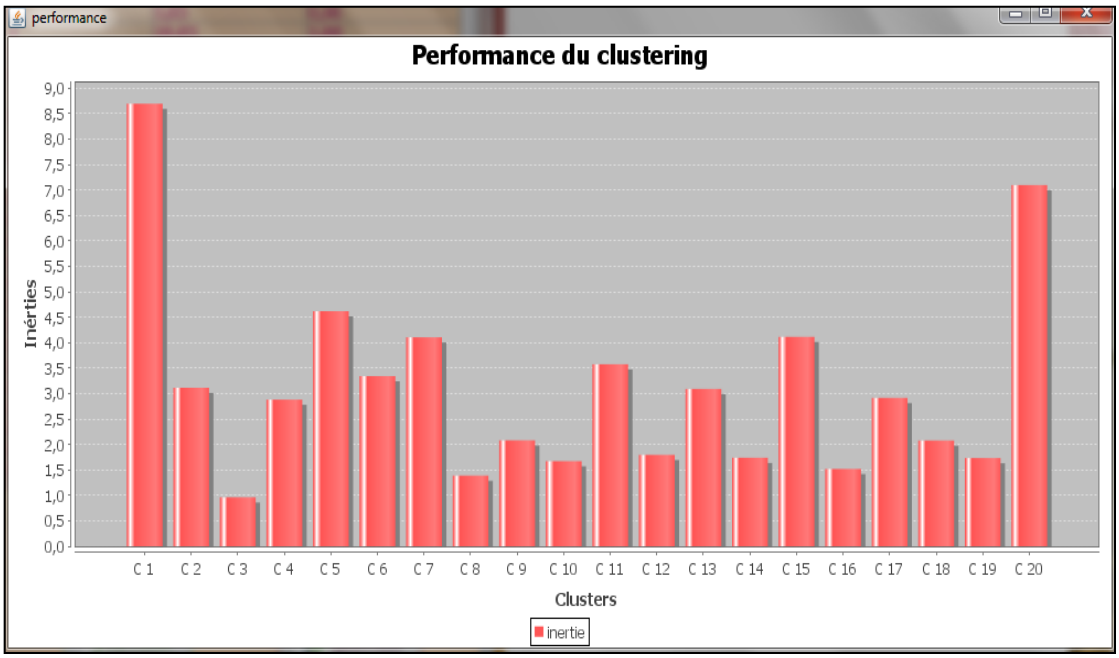


Figure 3.13 : Histogramme de performance des groupes

3.10.3 Expérimentation:

Dans notre travail on a pu rencontrer et vérifier l'inconvénient majeur de notre algorithme DBSCAN qui est l'influence du changement des valeurs de (Eps, MinPts) sur le même ensemble de points d'où l'influence sur la qualité (inertie) des groupes obtenus, comme le montre l'histogramme suivants **(Figure 3.14)**.

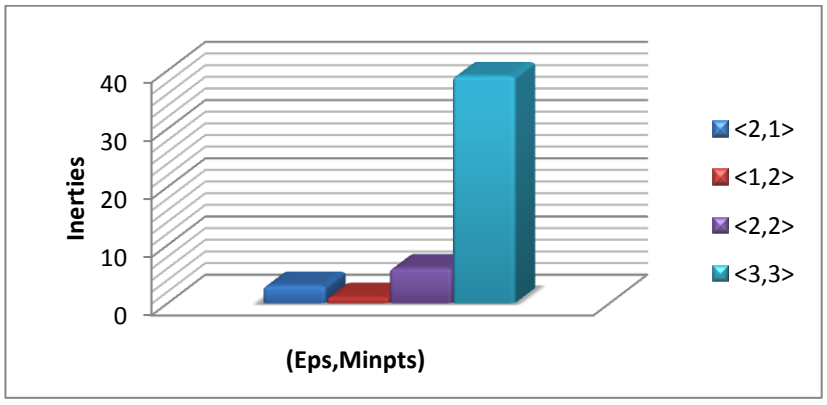


Figure 3.14 : Histogramme des inerties en fonction des valeurs (Eps, Minpts)

1) Moins l'inertie est grande ⇒ plus la qualité du Clustering est meilleur ⇒ mieux ses valeurs accordées précisément à (Eps, MinPts) sont adaptées à cet ensemble de points, dans notre cas Eps = 1, MinPts = 2, pour offrir l'inertie la plus minimale.

2) Plus le rayon de voisinage **Eps** est petit, plus le nombre de clusters est grand et le bruit important. Plus le nombre de points minimum **MinPts** est petit, plus le bruit est faible et le nombre de clusters est petit.

Conclusion :

Dans ce chapitre on a présenté, et expliqué le fonctionnement de l'algorithme DBSCAN, on a aussi donné un exemple simple ainsi que ses exigences, limites, avantages et inconvénients, de plus on a implémenté une version de cet algorithme qui permet de regrouper des points générés aléatoirement dans des groupes « clusters » similaires, à l'issue de deux conditions le critère de distance minimale entre les points **(Eps),** et le nombre minimale de points **(MinPts),** par la suite nous avons examiné les inerties de chaque groupe.

Conclusion Générale

" Je ne compte pas sur le passé, j'en tire des conclusions pour le présent."

Eric Fisher

Conclusion générale :

DBSCAN (*Densité-Based Spatial Clustering of Applications with Noise*) est l'un des algorithmes de classification basé sur la densité le plus populaire de sa catégorie.
Il permet de découvrir les clusters de formes arbitraires et de séparer les bruits.

Des limitations importantes de cet algorithme sont connues : il utilise des paramètres globaux de densité de sorte que le résultat de classification de base de données multi-densité est souvent inexact. En plus, son résultat est très sensible aux valeurs de ces paramètres qui sont fixés par l'utilisateur, ajoutant son appartenance au type dur *hard Clustering* c'est-à-dire que la probabilité d'un point (individu) appartient à un autre cluster autre que le sien, est nul contrairement à l'approche de l'algorithme EM (Expectation Maximisation) qui est une méthode du type soft Clustering, c'est-à-dire que la probabilité pour qu'un point appartient à d'autres clusters différents du sien existe avec un pourcentage raisonnable est logique calculé pour l'appartenance de chacun à ses derniers, si on aura à améliorer DBSCAN on lui combinant les points forts de EM notamment celui qui vient d'être cité dernièrement alors DBSCAN bénéficiera d'une amélioration performante. Néanmoins les résultats expérimentaux montrent que notre algorithme reste toujours efficient et le plus performant dans la détection des groupes de différentes densités

Référence :

- [1] A.Gersho and R.M. Gray. *Vector quantization and signal compression* Kluwer, Boston, 1992.
- [2] A. J. Izenman. Modern multivariate statistical techniques : regression, classification,and manifold learning. Springer, 2008.
- [3] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. ACM Comput.Surv., 31(3):264–323, September 1999.
- [4] De-Carvalho F.A.T. *Proximity coefficients between boolean object* . In New Approaches in Classification and Data Analysis, éd. Par E.Diday , Y.Lechevallier, M.Schaded, P.Bertrand. and B.Burtrchy. Springer Verlag, pp. 387-394. 1994.
- [5] E.W.Forgy . *Cluster Analysis of Multivariate Data : Efficiency Versus Interpretability of Classification* , Biometrics, 21, pp.768-780, 1965.
- [6] F. Brucker, Modèles de classification en classes empiétantes, Phd Thesis, Equipe d'acceuil : Dep. IASC de l'École Supérieure des Télécommunications de Bretagne,France, 2001.
- [7] G.Bison . *La ssimilarité : une notion symbolique/numérique*, In Apprentissage symbolique numérique, éd. Par B . Moulet Edition CEPADUES, pp. 169-201, 2000.
- [8] G.Celeux, E.Dilay, Y.Lechevallier, Govaert and *H.Ralambondrainy* . *Classification automatique des données*. Editions Dunod, Paris, 1989.
- [9] G. Cleuziou, Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information, Phd Thesis, Université d'Orleans, France, 2006
- [10] G. Karypis ., E.Han. and V.Kumar *Chameleon : A Hierarchical Clustering Algorithm Using Dynamic Modeling*, IEEE Computer, 32(8), pp. 68-75, 1999.
- [11] G. Saporta *Probabilités analyse des données et statistique*. Techniq, 1990.
- [12] G.Sheikholeslami, S.Chatterjee, and A.Zhang. Wavercluster : a wavelet-based clustering approach for spacial data in very large database. *The VLDB Journal*, 8(3-4) : 289-304, 2000.
- [13] Haim Brezis. *Analyse fonctionnelle : théorie et applications*, Editions Dunod, ISBN 9782100043149, Paris 1999.
- [14] H.Baccouch, LDBSCAN : Classification non supervisée basée sur la densité avec des paramètres de densités locales, Master en informatique , Université de Kairouan,2012.

- [15] J.Han,M. Kamber and Tung A. K. H. Spacial clustering methods in data mining, In Geographic Data Mining and Knowledge Discovery, éd. Par H.Miller and J.Han . Taylor and Franics, pp. 1-29, 2001.
- [16] J.P. Benzecri. *L'analyse des données*, volume Tome 1 : La taxonomie. Du-nord, 1973.
- [17] J.P. Nakache and J.Confais . Approche pragmatique de la classification, Editions Technip, Paris 2005.
- [18] L.Kaufman .and P.J.Rousseeuw. *Finding groups in data : An introduction to cluster analysis*, Johhn Wiley and Sons, New York, 1990.
- [19] M.Ankerst, M .M.Breunig, H.P.Kriegel, and J.Sander. OPTICS : Ordering points to identify the clustering structure. In Porceedings of the *ACM SIGMOD Conference*, pages 49-60, Philadelphia 1999.
- [20] M.Boubou, Contribution aux méthodes de classification non-supervisée via des approches prétopologiques et d'agrégation d'opinions,Doctorat Statistique-Informatique,Université Claude Bernard - Lyon1,2007.
- [21] M.Ester , H.P, J.Sander, Xu X. A density-based algorithm for discovering clusters in large data bases with noises. In Proceedings of 2nd International Conference of Knowledge Discovering in Databases and Data Mining (KDD-96), Portland, Oregon, August 1996.
- [22] M.Ester, H.P.Kriegel, J.Sander, and X .Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *In KDD*, pages 226-231, 1996.
- [23] M. Jambou. *Méthodes de base de l'analyse des données*. Eyrolles, 1999.
- [24] NG R. and J.*Han Efficient and effective clustering methods for spacial data mining*, Proceedings of the 20th Conference on VLDB, Santiago, Chilie., pp. 144-155, 1994.
- [25] NG R et J.Han. CLARANS : *A method for clustering objects for data minig*, IEEE Transactions on Knowledge and Data Engineering, 14(5), pp. 1003-1016, 2002.
- [26] R.Agrawal, J.Gehrke, D.Gunopulos, and P.Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. Pages 94-105, 1998.
- [27] S.Fouchal , Partitionnement d'objets complexes : méthodes et algorithmes,Doctorat en informarique,Université Paris8 Vincennes Saint-Denis,2011.
- [28] S.Guha, R.Rastogi, and K.Shim. Cure : an efficient clustering algorithm for large database. In SIGMOD '98 : *Proceedings of the 1998 ACM SIG-MOD international conference on management of data*, pages 73-84, New York , NY, USA, 1998, ACM Press.

- [29] T. Zhang, R.Ramakrishnan, and M. Livny.Brich : *an efficient data clustering method for very large databases. In SIGMOD '96 : Proceedings of the 1996 ACM SIGMOD international conference on management of data*, pages 103-114, New York, NY, USA, 1996. ACM Press.
- [30] T.Zhang, R.Ramakrishan, and M.Livny.Birch : Anew data clustering algorithm and its applications, 1997.
- [31] Wannes Meert, CLUSTERING MAPS , Master of Artificial Intelligence, Katholieke Universiteit Leuven, 2006.
- [32] W.T. Williams and J.M. Lambert. Multivariate methods in plan ecology. *Journal of Ecology*, 47(1) :83-101, 1959.
- [33] W.Wang, J.Yang, and R.R.Muntz. STING : A statistical information grid approach to spatial data mining, In Matthias Jarke, Michael J.Carey, Klaus R.Dittrich, Frederick H.Lochovsky, Pericles Loucopoulos, and Manfred A.Jeusfeld, editors, *Twenty-third International conference on Very Large Data Bases*, pages 186-195, Athens, Greecs, 1997. Morgan Kaufman.
- [34] X. Xu, M.ester, H.P. Kriegel, and J.Sander. *A distribution-based clustering algorithm from mining in large spacial databases. In ICDE '98 : Proceeding of the Fourteenth international conference on Data Engineering*, pages 324-331, Washington, DC , IEEE Computer Society, USA 1998.