

DATA MINING

RAPPORT

Mean Shift Clustering

Realise par :

KHALID MARZAQ

NAIMA EL MENANI

FATIMA ZAHRA TOUBA

nabilmarzaq74@gmail.com

elmenaninaimaa@gmail.com

toubatitim99@gmail.com

Table des matières

1	Introduction	2
2	Intelligence Artificielle, Machine Learning, Clustering	3
2.1	Intelligence artificielle	3
2.2	Machine learning	4
2.2.1	Types d'algorithmes d'apprentissage automatique . . .	4
2.3	Clustering	9
2.3.1	Algorithmes de clustering	9
2.3.2	Types d'algorithme de clustering	10
3	Mean Shift Clustering	12
3.1	Méthode mean-shift	12
3.1.1	Initialisation	12
3.1.2	Calculer la densité de probabilité estimée	12
3.1.3	Calcul du vecteur Mean-Shift	14
3.1.4	Mise à jour de la position	14
3.1.5	Vérification de convergence	15
3.1.6	Détermination des modes	16
3.2	Implementation	16
4	Conclusion	20

Introduction

Data Mining et Machine Learning sont des domaines qui se sont inspirés l'un de l'autre, bien qu'ils aient beaucoup de choses en commun, mais ils ont des objectifs différents.

Data Mining est effectuée par des humains sur certains ensembles de données dans le but de découvrir des modèles intéressants entre les éléments d'un ensemble de données. Le Data Mining utilise des techniques développées par le Machine Learning pour prédire les résultats.

Le Machine Learning, quant à lui, est la capacité d'un ordinateur à apprendre à partir d'ensembles de données exploités.

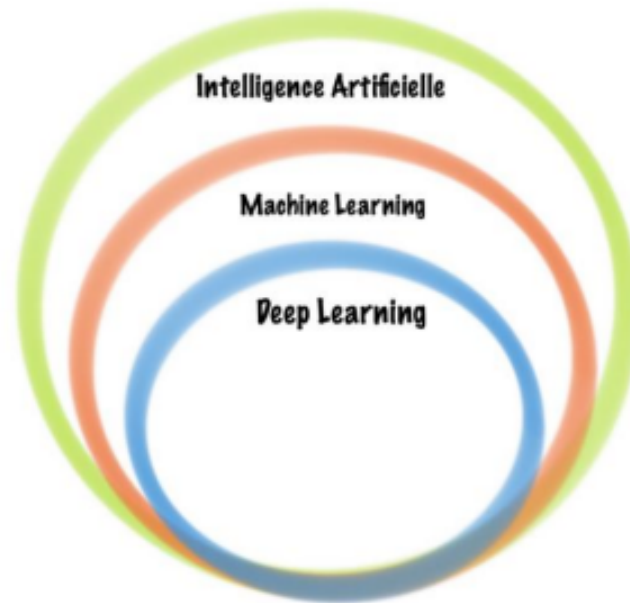
Les algorithmes de Machine Learning prennent les informations représentant la relation entre les éléments des ensembles de données et construisent des modèles afin de pouvoir prédire les résultats futurs. Ces modèles ne sont rien d'autre que des actions qui seront entreprises par la machine pour arriver à un résultat.

Intelligence Artificielle, Machine Learning, Clustering

2.1 Intelligence artificielle

L'intelligence artificielle (IA) est un ensemble de techniques, d'outils et de méthodes qui permettent aux machines de simuler l'intelligence humaine pour accomplir des tâches qui nécessitent normalement des capacités intellectuelles. Les systèmes d'IA peuvent être conçus pour apprendre à partir de données, raisonner et prendre des décisions, percevoir et interagir avec l'environnement, et même faire preuve de créativité et d'émotions.

Les techniques d'IA incluent notamment l'apprentissage automatique (machine learning), le traitement du langage naturel (NLP), la vision par ordinateur, la robotique et l'automatisation cognitive. Les applications de l'IA sont nombreuses, allant de la reconnaissance de la parole et de l'image à la recommandation de produits et de services, en passant par la prédiction de comportements et la conduite autonome.



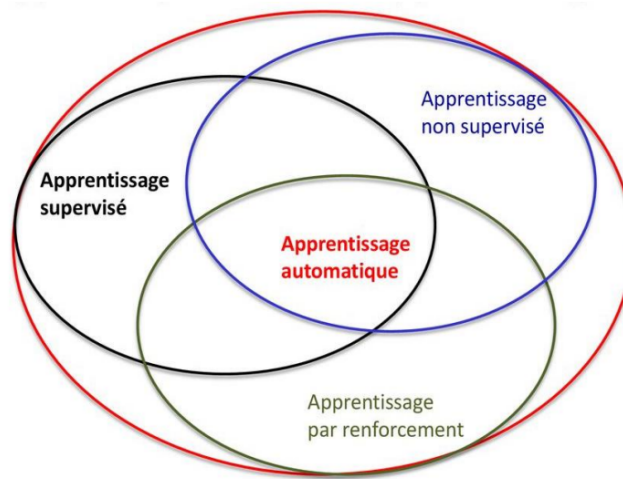
2.2 Machine learning

Définition

L'apprentissage automatique (Machine learning) est un domaine de recherche en informatique, qui implique des méthodes d'identification et de mise en œuvre de systèmes et d'algorithmes que les ordinateurs peuvent apprendre. Ce domaine est généralement associé à l'intelligence artificielle, plus spécifiquement l'intelligence computationnelle.

2.2.1 Types d'algorithmes d'apprentissage automatique

Il existe trois types d'algorithmes d'apprentissage automatique :



1. Apprentissage supervisé :

L'apprentissage supervisé est la tâche d'apprentissage automatique la plus simple et la plus connue. Il est basé sur un certain nombre d'exemples pré classifiés, dans lesquels est connu à priori la catégorie à laquelle appartient chacune des entrées utilisées comme exemples. Dans ce cas, la question cruciale est le problème de généralisation, après l'analyse d'un échantillon d'exemples, le système devrait produire un modèle qui devrait fonctionner pour toutes les entrées possibles.

L'ensemble de données pour l'entraînement, est constitué de données étiquetées, c'est-à-dire d'objets et de leurs classes associées. Cet ensemble d'exemples étiquetés constitue donc l'ensemble d'apprentissage.

Afin de mieux comprendre ce concept, prenons un exemple : un utilisateur reçoit chaque jour un grand nombre d'e-mails, certains sont des e-mails d'entreprises importants et d'autres sont des e-mails indésirables non sollicités ou des spam.

Un algorithme supervisé sera présenté avec un grand nombre d'e-mails qui ont déjà été étiquetés par l'utilisateur comme spam ou non spam. L'algorithme fonctionnera sur toutes les données étiquetées, faire des prédictions sur l'e-mail et voir si c'est un spam ou non. Cela signifie que l'algorithme examinera chaque exemple et fera une prédiction pour chacun pour savoir si l'email est un spam ou pas. Pour la première fois, l'algorithme fonctionne sur toutes les données non étiquetées, la plupart des e-mails seront mal étiquetés parce qu'il peut fonctionner assez mal au début. Cependant, après chaque exécution, l'algorithme compare sa prédiction au résultat souhaité (l'étiquette). En même temps, l'algorithme apprendra à améliorer ses performances et sa précision.

Dans l'exemple que nous avons utilisé, nous avons décrit un processus dans lequel un algorithme apprend à partir de données étiquetées (emails qui ont été catégorisés comme spam ou non-spam).

Dans certains cas, le résultat n'est pas nécessairement discret et il se peut que nous n'ayons pas un nombre fini de classes dans lesquelles classer nos données. Par exemple, nous essayons peut-être de prédire l'espérance de vie d'un groupe de personnes en fonction de paramètres de santé préétablis. Dans ce cas, comme le résultat est une fonction continue (nous pouvons spécifier une espérance de vie comme un nombre réel exprimant le nombre d'années que la personne devrait vivre), nous ne parlons pas d'une tâche de classification mais plutôt sur un problème de régression.

Il existe plusieurs algorithmes d'apprentissage supervisé qui ont été développés pour la classification et la régression. Parmi tous, les arbres de décision, les règles de décision, les réseaux de neurones et les réseaux bayésiens.

2. Apprentissage non supervisé :

Le deuxième type d'algorithmes d'apprentissage automatique est appelée apprentissage non supervisé, dans ce cas, nous n'étiquetons pas les données au préalable, nous laissons plutôt l'algorithme arriver à sa conclusion.

Les algorithmes d'apprentissage non supervisé sont particulièrement utilisés pour les problèmes de clustering. Dans les problèmes de clustering, étant donné un ensemble d'objets, nous espérons comprendre et montrer la relation entre eux. Une méthode standard consiste à définir une mesure de similarité entre deux objets, puis à rechercher tout groupe d'objets plus similaires les uns aux autres par rapport aux objets d'autres clusters.

Par exemple, dans les e-mails de spam / non-spam précédents, l'algorithme peut être en mesure de trouver du contenu commun à tous les e-mails de spam (par exemple, des mots mal orthographiés).

Bien que cela puisse fournir une meilleure classification que la classification aléatoire, ce n'est pas si facile à séparer le spam / non-spam.

3. Apprentissage par renforcement :

L'apprentissage par renforcement (RL) est une catégorie d'apprentissage automatique qui utilise des essais et des erreurs. Comparé à l'apprentissage automatique supervisé ou non supervisé, la RL est une méthode d'apprentissage plus axée sur les objectifs.


L'apprentissage par renforcement est un moyen puissant de résoudre les problèmes commerciaux sans grands ensembles de données historiques de

formation, car il utilise des modèles dynamiques avec des récompenses et des pénalités. Les modèles d'apprentissage par renforcement sont tirés des techniques interactives supervisées et non supervisées sont des méthodes complètement différentes qui peuvent apprendre de l'histoire pour prédire l'avenir.

Les modèles d'apprentissage par renforcement utilisent des mécanismes de récompense pour mettre à jour les comportements des modèles (résultats) en fonction des commentaires des comportements précédents (récompenses ou punitions). Le modèle ne sait pas quelle action entreprendre, mais essaie différentes options pour trouver l'action la plus utile.

Un modèle d'apprentissage par renforcement (« agent ») interagit avec son environnement pour choisir une action, puis passe à un nouvel état dans l'environnement. Lors de la transition vers le nouvel état, le modèle reçoit une récompense (ou une punition) associée à son action précédente. L'objectif du modèle est de maximiser sa récompense, permettant ainsi au modèle de s'améliorer continuellement à chaque nouvelle action et observation.

La figure suivante résume les trois types d'apprentissage avec les problèmes connexes à résoudre :



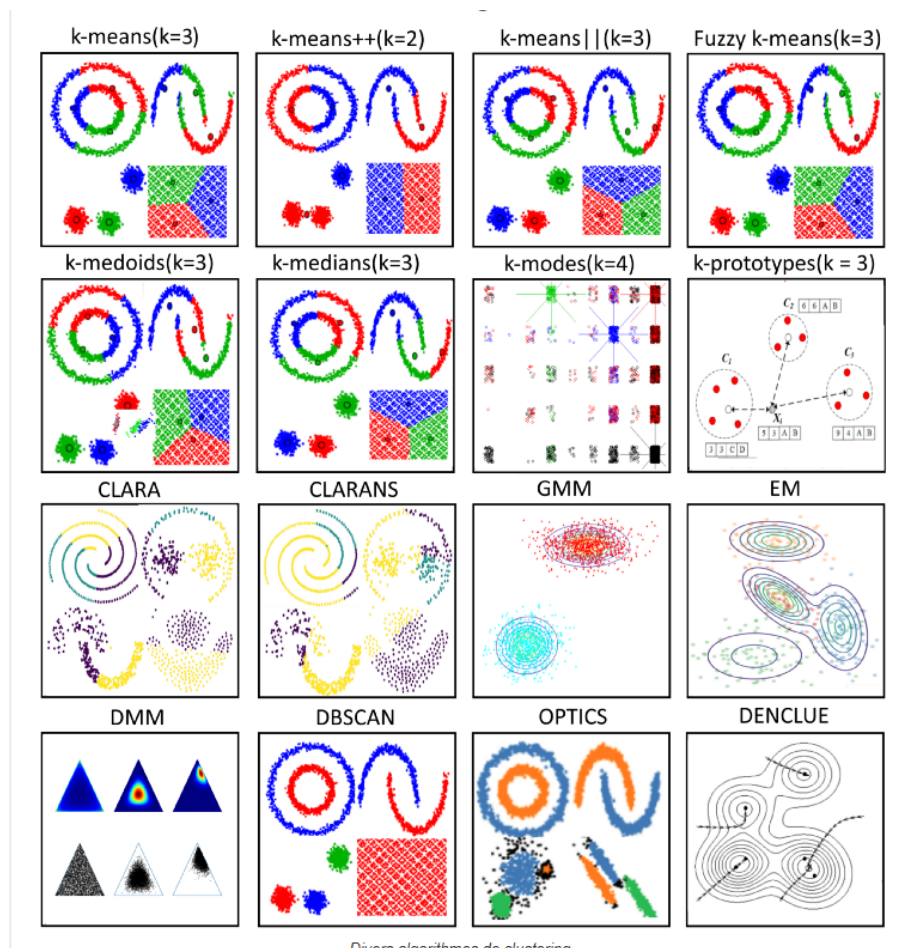
Apprentissage supervisé	<ul style="list-style-type: none"> • Classification • Régression
Apprentissage non supervisé	<ul style="list-style-type: none"> • Clustering • L'association
Apprentissage par renforcement	<ul style="list-style-type: none"> • Processus de décision • Système de recommandation

2.3 Clustering

Définition

clustering (ou regroupement en français) est une technique d'analyse de données qui consiste à regrouper des éléments similaires dans des ensembles, appelés clusters ou groupes. L'objectif du clustering est de trouver des similarités entre des données qui peuvent sembler différentes à première vue, et de regrouper ces données en fonction de leurs caractéristiques communes.

2.3.1 Algorithmes de clustering



2.3.2 Types d'algorithmes de clustering

Méthode de partitionnement crée d'abord un ensemble initial de k partitions, où le paramètre k est le nombre de partitions à construire. Il utilise ensuite une technique de déplacement itérative qui tente d'améliorer le partitionnement en déplaçant des objets d'un groupe à un autre. Les méthodes de partitionnement typiques incluent k -moyennes, k -médoids et CLARANS.

Méthode hiérarchique crée une décomposition hiérarchique de l'ensemble donné d'objets de données. La méthode peut être classée comme étant agglomérative (ascendante) ou divisive (descendante), en fonction de la façon dont la décomposition hiérarchique est formée. Pour compenser la rigidité de la fusion ou de la scission, la qualité de l'agglomération hiérarchique peut être améliorée en analysant les liens d'objets à chaque partitionnement hiérarchique (par exemple, dans Chameleon), ou en effectuant d'abord un microclustering (c'est-à-dire en regroupant des objets en « microclusters »), puis en opérant sur les microclusters avec d'autres techniques de regroupement telles que la relocalisation itérative (comme dans BIRCH).

Méthode basée sur la densité regroupe des objets basés sur la notion de densité. Il croît des clusters soit en fonction de la densité des objets de voisinage (par exemple, dans DBSCAN), soit en fonction d'une fonction de densité (par exemple, dans DENCLUE). L'OPTICS est une méthode basée sur la densité qui génère un ordre augmenté de la structure de clustering des données.

Méthode basée sur une grille quantifie d'abord l'espace objet en un nombre fini de cellules qui forment une structure de grille, puis effectue un regroupement sur la structure de grille. STING est un exemple typique de méthode basée sur une grille basée sur des informations statistiques stockées dans des cellules de grille.

Methodes	Caractéristiques
méthode de partitionnement	<ul style="list-style-type: none"> — Trouver des grappes de forme sphérique mutuellement exclusives — Basé sur la distance — Peut utiliser la moyenne ou le medoid (etc.) pour représenter le centre du cluster — Efficace pour les ensembles de données de petite à moyenne taille
méthode hiérarchique	<ul style="list-style-type: none"> — Le regroupement est une décomposition hiérarchique (c.-à-d. plusieurs niveaux) de corriger les fusions ou les splits erronés — Peut intégrer d'autres techniques comme la microclustration ou envisager des « liens » d'objet
méthode basée sur la densité	<ul style="list-style-type: none"> — Peut trouver des grappes de forme arbitraire — Les grappes sont des régions denses d'objets dans l'espace qui sont séparés par des régions à faible densité — Densité des grappes : Chaque point doit avoir un nombre minimum de points dans son « quartier » — Peut filtrer les valeurs aberrantes
méthode basée sur une grille	<ul style="list-style-type: none"> — Utiliser une structure de données de grille multirésolution — Temps de traitement rapide (généralement indépendant du nombre de objets de données, mais dépendant de la taille de la grille)

Mean Shift Clustering

Définition

Mean Shift Clustering est un algorithme de clustering non-paramétrique qui peut être utilisé pour identifier les groupes naturels de données dans un ensemble de données sans connaître à l'avance le nombre de clusters.

3.1 Méthode mean-shift

Il s'agit d'une méthode itérative permettant de trouver les modes d'une densité de probabilité à partir d'un ensemble d'observations. L'algorithme est défini comme suit :

3.1.1 Initialisation

Choisir un point de départ x_i dans l'espace de recherche. Estimation de la densité de probabilité

3.1.2 Calculer la densité de probabilité estimée

M-S est une méthode permettant de faire converger de manière itérative des observations x_i vers les modes de leur densité de probabilité f . Pour déplacer les observations vers les modes, il est nécessaire de calculer des estimation locales du gradient de la densité de probabilité $\nabla f(x)$. Ces estimations locales, aux points x , du gradient de la densité de probabilité f des observations x_i , sont

basées sur l'estimation non paramétrique de Parzen.

$$\nabla f(x) = \frac{1}{n} \sum_{i=1}^n \nabla K_H(x - x_i) \quad (3.1)$$

avec n le nombre d'observations, K une fonction appelée noyau qui est une fonction de densité de probabilité permettant d'en faire l'estimation locale, et H une matrice échelle carrée, symétrique et définie positive. En définissant H comme une matrice diagonale, les dimensions des observations sont normalisées indépendamment. Cependant rien n'oblige à choisir une telle matrice échelle, on peut imaginer utiliser des valeurs spécifiques $H_{ij} \neq 0$ afin de combiner certaines caractéristiques entre elles. Les noyaux K sont des fonctions de densité de probabilité centrées et éventuellement limitées à un support borné définies de R_p dans $[0; 1]$, p étant la dimension des observations x . Ce sont des fonctions positives, de moyenne nulle et dont les intégrales valent 1. K peut donc s'exprimer :

$$K_H(x) = |H|^{\frac{-1}{2}} \cdot K(H^{\frac{-1}{2}} x) \quad (3.2)$$

Les noyaux K sont généralement des fonctions sphériques. Par conséquent, ils peuvent être représentés par des noyaux monodimensionnels $k(x)$, $x \leq 0$, appelés profils. Ainsi, un noyau peut être défini en fonction de son profil k par :

$$K(x) = C \cdot k(x^T x) \quad (3.3)$$

avec C la constante de normalisation du noyau. L'équation du gradient de l'estimation de la densité de probabilité (3.1) peut alors s'écrire :

$$\nabla f(x) = \frac{2 \cdot C \cdot H^{-1}}{n \cdot |H|^{\frac{1}{2}}} \sum_{i=1}^n (x - x_i) \cdot k'((x - x_i)^T H (x - x_i)) \quad (3.4)$$

avec $k(x)$ la dérivée du profil k par rapport à x . En définissant $g(x) = k(x)$, et après quelques manipulations, cette expression devient :

$$\nabla f(x) = 2 \cdot C \cdot H^{-1} n \cdot |H|^{\frac{1}{2}} \sum_{i=1}^n g(d^2(x, x_i, H)) \left[\frac{\sum_{i=1}^n g(d^2(x, x^i, H)) \cdot x^i}{\sum_{i=1}^n g(d^2(x, x^i, H))} - x \right] \quad (3.5)$$

avec $d(x, x_i, H) = (x - x_i)^T H^{-1} (x - x_i)$ la "distance euclidienne normalisée". Elle est souvent appelée distance de Mahalanobis dans le contexte mean-shift, bien que H ne soit pas une matrice de covariance ni x une moyenne. Le premier terme de cette équation est proportionnel à l'estimation de densité en x . Le terme entre crochets est appelé vecteur mean-shift. Il décrit le déplacement dans la direction du gradient de la densité et exprimé sous une forme itérative, permettra de converger vers le mode local.

3.1.3 Calcul du vecteur Mean-Shift

Le vecteur Mean-Shift, noté $v(x)$, décrit la direction du gradient de la densité de probabilité au point x . Il est calculé en utilisant l'estimation locale du gradient de la densité de probabilité, notée $\hat{\nabla} f(x)$, obtenue grâce à la méthode de Parzen. Le vecteur Mean-Shift est défini par l'équation suivante :

$$v(x) = \frac{\sum_{i=1}^n k_H(x - x_i) \cdot x_i}{\sum_{i=1}^n k_H(x - x_i)} - x \quad (3.6)$$

où K_H est un noyau de densité de probabilité, x_i sont les observations et H est une matrice diagonale qui permet de normaliser les distances dans chaque dimension de l'espace de recherche.

Intuitivement, le vecteur Mean-Shift indique la direction à suivre pour se déplacer vers la région de densité de probabilité la plus élevée.

3.1.4 Mise à jour de la position

Mettre à jour la position en déplaçant le point x_i vers le mode local à partir du vecteur Mean-Shift : Après avoir calculé le vecteur Mean-Shift $v(x_i)$ à l'étape précédente, on peut mettre à jour la position du point de départ x_i en le déplaçant vers le mode local de la densité de probabilité. On utilise donc la formule :

$$x_{i+1} = x_i + v(x_i) \quad (3.7)$$

où x_i est la position courante du point de départ et $v(x_i)$ est le vecteur Mean-Shift calculé à l'étape précédente. Cette formule permet de converger itérativement vers le mode local de la densité de probabilité

3.1.5 Vérification de convergence

Vérifier si la position x_i a convergé vers un mode local. Si oui, arrêter l'algorithme. Sinon, revenir à l'étape 2. La vérification de convergence peut être réalisée de plusieurs manières.

Seuil de convergence : Une façon commune de vérifier la convergence est de calculer la distance entre la position actuelle x_{i+1} et la position précédente x_i . Si la distance entre ces deux positions est inférieure à un seuil prédéfini, alors l'algorithme peut être considéré comme convergé et peut être arrêté. Ce seuil peut être choisi en fonction de la précision souhaitée pour la convergence.

la vérification de convergence peut être réalisée en utilisant la distance euclidienne entre les positions actuelles et précédentes :

$$\|x_{i+1} - x_i\| < \epsilon \quad (3.8)$$

où $\|\cdot\|$ représente la norme euclidienne, ϵ est le seuil de convergence choisi et x_i est la position précédente de x_0 . Si cette condition est vérifiée, alors l'algorithme peut être considéré comme convergé. Il est possible de mettre en place d'autres vérifications pour améliorer la performance de l'algorithme de Mean-Shift. Par exemple :

Nombre maximum d'itérations : Si l'algorithme ne converge pas après un certain nombre d'itérations, il est peut-être temps d'arrêter la recherche de modes locaux.

Vérification de la stabilité : Il est possible de vérifier si les modes locaux identifiés sont stables, c'est-à-dire s'ils restent les mêmes même lorsque des pertur-

bations mineures sont introduites dans les données. Cette vérification peut être effectuée en modifiant légèrement les données et en vérifiant si les mêmes modes locaux sont toujours identifiés. Si ce n'est pas le cas, cela peut indiquer que les modes locaux identifiés ne sont pas fiables.

3.1.6 Détermination des modes

Cette étape consiste à regrouper les points convergents en clusters. Une fois que chaque point a convergé vers son mode local, les points qui convergent vers le même mode sont regroupés dans le même cluster. Les modes locaux représentent les centres de ces clusters.

Si deux points convergent vers le même mode, cela signifie qu'ils sont proches l'un de l'autre dans l'espace des données et qu'ils appartiennent donc au même cluster. En fin de compte, l'algorithme Mean-Shift produit un certain nombre de clusters, où chaque cluster est représenté par son centre (mode local) et un ensemble de points appartenant au cluster.

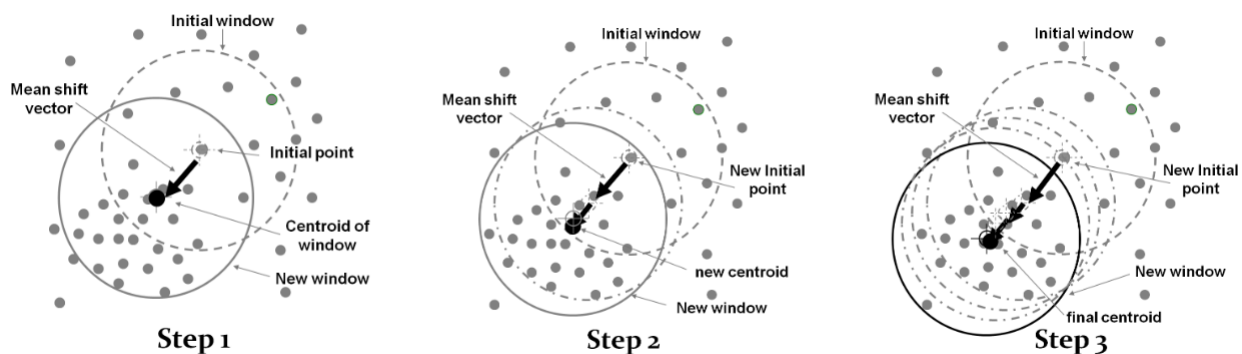


Figure 3.1 – Illustration, pour une observation donnée, du principe de la convergence M-S vers un mode

3.2 Implementation

Dans notre code on a utilise la bibliothèque NumPy, Matplotlib, Scikit-image et Scikit-learn pour effectuer la segmentation d'image en utilisant l'algorithme

Mean Shift.

Code :

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn.cluster import MeanShift, estimate_bandwidth
4 from skimage import io
5
6 # Charger l'image
7 img = io.imread('bird_small.png')
8
9 # Transformer l'image en une matrice 2D de pixels
10 w, h, d = tuple(img.shape)
11 image_array = np.reshape(img, (w * h, d))
12
13 # Estimer le rayon de bande optimal
14 bandwidth = estimate_bandwidth(image_array, quantile=0.1)
15
16 # Effectuer le clustering avec l'algorithme Mean Shift
17 ms = MeanShift(bandwidth=bandwidth, bin_seeding=True)
18 ms.fit(image_array)
19
20 # Obtenir les tiquettes de cluster pour chaque pixel
21 labels = ms.labels_
22
23 # Rorganiser les pixels en une image
24 clustered = np.reshape(labels, (w, h))
25
26 # Afficher l'image clusterise
27 plt.subplot(121)
28 plt.imshow(img)
29 plt.title('Image originale')
30 plt.subplot(122)
31 plt.imshow(clustered)
32 plt.title('Image clestring')
33 plt.show()

```

Voici comment le code fonctionne :

L'image '*bird-small.png*' est chargée à l'aide de la fonction *io.imread()* de la bibliothèque *Scikit-image*, et est stockée dans la variable *img*.

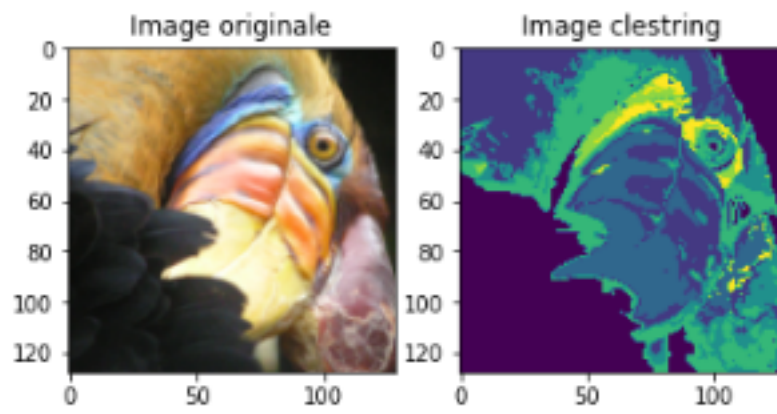
L'image est transformée en une matrice 2D de pixels en utilisant la fonction *np.reshape()* de la bibliothèque *NumPy*, et est stockée dans la variable *image array*.

Le rayon de bande optimal est estimé à l'aide de la fonction *estimate_bandwidth()* de la bibliothèque *Scikit-learn*, en passant *image array* comme données d'entrée et en utilisant le quantile 0.1 comme paramètre. La valeur estimée est stockée dans la variable *bandwidth*.

L'algorithme Mean Shift est appliqué en utilisant la classe *MeanShift()* de la bibliothèque *Scikit-learn*, en passant *bandwidth* comme paramètre de bande, et en activant l'option *bin seeding* pour améliorer la performance de l'algorithme. Le modèle est ajusté aux données d'entrée *image array* en utilisant la méthode *fit()*, et les étiquettes de cluster sont obtenues en utilisant l'attribut *labels* de l'objet *ms*.

Les étiquettes de cluster sont réorganisées en une image en 2D de même taille que l'image originale en utilisant la fonction *np.reshape()*, et sont stockées dans la variable *clustered*.

Finalement, les deux images, l'image originale et l'image clusterisée, sont affichées côte à côte en utilisant la fonction *plt.subplot()* et *plt.imshow()* de la bibliothèque *Matplotlib*, avec les titres appropriés pour chaque sous-figure, et sont affichées à l'aide de la fonction *plt.show()*.



Conclusion

La méthode Mean-Shift présente plusieurs avantages par rapport à d'autres méthodes de clustering. Tout d'abord, elle n'a pas besoin de connaître le nombre de clusters à l'avance, ce qui la rend particulièrement adaptée aux données dont la structure est inconnue. De plus, elle est robuste aux formes de clusters non convexes et peut trouver des clusters de forme arbitraire. Enfin, elle peut être utilisée avec des données de haute dimension.

Cependant, la méthode Mean-Shift peut également présenter certaines limites. Tout d'abord, elle peut être sensible aux paramètres de la méthode, tels que le choix du noyau et de la matrice d'échelle. De plus, elle peut être lente pour des ensembles de données volumineux et peut converger vers des minima locaux si plusieurs modes de densité de probabilité sont proches les uns des autres. Enfin, la méthode peut ne pas fonctionner correctement si les densités de probabilité ne sont pas bien estimées par la méthode de Parzen.