MATHÉMATIQUES ET INFORMATIQUE
Sciences
Université de Paris

Master 1 Informatique
Rapport de projet TER

INTERFACE POUR L'ENSEMBLE CLUSTERING

Ait Hammou Yanis – Sadeg Said

Année universitaire : 2019 – 2020

Encadrant : Lazhar Labiod

RESUME:

En raison de la croissance explosive des données issue du web, avènement des moteurs de recherche et des réseaux sociaux dans notre vie quotidienne, l'utilisation du maching learning est devenue primordiale pour pouvoir analyser ces données, notamment le clustering qui constitut l'une des tâches les plus importante pour le traitement des données.

Dans ce document nous allons aborder l'une des récentes approches de clustering qui est l'ensemble clustering, cette approche vienne en effet pour pallier les lacunes des algorithmes de clustering habituels et d'avoir des résultats supérieurs à eux dans de nombreux domaines

Ensuite nous allons présenter un algorithme de l'ensemble clustering que nous avons réalisé et implémenter un algorithme qui se base sur les preuves extraites de la matrice de co-association ainsi qu'une interface graphique qui permettra une utilisation simple et intuitive de cet algorithme

TABLE DES MATIERE

Table des matières

LISTE DES FIGURES

:

INTRODUCTION

1. Introduction

1.1 Contexte et motivations

L'ensemble clustering est méthode récente qui a pour but d'amélioré les approches classiques du clustering, cette méthode utilise des partitions issues des approches classiques de clustering pour produire un partitionnement final qui permettra d'éliminer les erreurs commises dans les partitions initiales produites par les algorithmes classiques. Notre travail consistera à réaliser une interface graphique qui facilitera l'utilisation de cette méthode et permettre à des personnes peu connaissantes dans le domaine du clustering d'utilisé cette méthode sans se soucier de l'implémentation de celle-ci.

1.2 Contributions et organisation du rapport

Notre rapport sera composé de trois parties, la première partie exposera les notions générales du clustering ainsi que de l'ensemble clustering et la deuxième partie sera consacré aux différentes méthodes et approches utilisées pour la réalisation de l'ensemble clustering, une troisième partie sera consacré à la présentation de notre implémentation et aux caractéristiques de notre interface.

LE CLUSTERING

2. Le clustering

2.1 Introduction

L'ensemble clustering et globalement les méthodes de clustering ont connu ses dernières années beaucoup de sucée, parmi les méthodes de machine learning, vue que ce sont des méthodes non supervisées qui n'ont pas besoin de données étiquetées, ce qui leurs permet d'explorer d'autres domaines que les méthodes non supervisées ne peuvent accéder ainsi que de réduire les coûts d'étiquetage des données.

2.2 Le clustering

2.2.1 Définition

Le clustering est méthode de classification non supervisé qui a pour but de regrouper les objets similaires dans le même cluster de façon à ce que les objets dans le même cluster soient le plus proches possible entre eux et qu'ils soient le plus loin possible des objets des autres cluster.

Il existe plusieurs approches pour la réalisation du clustering parmi lesquels on trouve le k-means, le clustering hiérarchique qui sont les algorithmes les plus connues, ainsi que d'autres algorithme tel que le DBSCAN et l'ensemble clustering.

2.2.2 Les algorithmes de clustering

2.2.2.1 Le k-means [1] :

Le k-means est l'un des approches les plus populaire du clustering c'est un algorithme qui se base sur l'affectation de chaque observation au cluster qui a la moyenne la plus proche, sont but est de minimiser la variance intra-classe (distance euclidienne au carrée).

L'algorithme de k-means est un algorithme itératif qui procède de la manière suivante :

Etant donnée un ensemble initial de k moyenne $m_1, m_2, \ldots, m_k$

Il affecte chaque observation au cluster dont le centre(moyenne) est le plus proche

$$S^{(t)} = \{x_p : \| x_p - m_i^{(t)} \|^2 \leq \| x_p - m_j^{(t)} \|^2 \; v_j \; 1 \leq j \leq k \}$$

puis il recalcul les moyenne (centroïde)

L'algorithme se converge lorsque les affectations ne changent plus.

L'algorithme de k-means se caractérise par sa simplicité mais pose le problème du choix du bon k

2.2.2.2 Le clustering hiérarchique [2] :

La classification ascendante hiérarchique est une méthode de clustering qui consiste à regrouper les individues qui sont les plus proches entre eux, il se repose sur le principe d'existence d'une mesure de dissimilarité entre les individues, comme la distance euclidienne ou la distance de Manhattan… etc.

la phase initiale de cet algorithme consiste à considérer que chaque individue constitue un cluster ensuite regroupe d'une manière itérative les individues les plus proches entre eux jusqu'à n'avoir qu'un seul cluster qui regroupe tous les individues ce qui constituera un arbre des clusters.

Ensuite il faut choisir à quel niveau on doit couper cet arbre pour former les clusters.

Cet algorithme pose le même problème que le k-means vue qu'on est obligé de choisir à quel niveau l'arbre doit être couper pour construire les clusters.

2.2.2.3 Le DBSCAN (density-based spatial clustering of applications with noise) [3]:

L'algorithme de DBSCAN est un algorithme de clustering apparue en 1996 qui est un algorithme basé sur le concept de densité entre les points dans l'espace, en général cette densité est estimée avec une distance généralement euclidienne.

Cet algorithme possède deux paramètres le premier est le nombre minimum de point que dois contenir un cluster et l'autre spécifie à quel degrés ces points doivent être proches pour appartenir à un cluster, c'est-à-dire quel est la limite de distance pour considérer que deux points sont voisins.

Cet algorithme considère les points qui ne sont pas proches d'aucun cluster et ne peuvent former un cluster comme des outsiders qui n'appartiennent à aucun cluster.

ENSEMBLE CLUSTERING

3. L’ensemble clustering

3.1 Introduction

L’ensemble clustering est méthode de clustering qui permet combiner plusieurs partitionnements de base en un partitionnement qui sera probablement meilleur que ceux de base, pour cela depuis plusieurs années cette méthode suscite de plus en plus d’intérêts chez la communauté scientifique.

3.2 Description de l’ensemble clustering :

L’ensemble clustering se décompose en deux étapes :

1. La génération des partitions : dans cette étape consiste à générer un ensemble de partitions depuis le data set en utilisant soit plusieurs algorithmes de clustering soit en utilisant un seul algorithme avec des initialisations de paramètres différents selon la méthode d’ensemble clustering utilisé, en plus dans cette étape on peut avoir soit des ensembles
2. La fonction de consensus : cette étape sera la responsable de la combinaison des différentes partitions obtenue lors du processus de génération afin d’obtenir une partition consensus qui résumera au mieux les informations issues des l’étape précédente, pour cela il existe deux types de fonction de consensus les fonctions basées sur la co-occurrence des objets ainsi que les fonctions basées sur la partition médiane.
   Dans la première approche, l’idée est de déterminer l’étiquettes du cluster auquel doit appartenir l’objet dans la partition consensus, pour ce faire on analyse combien

de fois chaque objet apparient à un cluster où combien de fois deux objets appartiennent au même cluster, en d'autres termes chaque objet doit procéder au vote pour déterminer auquel cluster final doit appartenir.

Dans la deuxième approche la partition de consensus est obtenue par la résolution d'un problème d'optimisation pour déterminer la partition que se reproche au mieux de toutes les autres partitions initiales.

3.3 Les approches de l'ensemble clustering [4]:

Plusieurs méthodes de consensus clustering existes et qui se différent les uns des autres par plusieurs aspects tel que la fonction de consensus où les caractères des partitions initiales :

3.3.1 Les Méthode basée sur le ré étiquetage et le vote :

cette méthode se décompose en deux sous problème, le problème de ré étiquetage des objets ensuite vient le processus de vote. Dans cette approche le label associé à chaque objet est symbolique et il n'existe pas de relation entre les labels donnés par un algorithme donné et les labels d'un autre algorithme. Pour le processus de vote il existe plusieurs techniques de vote tel que le plurality voting qui se pose sur un vote par majorité pour désigner un cluster gagnant pour chaque objet, le voting mergin qui propose de scheduler le vote pour ensuite faire des fusions des votes obtenue pour avoir un vote finale qui vas construite les clusters finaux, enfin il y'a technique de voting active cluster qui permet de combiné des cluster situé dans différentes locations, c'est-à-dire faire faire du clustering sur des partions des données dans différentes centres de calculs pour ensuite obtenir une partition consensus via un processus de vote.

3.3.2 Les méthodes basées sur la matrice de co-occurrence:

cette approche consiste à construite une matrice de co-occurrence de la manière suivante :

$$CAij = \frac{1}{m} \sum_{t=1}^{m} \delta(Pt(Xi), Pt(Xj))$$

Tel que Pt (X_i) est le label associé à l'objet X_i dans la partition Pt et δ(a,b) vaut 1 si a = b, 0 sinon, et *m* est le nombre de partitions initiales. Cette matrice peut être vue comme étant une mesure de similarité entre les objets, plus les ces objets appartiennent au même cluster plus ils sont similaires. En utilisant cette matrice comme mesure de similarité une partition consensus est obtenue en appliquant un

algorithme de clustering, en prenant par exemple un seuil de 0.5 pour regrouper les objets ayants un résultat supérieur à ce seuil dans le même cluster.

3.3.3 Les méthodes basées sur des graphes et hypergraphes :

Ces méthodes s'appuient sur la transformation du problème de combinaison de clusters en un graphe où un hypergraphe, pour cela il existe différentes façons de construire les graphes ou hypergraphes à partir des partitionnements initiaux, ainsi que différentes façons de découper ces graphes/hypergraphes pour construire le partitionnement consensus comme :

Cluster-based Similarity Partitioning Algorithm (CSPA): qui construit une matrice de co-occurrence à partir de l'hypergraphe, qui sera vue comme étant une matrice d'adjacence dans un graphe entièrement connecté. Pour construire le partitionnement consensus l'algorithme de partitionnement de graphe METIS est utilisé.

HyperGraphs Partitioning Algorithm (HGPA) : cet algorithme partitionne directement l'hypergraphe en éliminant un nombre minimal d'hyper-arrêtes, cela en considérant que tous les hyper-arrêtes ont un même poids et la recherche se fait en coupant le minimum possible d'hyper-arrêtes de façon à partitionner le graphe en k composantes connexes de même taille approximativement.

Meta-Clustering Algorithm (MCLA) : dans cette méthode, une mesure de similarité entre les clusters est définie en utilisant l'indice de Jaccard, puis une matrice de similarité entre les clusters est formée qui sera considéré comme une matrice d'adjacence d'un graphe construit en considérant les clusters comme des nœuds et la similarité entre les clusters comme des poids pour les arrêtes. Ensuite un partitionnement par l'algorithme METIS est utilisé pour former des clusters appelé méta-cluster, et finalement le nombre de fois où un objet appartient à un méta-cluster est calculé pour déterminer le partitionnement consensus.

Hybrid Bipartite Graph Formulation (HBGF) : cet algorithme représente les objets et les clusters dans un même graphe. Le graphe dans cette méthode est construit comme étant un graphe biparti où il n'existe pas d'arrête entre deux sommets s'ils sont tous les deux clusters où objets, et il n'existe d'arrêtes qu'entre un objet et le cluster auquel il appartient. Le partitionnement consensus est obtenue en utilisant l'algorithme METIS.

3.3.4 Les méthodes basées sur la distance de Merkin :

La distance de Merkin se définie comme suit :

Soit P_a ,P_b deux partitions du même jeu de données X et soit :

n_{00} : le nombre de pairs d'objets qui sont dans des clusters différents dans Pa et Pb
n_{01} ; le nombre de pairs d'objets qui sont dans différents clusters en Pa et le même cluster en Pb
n_{10} : le nombre de pairs d'objets qui sont dans le même cluster en Pa et dans différents clusters en Pb
n_{11} ;le nombre de paire d'objets qui sont dans le même cluster dans Pa et Pb.
La distance de Merkin est alors : M(Pa,Pb) = $n_{01} + n_{10}$, cela correspond alors au nombre de différences entre deux partitions . ensuite le problème de partitionnement médiane est défini comme suit :

$$P^* = \arg\min_{p\ Px} \sum_{j=1}^{m} M(P, Pj)$$

Il existe plusieurs heuristiques pour résoudre ce problème comme la *Best-of-k* (BOK) qui sélectionne la solution la plus proche du problème ci-dessus, ou la *Simulated Annealing One-element Move* (SOAM) qui consiste à deviner une partition initiale puis enlever des objets d'un cluster vers un autre dans le but de trouver un meilleur partitionnement et une méta-heuristique est utilisé pour converger vers un minimum local, l'autre heuristique est la *Best One-element Move (BOM),* qui est similaire à l'heuristique précédente mais qui à chaque fois qu'il trouve une meilleurs solution il l'utilise comme la nouvelle solution pour l'étape prochaine.

3.3.5 Les méthodes basées sur la théorie de l'information :

Dans ces méthodes la *category utility function U(Ph; Pi)* est utilisé comme mesure de similarité entre deux partitions : P*h* = { C_1^h , C_2^h , C_3^h, …C_d^h} et P*i* = { C_1^i , C_2^i , C_3^i …C_d^i} qui est définie comme suit :

U(Ph; Pi) = $\sum_{r=1}^{dh} (C_r^h) \sum_{j=1}^{di} (C_i^h|C_r^h)^2 - \sum_{j=1}^{di} (C_j^i)^2$

Tel que $(C_r^h) = \frac{|C_r^h|}{n}$ et $(C_i^h|C_r^h) = \frac{|C_i^h \cap C_r^h|}{C_r^h}$, et $(C_j^i) = \frac{|C_j^i|}{n}$

Ensuite le partitionnement consensus est défini comme étant :

$$P^* = \arg\min_{p\ Px} \sum_{i=1}^{m} U(P, Pi)$$

Et il a été prouvé que cette fonction U est équivalente à la minimisation de la variance inter-cluster ce qui peut être obtenue par l'application de l'algorithme K-means qui sera utilisé comme une heuristique pour la résolution du problème ci-dessus.

3.3.6 Les méthodes basées sur les algorithmes génétiques :

Ces méthodes utilisent les capacités de recherche des algorithmes génétiques le partitionnement consensus. Dans ces algorithmes les populations initiales sont créées généralement avec les partitions initiales de l'ensemble clustering ensuite une fonction de fitness est appliquée pour déterminer quel est chromosome (partitions d'objets initiales) qui se rapproche le plus du partitionnement recherché. Ensuite des étapes de croisement et mutation sont appliquées pour trouver des nouveaux descendants et renouveler la population, et durant ce processus si un des critères d'arrêt est atteint la partition qui a la plus grande valeur de fitness est choisi comme partition consensus. Plusieurs approches sont basées sur ces concepts comme l' *Heterogeneous Clustering Ensemble* où la population initiale est obtenue avec n'importe quel mécanisme de génération et le processus de reproduction utilise la fonction de fitness est le seul moyen utilisé pour décider si deux chromosome(partitions) vont survivre à l'étape suivantes ou pas. L'autre approche cherche la partition consensus en minimisant une fonction de critère de théorie de l'information avec un algorithme génétique.

3.3.7 Les méthodes basées sur Locally adaptive clustering algorithm(LAC):

Ces méthodes s'applique sur un ensemble d'objets numériques $X = \{x_1 , x_2, x_3 , .. x_n\}$ et produit un partition $P = \{ C_1 , C_2 , \ldots, C_n\}$ qui peuvent être aussi assimilé par deux ensembles $\{c_1 , c_2, .. , c_n \}$ et $\{w_1, w_2, \ldots, w_n \}$ où w_i et ci sont les centroïdes et les poids du cluster C_i . et l'ensemble de partitionnement $P = \{P_1, P_2, .., P_m\}$ est obtenue en appliquant *m* fois l'algorithme LAC.

Pour la fonction consensus plusieurs méthodes sont utilisé comme le *Weighty Similarity Partition Algorithm (WSPA)*, où dans pour chaque objet x_i sa distance avec un cluster C_t est calculée par :

$$d_{it} = \sqrt{\sum_{s=1}^{l} \quad Wts(Xis - Cts)^2}$$

et soit $D_i = \max_t\{d_{it}\}$ la distance maximal entre l'objet x_i avec tous les clusters , et $\rangle(C_t|x_i) = \frac{Di - dit + 1}{q.Di + q - \sum_t \quad dit}$ est la probabilité d'assigner un objet x_i à un cluster C_t , se qui donne vecteur de probabilité $\rangle_i = (\rangle(C_1|x_i), \rangle(C_2|x_i) , \ldots, \rangle(C_q|x_i))^T$

et pour calculer la similarité entre deux objets x_i et x_j on utilise :

$$cs(x_i,x_j) = \frac{i^T \, j}{\|i\|\|j\|}$$

Et ces similarités seront combinées dans une matrice S tel que $S_{ij} = cs(x_i,x_j)$ et on construit *m* matrice pour les *m* partitions de P

Enfin on calcul la somme des *m* matrices et on transforme la matrice résultante en un graphe auquel l'algorithme METIS est appliqué pour déterminer le partitionnement consensus.

3.3.8 Les méthodes basées sur la matrice de factorisation non-négatives :

Cette méthode est basée sur le problème de la factorisation d'une matrice non négative M en deux matrices non-négatives M H AB sachant que A et B soient non négatives.

Dans cette méthode une distance entre partitions est définie :

$$\lceil(P,P') = \sum_{i,j=1}^{n} \quad i,j(P,P')$$

Tel que $i,j(P,P')$ = 1 si x_i et x_j sont dans le même cluster dans la partition P et dans des différents clusters dans P' $i,j(P,P')$ = 0 sinon.

Et ainsi une matrice de connectivité est construite tel que :

$M_{i,j}(P_v) = \{1\ si\ \ Ct^V\ \ Pv, tel\ que\ xi\ Ct^V\ \ et\ xj\ Ct^V\ \ 0\ sinon$

Cela induit à ce que : $i,j(P,P') = |Mi,j(P) - Mi,j(p')| = (Mi,j(P) - Mi,j(P'))^2$

Et le problème de partitionnement consensus sera défini comme étant :

P^* = arg min $_{p\ Px}\ \frac{1}{m}\sum_{v=1}^{m}\ (P,Pv)$ = arg min $_{p\ Px}\ \frac{1}{m}\sum_{v=1}^{m}\ \sum_{i,j}^{n}\ (Mi,j(P) - Mi,j(Pv))^2$

Et on mettant $U_{i,j} = M_{i,j}$ (P*) comme étant la solution au problème d'optimisation, ce problème devient ainsi :

Min $_{\cup}\ \sum_{i,j=1}^{n}\ (Mi,j^{\sim} - Ui,j)^2$ = Min $_{\cup}\ \|M^{\sim} - U\|^2$

Où $M^{\sim} = \frac{1}{m}\ \sum_{v=1}^{m}\ Mi,j(Pv)$

Ensuite une méthode de factorisation sera utilisée pour déterminer une solution pour le problème ci-dessous.

3.3.9 Les méthodes basées sur le clustering flou :

Ces méthodes ont la particularité d'accepter comme entrées les partitions floues c'est-à-dire des partitions où les n'appartiennent pas qu'un seul cluster mais un objet peut appartenir à plusieurs clusters avec des degrés différents. Pour faire des modifications sont apportées dans les méthodes classique de l'ensemble clustering pour leurs permettre de traiter les clusters floues, parmi ces méthodes on trouve sCPSA qui change la manière dont il calcul la matrice de similarité entre les partitions, ainsi que sMLCA qui utilise une matrice de similarité basée sur la distance euclidienne au celle basée sur l'indice de Jaccard.

ALGORITHME

4. Algorithme

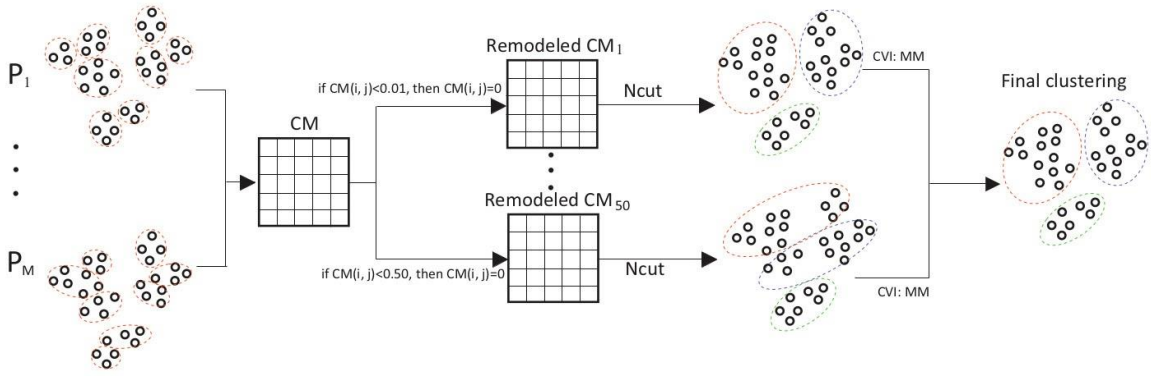

Figure. 1: Plan d'exécution de l'algorithme d'ensemble clustering

Dans la première étape de cet algorithme **figure.1,** plusieurs partitions de base sont générées en parallèle en utilisant l'algorithme KMeans de Spark, avec un nombre de clusters important, afin d'extraire les informations locales des clusters, ensuite ses informations seront insérées dans la matrice de co-association, où chaque valeur correspond à la fréquence d'apparition d'une paire d'objets dans le même cluster dans les

partitions de bases, donc on peut voir ça comme une transformation d'un noyau de base vers une matrice de co-association, tel que le montre la f**igure.2** ci-dessous.

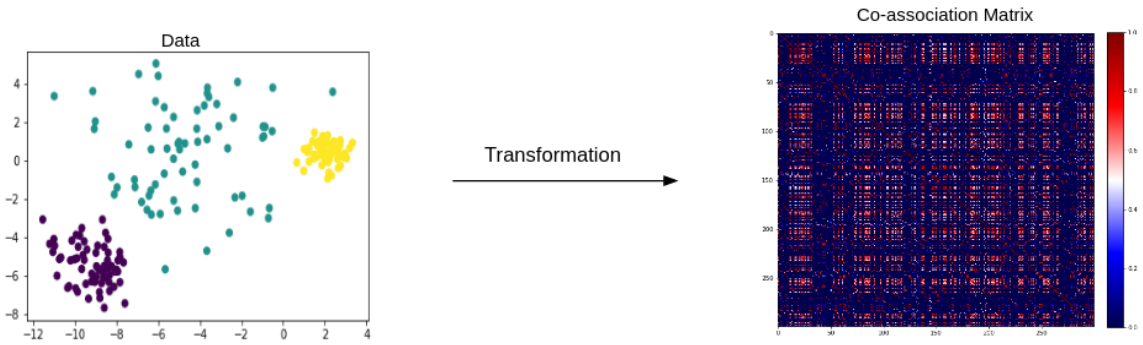

Figure. 2: Transformation d'un noyau de base vers une matrice de co-association.

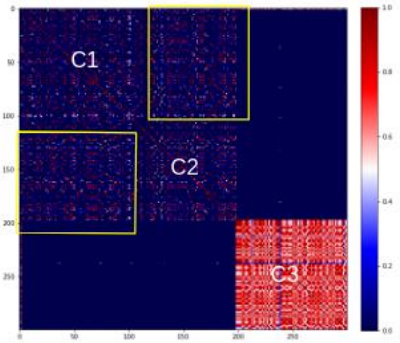

Figure. 3: La matrice de co-association réorganisée en utilisant la méthode **VAT**.

La **figure.3**, montre la matrice de co-association réordonnée en utilisant la méthode d'évaluation visuelle de la densité, on peut voir ici qu'il y a trois clusters, le cluster c3 est bien distingué des autres, et on remarque qu'il y a une confusion entre c1 et c2, ce qui fait que si on applique un modèle directement sur cette matrice, on risque d'avoir de mauvais résultats, car des erreurs sont introduites lors de la phase de transformation, ses erreurs sont dans les rectangles jaunes de la figure. Il faut donc les supprimer afin d'avoir des résultats plus précis, sur la **figure.4**, nous avons sur la droite les valeurs prises par les informations positives, c'est-à-dire les fréquences des paires d'objets qui appartiennent aux même clusters, et à gauche on voit les informations négatives c'est-à-dire les erreurs, ici on

suppose que les informations négatives correspondant aux fréquences qui ont des valeurs inférieures à 0,5, ce n'est pas toujours vrai car comme cette **figure.4**, donc, si on supprime ses erreurs directement c'est-à-dire les fréquences qui ont des valeurs en dessous de 0.5, on risque de supprimer beaucoup d'informations positives.

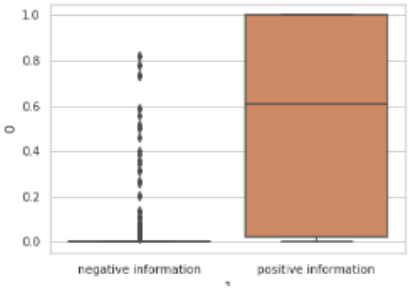

Figure. 4: boxplote correspond aux informations positives / négatives de la matrice de co-association.

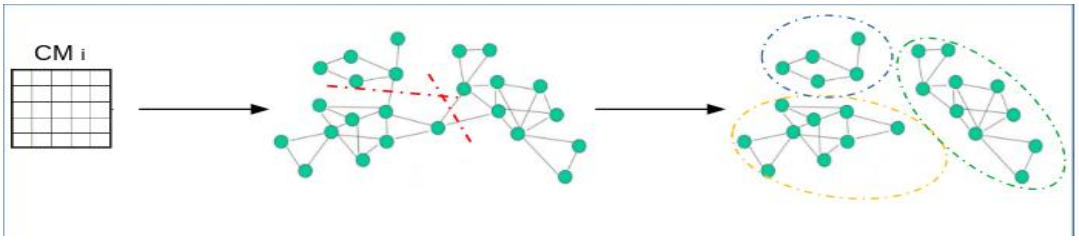

Figure. 5: l'algorithme de coupes normalisées est appliqué à chaque matrice de co-association générique

Pour remédier à ce problème, dans la deuxième étape de cet algorithme, nous allons effectuer des suppressions à plusieurs niveaux, en utilisant une boucle avec un seuil qui va de 0 à 0,5 avec un pas de 0,01, à chaque itération une matrice de co-association CMI est générée dont les fréquences qui ont des valeurs en dessous de ce seuil seront supprimés, puis l'algorithme de segmentation d'image est appliqué à chaque matrice générée CMI générée, comme le montre la **figure.5**, la matrice sera transformée en graphe pondéré, puis ce graphe sera divisé en plusieurs composantes connexes, à la fin de cette étape, nous aurons exactement 50 partitions candidates finales. Dans la dernière étape de cet

algorithme, une seule partition sera sélectionnée à l'aide d'une métrique d'évaluation interne qui utilise uniquement les informations de la matrice de co-association, la méthode que nous avons utilisée ici est basée sur le degré de confiance d'appartenance d'un objet à son cluster [5], cette méthode est proposée sous trois approches.

1. Le degré confiance moyenne d'affectation des objets aux clusters: cette première approche calcule pour chaque objet xi le degré de confiance de son appartenance aux clusters:

$$AC(P^*) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{conf}(\mathbf{x}_i)$$

Où P* est la partition à évaluer, n le nombre d'objets et conf(xi) est calculé comme suit:

$$\mathrm{conf}(\mathbf{x}_i) = \left(\frac{1}{|C_{P_i}|-1}\sum_{j:\mathbf{x}_j\in\{C_{P_i}\}\setminus\mathbf{x}_i} C_{ij}\right) - \left(\max_{1\le k\le K, k\neq P_i}\frac{1}{|C_k|}\sum_{j:\mathbf{x}_j\in C_k} C_{ij}\right)$$

où |Cpi| correspond au nombre de clusters, et la valeur de confiance est comprise entre -1 et 1, plus il est proche de 1 plus l'objet xi est considéré comme bien classé, respectivement plus elle est proche de -1 plus l'objet Xi et considéré comme mal classé, donc la meilleure partition correspond à celle dont le degré de confiance moyenne de ces objets est la plus élevée.

2. Le degré de confiance moyenne de m plus proches voisins: la deuxième approche est basée sur les m plus proche voisin, ici au lieu de calculer la moyenne des fréquences de xi avec tous les objets, seuls les m les plus proches du xi en matière de distance seront calculés, et m sera donné par l'utilisateur,

$$ANC(P^*,m) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\sum_{j:\mathbf{x}_j\in V(\mathbf{x}_i,C_{P_i},m)} C_{ij}}{|V(\mathbf{x}_i,C_{P_i},m)|} - \max_{1\le k\le K,k\neq P_i}\frac{\sum_{j:\mathbf{x}_j\in V(\mathbf{x}_i,C_k,m)} C_{ij}}{|V(\mathbf{x}_i,C_{P_k},m)|}\right).$$

3. Le degré de confiance moyenne de m plus proches voisins avec m calculé dynamiquement : cette troisième approche est une extension de la seconde, au lieu d'utiliser la même valeur m pour tous les clusters, elle sera calculée dynamiquement comme suit :

$$m_i = \left\lceil \alpha \sum_{j \in \{1,\cdots,n\} \setminus i} C_{ij} \right\rceil,$$

INTERFACE GRAPHIQUE

5. Interface graphique :

5.1 Introduction :

Dans le but d'expliquer et d'aider les personne non initié au clustering de comprendre le fonctionnement de de notre algorithme nous avons réalisé une interface simple et intuitive qui permettra de comprendre au mieux le fonctionnement de notre algorithme ainsi que de permettre de visualisé les résultats de l'exécution de cet algorithme sur différents datasets .

5.2 Description de l'interface :

Cette interface est composée principalement de trois parties :

Partie de paramétrage de l'algorithme : dans cette partie nous allons pouvoir choisir les différents paramètres de l'algorithme ainsi que de choisir le dataset à utiliser. Et on peut trouver dans cette partie :

1. Choix du dataset : ici l'utilisateurs peut choisir le dataset qui le convient soit dans 'ensemble des datasets existant que nous lui mettons à disposition(Moons, Circle

,Anisotropicity Distributed Data ,Data with varied variance ,Pathbashed, Agregation, Flame) où d'uploader son dataset pour qu'il soit traiter

2. choix des paramètres de l'algorithme : ici l'utilisateur pourra choisir les différents paramètres pour l'exécution de l'algorithme comme le choix du nombre des clusters, le nombre de partitions de base à générer, si le nombre de cluster dans chaque partition est fixé ou randomisé ainsi que la fonction de validation consensus à utiliser.

figure 6 : partie gestion de paramètres

Partie Graphe : après l'exécution de l'algorithme un graphe qui présentera les résultats seront affichés ce qui permettra de visualiser les données une fois mise dans leurs clusters.

Figure 7 : partie graph

Partie métrique d'évaluation : dans cette partie les résultats de l'évaluation des performances de l'algorithme seront affichés sous forme d'un graphique en barre et cela en utilisant les métriques suivantes :

Adjusted-Mutual-Information (AMI) : cette mesure sert à déterminer à quel point les informations réelles sont représentées dans le résultat du clustering.

Adjusted Rand Index (ARI) : cette métrique calcul la similarité entre deux partitions en comptant les pairs qui sont au même ou à des différents clusters dans les clusters réels et prédits.

V-mesure : représente la moyenne harmonique entre l'Homgeneity et la Completeness.

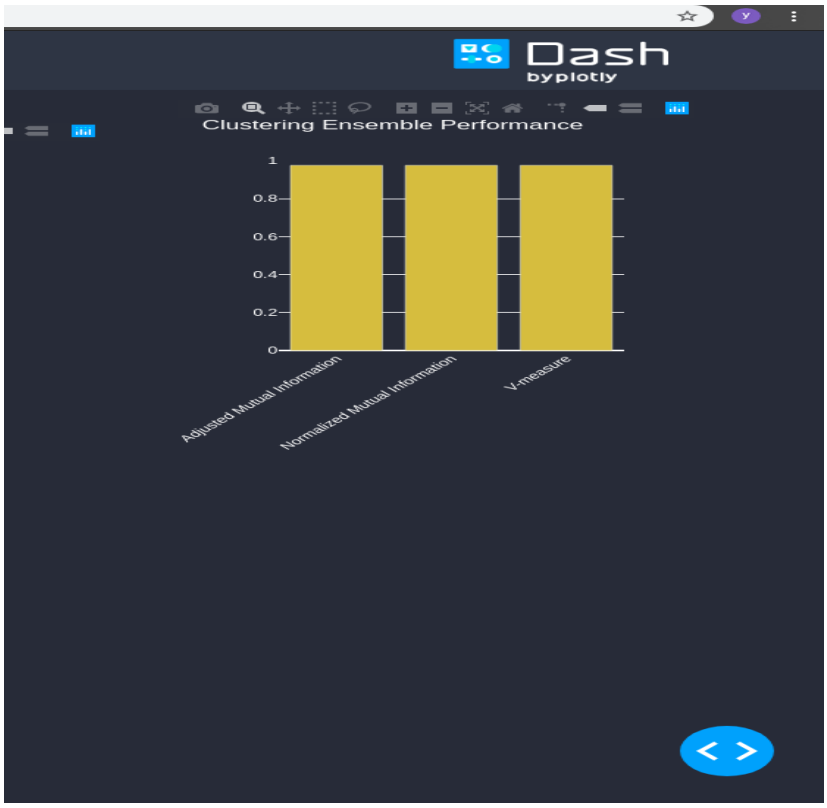

Figure 8 : partie métrique

Lorsqu'on exécute l'algorithme sur les dataset qui existent dans l'interface on retrouve les résultats ci-dessous :

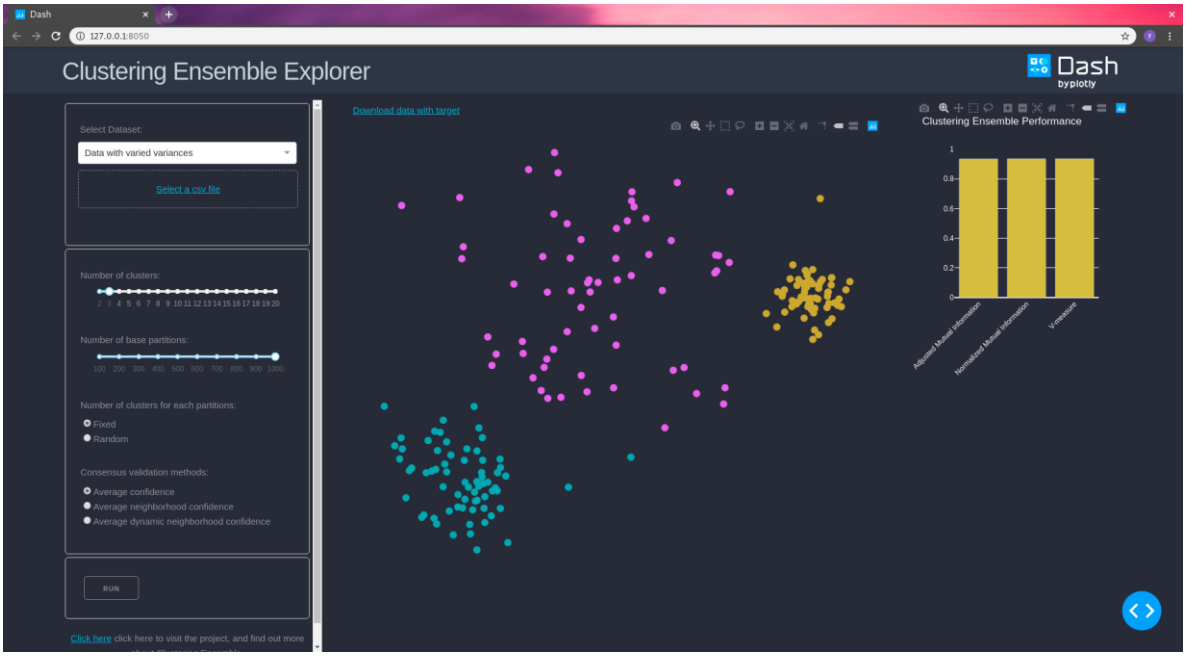

Figure 9 : test avec les données Data With varied variance

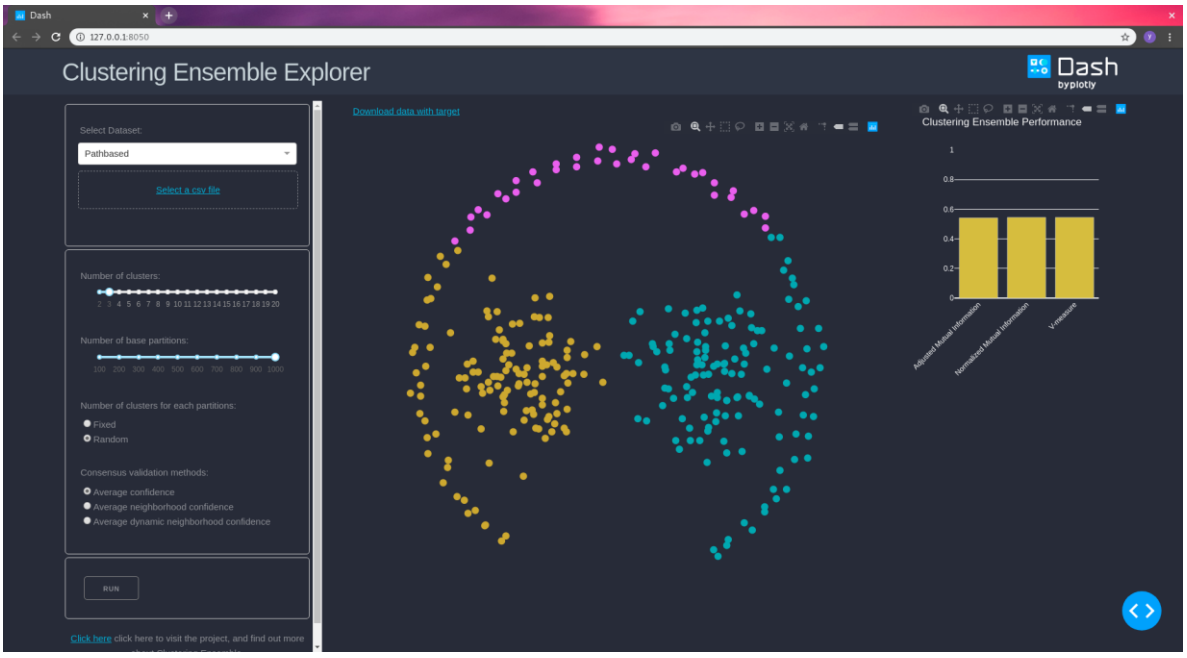

Figure 10 : test avec les données Pathbashed

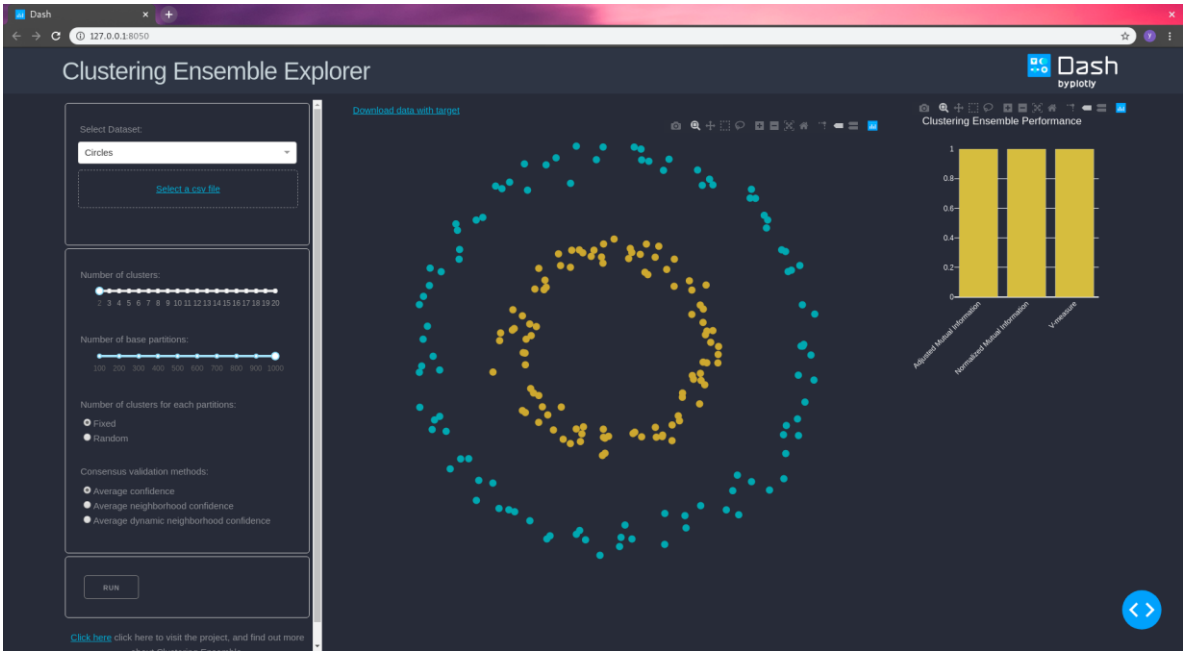

Figure 11 : test avec les données Circle

CONCLUSION GENERALE

6. Conclusion Générale

Dans ce projet nous avons eu l'occasion de travailler sur l'une des approches les plus récentes en clustering et en machine learning en général qui est l'ensemble clustering, cette approche qui a pour but d'amélioré les résultats de clustering habituels en générant plusieurs modèles de clustering et en utilisant ensuite une fonction consensus pour retrouver le modèle de clustering qui se reprochera le plus des modèles générer et qui produira une possible partition des données meilleurs que les modèles initiaux.

Nous avons aussi développé et implémenter un algorithme de l'ensemble clustering qui se base sur les preuves extraites de la matrice de co-association ainsi qu'une interface graphique simple et intuitive qui permet de tester et comprendre le fonctionnement de cet algorithme.

Enfin nous souhaitons par la suite de nos études poursuivre l'exploration des différentes approches de clustering et ainsi pouvoir apprendre plus sur ce domaine et pouvoir par la suite développer d'autre approches de clustering.

REFERENCES

:

7. Références

[1] https://en.wikipedia.org/wiki/K-means_clustering

[2] https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec

[3] Martin Ester ,Hans-Peter Kreigel , A density-based algorithme for discovering clusters in large spatial databases with noise

[4] Sandro Vega-Pons , José Ruiz-Schulcloper , A survey of clustering ensemble algorithms

[5] João M. M. Duarte, F. Jorge F. Duarte, Ana L. N. Fred, Adaptive Evidence Accumulation Clustering Using the Confidence of the Objects' Assignments.