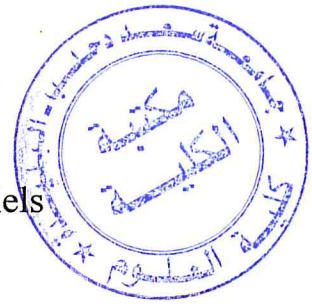
République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la recherche Scientifique

Université SAAD Dahlab-Blida USDB
Faculté des Sciences
Département d'informatique



Projet de fin d'études

En vue de l'obtention du

Diplôme de Master *en* Ingénierie des logiciels

Titre :

Etude et comparaison de l'algorithme k-means et ses variantes pour le clustering de texte

Présenté par :

M^elle^ Salma OULD ALI

M^elle^ Nawal NEMAS

Promoteur :

M^r^. Nacim Fateh CHIKHI

Soutenue le : 30 /06 /2013

Devant le jury :

Président : M^r^ HAMOUDA

Examinateur : M^r^ FERFERA

Examinatrice : M^me^ FARAH

Promotion: 2012/2013

REMERCIEMENTS

Nous tenons tout d'abord à exprimer nos plus sincères remerciements et notre profonde gratitude à notre promoteur M. CHIKHI Nacim Fateh pour nous avoir encadré tout au long de ce mémoire, pour son entière disponibilité, pour sa gentillesse, son soutien continu et pour nous avoir fait confiance ce qui a permis la réalisation de ce travail dans les meilleures conditions.

Nous adressons nos remerciements aux membres du jury pour avoir accepté d'évaluer ce travail et pour le temps consacré à la lecture de ce mémoire.

Enfin, nous remercions toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

Résumé

Dans cette thèse, nous nous intéressons à la caractérisation de grandes collections de documents afin de faciliter leur utilisation et leur exploitation par des humains ou par des outils informatiques.

Le clustering de textes est une méthode qui a pour objectif d'organiser un ensemble de données brut en groupes similaire. Plusieurs algorithme de clustering existent, dans notre recherche on s'intéresse particulièrement à l'algorithme k-means et ses variantes.

Nous avons ainsi étudie et comparé plusieurs variantes de l'algorithme K-means afin de déterminer celle qui est la plus adaptée au clustering de textes en termes de rapidité et de qualité de clustering.

Mots clés :

Fouille de données, *fouille de textee*, segmentation, algorithme K-means classique, K-means sphérique, K-means harmonique, bisecting K-means.

Abstract

In this thesis, we focus on the characterization of large collections of documents to facilitate their use and exploitation by humans or by computer tools.

Text clustering is a method that aims to organize a set of raw data into similar groups. Several clustering algorithm exist in our research we focus on the k-means algorithm and its variants.

In our work, we are interested in clustering texts. Specifically, we studied and compared several variants of the K-means algorithm to determine which one is best suited for the clustering of texts in terms quality of clustering.

Keys words:

Data mining, text mining, clustering; the K-means algorithm, Spherical k-means, harmonic k-means, bisecting k-means.

ملخص

نهتم بتوصيف مجموعات كبيرة من الوثائق لتسهيل استخدامها واستغلالها من قبل البشر أو عن طريق أدوات الكمبيوتر.

تجميع النص هو الأسلوب الذي يهدف إلى تنظيم مجموعة من البيانات الخام إلى مجموعات مماثلة حيث يوجد العديد من خوارزمية التجميع.

لقد درسنا مقارنة العديد من المتغيرات الخوارزمية لتحديد أيهم الأنسب لتجميع النصوص من حيث النوعية.

الكلمات المفتاحية :

استخراج البيانات - استخراج النص - تجميع البيانات - خوارزمية التجميع.

Sommaire

Introduction générale

Chapitre1 : le clustering de textes

Chapitre 2 : Algorithme k-means et ses variantes

Chapitre 3: Evaluation et comparaison

Liste des figures

Liste des tableaux

Introduction générale

Dans plusieurs domaines, de très grandes quantités de données textuelles sont générées. Le web est un exemple typique ou des centaines de milliers (voire des millions) d'articles sont publiés chaque jour. Cela est principalement dû au web 2.0 qui permet à n'importe quel internaute de publier des commentaires, des avis, des articles, etc. sur le web. La recherche scientifique est un autre exemple où un très grand nombre d'articles est publié chaque année à travers les conférences, les revues et les livres.

Afin de faciliter l'accès à ces énormes collections de documents, les chercheurs ont développé des outils pour l'organisation automatique de ce type de données. Le clustering de documents est un exemple de ces techniques qui est devenu récemment un domaine de recherche très actif. Plusieurs algorithmes de clustering de documents ont ainsi été proposés.

Les données textuelles représentent un type de données particulier qui se caractérisent par leur non structuration qui fait que beaucoup d'algorithmes de clustering ne soient pas adaptés à l'analyse de ce genre de données. Nous tentons dans ce travail de répondre à la question suivante : Quelle est la variante de l'algorithme K-means qui est la plus adaptée au clustering de textes ? L'algorithme K-means est l'un des algorithmes les plus populaires et les plus utilisés en datamining. Il figure parmi les 10 meilleurs algorithmes de data mining [ref – Top 10 data mining algorithms]. Bien que simple et efficace, l'utilisation de l'algorithme K-means n'est pas toujours évidente en raison du grand nombre des variantes de celui-ci. Plusieurs versions de cet algorithme existent en effet dans la littérature tel que le K-means sphérique, le K-means harmonique, bisecting k-means, etc.

Dans ce travail nous nous sommes intéressées à étudier et comparer plusieurs variantes de l'algorithme K-means afin de déterminer celle qui est la plus adaptée au clustering de textes en termes de qualité de clustering. La comparaison a été effectuée en utilisant des corpus textuels et des mesures classiques d'évaluation de clustering tel que la NMI (Normalized Mutual Information). Nous avons également étudié l'effet des techniques de pondération tel que TFIDF ainsi que l'impact de la LSI (Latent Semantic Indexing) sur les

différentes variantes de K-means. Enfin nous avons analysé la convergence des différents algorithmes.

Ce mémoire est organisé en trois chapitres.

Le chapitre 1 aborde le processus KDD (Knowledge Discovery in Databases) et ses différentes étapes en s'intéressant de près à l'étape de data mining. Le domaine de text mining et le clustering de textes y sont ensuite introduits.

Dans le chapitre 2 nous présenterons les méthodes de clustering en accordant une attention particulière aux méthodes basées sur le partitionnement. Nous décrivons en détail l'algorithme k-means et ses différentes variantes.

Le chapitre 3 décrit la méthodologie et l'environnement d'expérimentation qui ont été utilisés pour l'évaluation et la comparaison de K-means et ses versions. Il présente également les résultats expérimentaux obtenus lors de l'analyse des différents algorithmes par rapport à divers critères.

Enfin, une conclusion générale résume le travail réalisé et les principaux résultats obtenus et donne quelques pistes de recherche à poursuivre pour des travaux futurs.

Chapitre 1 : le clustering de textes

1 Introduction

Aujourd'hui, de grandes masses de données de divers types, sont collectées et stockées par les entreprises. Les méthodes classiques d'analyse de données s'avèrent incapables de répondre aux besoins actuels (analyser, résumer, et extraire des connaissances cachées à partir de données brutes). Une nouvelle génération d'outils permettant de travailler avec une telle quantité de données ont ainsi été développés ces dernières années : il s'agit de techniques de fouille de données (ou Data Mining) et de découverte de connaissances à partir de données (Knowledge Discovery in Data ou KDD).

La fouille de données consiste à rechercher et extraire de l'information utile et inconnue de gros volumes de données stockées dans des bases ou des entrepôts de données.

Le clustering est une des techniques offertes par le data mining permettant la création automatique de classe où chaque classe correspond à un groupe d'objets similaires.

2 KDD (Knowledge Discovery in Database)

2.1 Définition

Au vu de l'émergence de nouveaux champs d'application grâce au Data Mining, les experts du domaine travaillèrent à réunir plusieurs techniques afin de créer une méthode de recherche de connaissance intégrant toutes les étapes, du recueil des données à l'évaluation de la connaissance acquise. C'est ainsi qu'est né le terme d'Extraction de Connaissances à partir de Données (ECD), ou en anglais Knowledge Discover in Database (KDD).

La figure 1.1 décrit le processus d'ECD qui englobe le stockage et la préparation des données, l'analyse de celles-ci par différentes techniques, et enfin, l'interprétation et l'évaluation des connaissances acquises. Il est ici important de différencier les trois termes suivants:

- **Donnée:** valeur d'une variable pour un objet (comme le montant d'un retrait d'argent par exemple),

- **Information**: résultat d'analyse sur les données (comme la répartition géographique de tous les retraits d'argent par exemple),

- **Connaissance**: information utile pour l'entreprise (comme la découverte du mauvais emplacement de certains distributeurs).

Ainsi à l'aide du KDD, et à partir de données sur lesquelles nous ne faisons aucune hypothèse, il est possible d'obtenir des informations pertinentes, et de celles-ci, tirer des connaissances.

Certes, le Data Mining n'est qu'un rouage dans la formidable machine qu'est le KDD, mais il est sans conteste le cœur et le moteur de tous ce processus (Figure1.1).

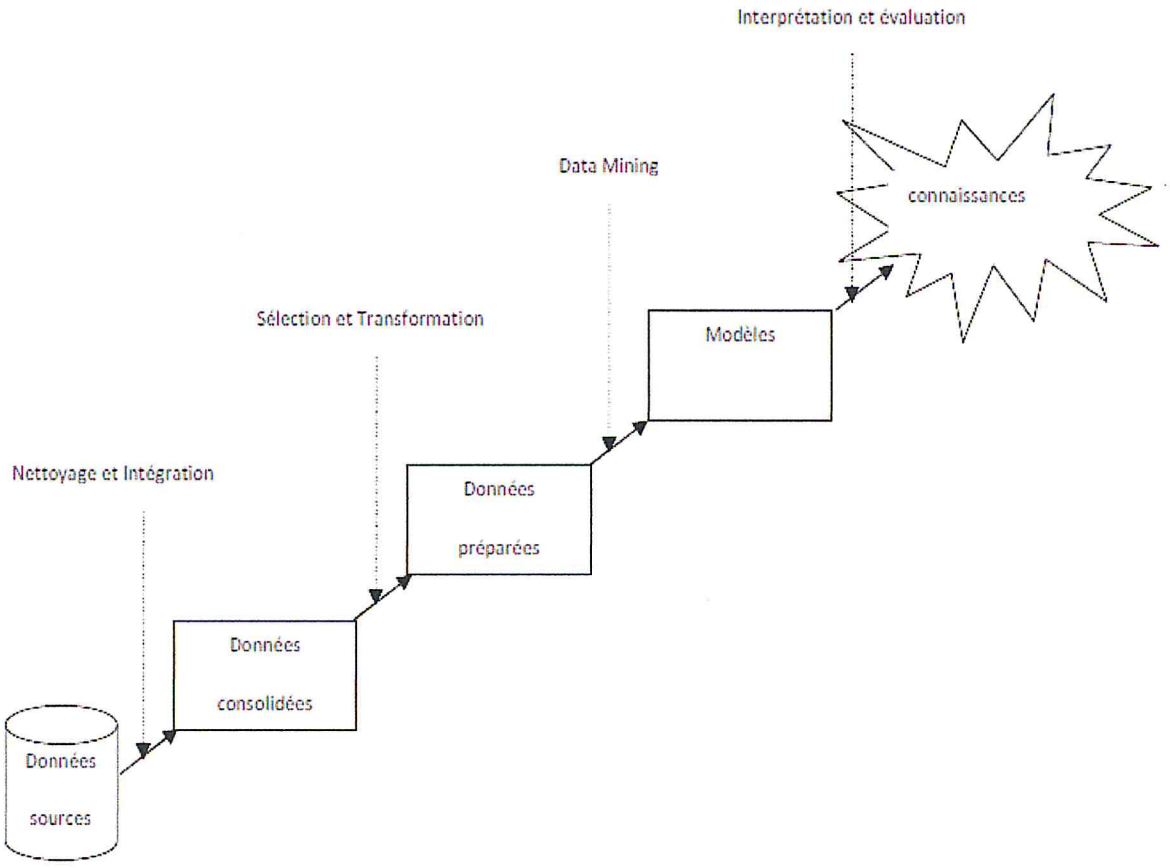


Figure 1.1 : Processus d'extraction des connaissances (ECD) ou knowledge discovery in databases (KDD).

2.2 Etapes du processus KDD

Le KDD est divisé en sept étapes essentielles:

a- Poser le problème

Dans cette étape, le but est d'exposer la problématique de la manière la plus claire possible de définir l'information recherchée ainsi que les objectifs escomptés (car un projet mal défini est une porte ouverte à des résultats biaisés et donc à un échec) et d'énoncer tout ce la sous une forme qui puisse être utilisé par les techniques et outils de modélisation.

b- La sélection (collecte de données)

La sélection consiste à choisir les données qui sont en accord avec les objectifs fixés. Généralement cette étape demande souvent la présence d'un expert qui épure les données pour ne garder que celles qui décrivent au mieux la problématique à résoudre (sélectionner les données pertinentes).

c- La préparation (nettoyage de données) [Maz 04]

Une faible qualité des données (erreurs de saisie, champs nuls, valeurs aberrantes) impose, généralement, une phase de nettoyage des données. Celle-ci a pour objectif de corriger ou de contourner les inexactitudes ou les erreurs de données. [**Lef 98**]

Parmi les types de valeurs pouvant créer des ambiguïtés:

- ✓ Valeurs aberrantes*:* ce sont les données qui ont des valeurs anormales par rapport à leurs natures (leurs types). Par exemple, un client qui a une date de naissance égale à l'année en cours. Un contrôle sur les domaines des valeurs permet de retrouver ces valeurs.

✓ Valeurs manquantes: terme utilisé pour désigner les champs qui n'ont aucune valeur. Il faut donc gérer de manière spécifique ces valeurs manquantes selon l'une des méthodes suivantes:

- Remplacer les données manquantes: par exemple par la moyenne, valeur héritée (95% des voitures ont quatre roues. Donc remplacer le nombre de roues, s'il n'est pas enregistré, par quatre).

- Exclure les enregistrements incomplets: ce choix est pénalisant car il réduit la base d'apprentissage.

- Gérer les valeurs manquantes: par exemple, considérer la valeur manquante comme facteur d'indécision et dupliquer la variable, comme le montre la figure **[Figure 1.2]**

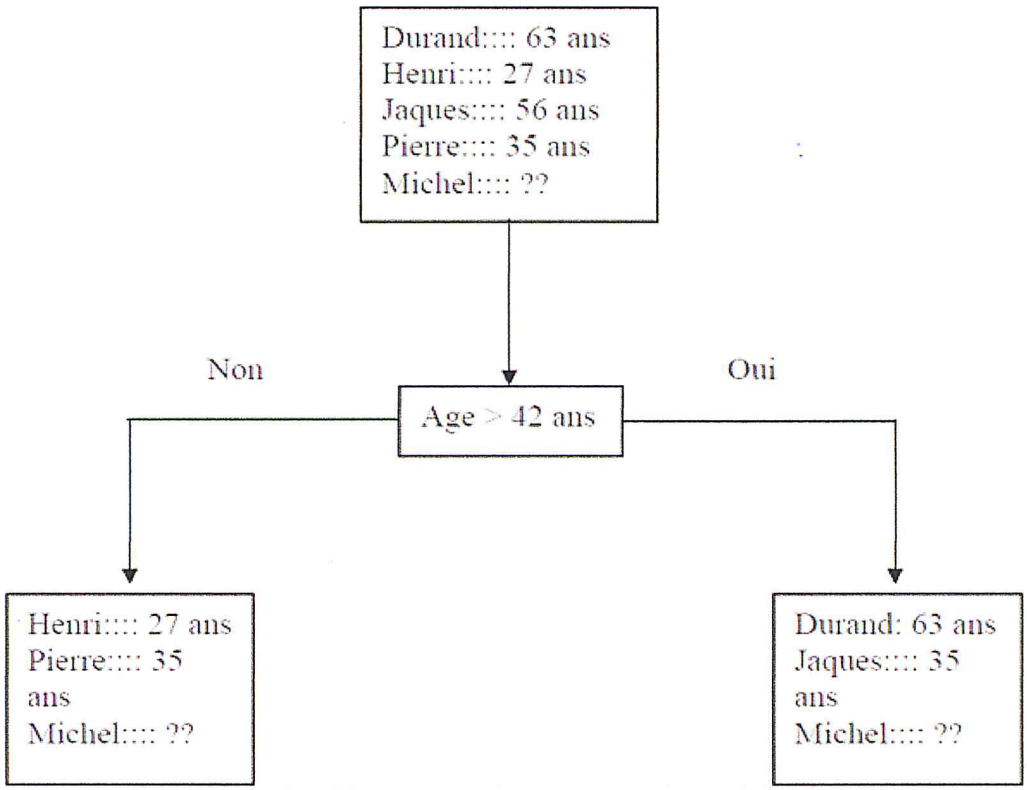


Figure 1.2 : Exemple de duplication d'une variable manquante [Lef 98]

d- La transformation (action sur les variables)

Bien que les données soient maintenant pertinentes et fiables, elles ne peuvent pas encore être directement exploitées par les outils de Data Mining. C'est pour cela qu'il est nécessaire de nous devons passer par une phase de transformation des variables.
Ces transformations se résument généralement par trois actions:

- Uniformisation d'échelle (normalisation): et cela afin d'éviter toutes disproportions dans les systèmes d'unité et de variables.

- Regroupement: fusionner plusieurs variables en une seule dans le cas où l'information unitaire est insignifiante.

- modification de type : des fois, un type a beaucoup moins de valeurs, du point de vue modélisation, qu'un autre (comme les dates et les durées). Il vaut mieux stocker l'ancienneté d'un client que la date de son premier achat ou de son dernier achat.

e- L'exploitation (modélisation)

Cette phase consiste à extraire la connaissance utile à partir d'un large volume de données et à la présenter sous une forme synthétique. Elle repose sur une recherche exploratoire, c'est à dire dépourvue de préjugés concernant les relations entre les données. **[Lef 98]**

Le modèle choisi dépend principalement du type du problème à résoudre et de la tâche (certaines techniques sont plus aptes à résoudre un certain type de tâches). Il faudra faire des compromis selon les besoins dégagés et les caractéristiques communes des techniques.

f- Evaluation du résultat

Maintenant que le modèle est construit, il faut évaluer ses performances et sa capacité déterminer correctement les résultats de nouvelles situations.

L'évaluation permet d'estimer la qualité du modèle qui a été construit, ainsi qu'à déterminer sa capacité à prévoir correctement les résultats de nouvelles situations.

Généralement on effectue des essais sur une base de données test prévue spécialement pourcela pouvant ainsi vérifier la pertinence des résultats donnés par le modèle.

g- L'intégration de la connaissance

Le modèle construit est intégré dans le processus de l'entreprise, c'est l'étape de la transition du domaine des études au domaine opérationnel **[Lef 98]**.

3 Data Mining

Le mot data mining se traduit en français par : « fouille de données »

3.1 Définition

Le Data Mining consiste essentiellement à extraire de l'information d'immenses bases de données de la façon la plus automatique possible. Plus concrètement, Souvent le terme data mining est employé comme synonyme de ECD.

Le Data Mining se situe à la croisée des statistiques, de l'intelligence artificielle et des bases de données.

Il existe dans littérature plusieurs définition du data mining :

- L'extraction de connaissances à partir des données est un processus non trivial d'identification de structures inconnues, valides et potentiellement exploitables dans les bases de données. **[Fay 96]**

- Le data Mining est la candidature d'algorithmes spécifiques pour extraire des modèles des données. **[Fay 96]**

- La découverte de nouvelles corrélations et liens par le traitement d'un large volume de données. **[Sup]**
- Le data Mining est l'art d'extraction des connaissances à partir des données.**[Mor]**

- Un processus qui permet de découvrir dans de grosses bases de données des informations jusque-là inconnues (connaissances), qui peuvent être utiles, et d'utiliser ces informations pour soutenir des décisions commerciales tactiques et stratégiques. **[Mat XX]**

- L'exploration et l'analyse de grandes quantités de données afin de découvrir des formes et des règles significatives, en utilisant des moyens automatiques ou semi-automatiques. **[Lin 97]**

- Le Data Mining est un processus d'extraction automatique d'informations prédictives à partir de grandes bases de données

- Un Processus inductif, itératif et interactif de découverte dans les BD larges de modèles de données valides, nouveaux, utiles et compréhensibles.

  - ✓ Itératif : nécessite plusieurs passes
  - ✓ Interactif: l'utilisateur est dans la boucle du processus.
  - ✓ Nouveaux: non prévisibles.
  - ✓ Utiles: permettent à l'utilisateur de prendre des décisions
  - ✓ Compréhensibles: présentation simple.

3.2 Raisons de développement du Data Mining

Le développement récent de la fouille de données (depuis le début des années 1990) est lie à plusieurs facteurs :

- ❖ une puissance de calcul importante est disponible sur les ordinateurs de bureau ou même à domicile ;

- ❖ le volume des bases de données a augmenté énormément ;

- ❖ l'accès aux réseaux de taille mondiale. Ces réseaux ayant un débit sans cesse croissant qui rendent le calcul distribué et la distribution d'information sur un réseau d'échelles mondiale viable ;

- ❖ la prise de conscience de l'intérêt commercial pour l'optimisation des processus de fabrication, vente, gestion, logistique, ...

- ❖ La fouille de données a aujourd'hui une grande importance économique du fait qu'elle permet d'optimiser la gestion des ressources (humaines et matérielles).

3.3 Tâches du Data Mining

Le choix des techniques de Data Mining à appliquer dépend de la tâche particulière à accomplir et des données disponibles pour l'analyse. La première étape consiste à traduire un objectif commercial en une ou plusieurs tâches.
Nous listons ci-dessous les tâches de base:

a- La classification

Consiste à examiner les caractéristiques d'un objet afin de l'affecter à une classe d'un ensemble prédéfini.

Les objets à classer sont généralement représentés par des enregistrements.
Des exemples de taches de classification sont :

- Attribuer ou non un prêt à un client
- Etablir un diagnostic.

b-L'estimation

Le résultat de l'estimation permet d'obtenir une valeur continue au lieu de discrète en fonction des caractéristiques d'objets. Exemple : déterminer le salaire d'une personne en fonction de l'âge, sexe, expérience, etc.…

c-La prédiction

Consiste à trouver une valeur future à partir des valeurs connues. Elle se base sur le passé et le présent, et le résultat se situe dans le Futur.

Exemple : Prédire les gagnants du championnat par rapport à une comparaison des équipes.

d-Règles d'associations

Consiste à déterminer les objets qui vont naturellement ensemble;
Exemple: la principale application est la recherche d'associations pour trouver les articles achetés ensemble.

e-Segmentation (Clustering)

Consiste à former des groupes homogènes à l'intérieur d'une population (l'ensemble des objets sur lequel on fait l'expérience où chaque objet représente un individu) homogène, contrairement à la classification les groupes ne sont pas préétablis.

Pour cette tâche il n'y a pas de valeur à estimer ou à prédire mais il s'agit de maximiser l'homogénéité à l'intérieur de chaque groupe et la minimiser entre les groupes c-à-dire : maximiser l'hétérogénéité entre les groupes. Une fois les tâches identifiées, elles sont utilisées pour restreindre la gamme des méthodes prises en compte. En termes généraux, le but est de sélectionner la technique de data Mining qui minimise le nombre et la difficulté des transformations de données qui doivent être effectuées pour produire de bons résultats. Les données brutes peuvent demander différentes manières d'être résumées, les valeurs manquantes doivent être traitées, etc. Ces transformations sont nécessairement indépendantes de la technique choisie.

3.4 Domaines d'utilisation du Data Mining

Les domaines d'application du Data Mining sont vastes et variés. Cependant un trait commun les lies ; c'est le fait qu'ils traitent un volume important de données et tirent des

informations qui visent à améliorer la qualité du produit ou du service. Parmi les domaines où l'utilisation du Data Mining est devenue monnaie courante:

a- Laboratoires pharmaceutiques et médicaux

La médecine représente un grand chantier pour les outils du Data Mining, notamment dans la détection des tumeurs et des maladies rares à partir des prélèvements d'autres maladies qui présentent des symptômes similaires dans le but d'identifier les meilleures thérapies.

b- Assurances

Le Data Mining est un outil très puissant pour l'analyse des sinistres et la recherche des critères explicatifs du risque ou de la fraude mais il fournit aussi des modèles de sélection et de tarification **[Lef 98]**.

c- Banques et les grandes administrations

Le Data Mining fournit des modèles de prédiction de fraudes ainsi que des modèles de pré autorisation de crédit automatique à partir des informations stockées dans leurs bases de données.

d- Automobiles et grandes industries

Afin d'améliorer la qualité du produit, on utilise le Data Mining pour anticiper les défauts de production. Par ailleurs, les méthodes du Data Mining servent pour la prévention des ventes.

e- Les transports à grandes échelles

On utilise le Data Mining pour l'optimisation des tournées et dans le Marketing afin de définir des programmes et des promotions selon les classes de clients et leurs caractéristiques.

f- Grande distribution et vente en correspondance

L'étude dans ce cas se base sur l'analyse des similarités des consommateurs en fonction de critères géographiques et sociodémographiques et l'analyse des comportements des individus.

4 Text Mining

Chaque jour, en particulier en raison de l'essor des communications électroniques, le nombre de documents disponibles croit de manière exponentielle et l'utilisateur (entreprise, organisme ou individu) se trouve submergé par la quantité d'informations disponible. Ces utilisateurs ne sont donc plus capables d'analyser ou d'appréhender ces informations dans leur globalité. Les documents textuels sont devenus prédominants sur le web et les informations utiles sont souvent enfouies. De nombreux travaux de recherche, notamment issus du web mining et du text mining s'intéressent au traitement de bases de documents textuelles.

4.1 Définition

Le text mining est l'ensemble des techniques et méthodes destinées au traitement automatique de données textuels, disponibles sous forme informatique, en assez grande quantité **[Tuf 07].**Bien qu'il y ait de nombreuses techniques et approches de text mining, le but global est simple ; il découvre des informations nouvelles et utiles contenues dans un ou plusieurs documents textuels **[Bil 08]** (à partir d'un document texte, un outil de text mining va générer de l'information sur le contenu du document. Cette information n'était pas présente ou explicite dans le document sous sa forme initiale, elle va être rajoutée, et donc enrichir le document).
Cela peut servir à :

- Classifier automatiquement des documents.
- Avoir un aperçu du contenu d'un document sans le lire.
- Alimenter automatiquement des bases de données.
- Enrichir l'index d'un moteur de recherche pour améliorer la consultation du document.

La particularité du text mining réside dans le mélange de techniques linguistiques et statistiques provenant du data mining, de l'apprentissage automatique, du traitement automatique du langage naturel, des statistiques et de l'extraction de connaissances **[Adj et Him 05]**.

Le processus de Text Mining s'effectue en trois étapes :

- Le prétraitement des données (découpage, nettoyage…),
- L'indexation ou représentation formelle (caractérisation de chaque document par des termes caractéristiques),
- L'analyse des données indexées : classement des documents par thèmes, recherche de relations…

4.2 Domaines liés au Text Mining

La recherche d'information et le text mining sont des domaines très liés, et partagent un certain nombre de caractéristiques, tel que les problèmes à traiter(le classement ou le clustering) ou le processus de traitement (le prétraitement de texte, l'extraction des termes, etc.).De mêmeils font souvent appel au domaine d'extraction d'information par exemple pour l'étape d'extraction des termes, la seule différence est la méthode ou bien le contexte de traitement : la recherche d'information se base sur les besoins de l'utilisateur (des requêtes utilisateur)pour classifier les documents, par contre le text mining fait le classement automatiquement sans besoin d'une requête utilisateur.

a- La recherche d'information

Le processus de recherche d'information a pour but la mise en relation des informations disponibles d'une part, et les besoins de l'utilisateur d'autre part. Ces besoins sont traduits de façon structurée par l'utilisateur sous forme de requêtes. La mise en relation des besoins utilisateur et des informations est effectuée grâce à un système de recherche d'information (SRI), dont le but est de retourner à l'utilisateur le maximum de documents pertinents par

rapport à ses besoins (et le minimum de documents non-pertinents). La notion de pertinence est difficile à automatiser, car elle est fortement subjective, c'est-à-dire dépendante de l'utilisateur. Le but du SRI est alors de faire correspondre au mieux la pertinence système avec la pertinence utilisateur. L'application la plus célèbre est les moteurs de recherches en ligne **[Bil 08] [Wei et al. 05].**

Le concept de base de la recherche d'information est la mesure de similarité, pour la comparaison. Même un petit ensemble de mots données en entré à un moteur de recherche peut être considérer comme un document qui va être comparé (s'il est bien assorti, apparié) avec d'autre documents **[Wei 05].**

Le processus de recherche d'information est composé de trois fonctions principales :

- ✓ l'indexation des documents et des requêtes ;
- ✓ l'appariement requête-document, qui permet de comparer la requête et le document ;
- ✓ la fonction de modification, qui intervient en réponse aux résultats obtenus.

La classification peut être considérée comme un problème de recherche d'information pour lequel les requêtes de l'utilisateur sont les classes dont lesquelles on cherche à classer les documents. Le clustering comme expliqué précédemment, consiste à créer a priori sur un ensemble de documents un ensemble de clusters. Pour une requête donnée, le système de RI va tout d'abord chercher le cluster le plus pertinent pour la requête puis trier les documents par ordre de pertinence dans ce cluster. Cette méthode permet de limiter la complexité de recherche dans les grandes bases de données et de renvoyer un sous-ensemble de documents tous pertinents pour la requête puisque le choix du cluster est évalué à partir des caractéristiques communes des documents qui le composent. C'est une technique qui permet habituellement d'augmenter la précision d'un système de RI **[Baeza et Rib 00].**

b- L'extraction d'information

L'extraction d'informations consiste à recueillir des connaissances contenues dans un document textuel. Il s'agit d'extraire des informations pertinentes répondant à des questions précises. L'extraction de connaissances fait souvent appel à une analyse linguistique et sémantique du texte comme dans le cas des systèmes de questions-réponses. Un exemple d'un tel système est le remplissage de formulaire prédéfini. Le système doit remplir chaque champ

du formulaire, à partir d'information émanant du document à analyser, sachant qu'à chaque champ est associé une question. Ce problème a fait surgir d'autres sous-problèmes tels que la reconnaissance et l'identification d'entités nommées. Les entités nommées sont généralement des noms de personnes, d'organisation, de lieux géographiques, des dates mais peuvent aussi être étendus a d'autre domaines tel que la biologie moléculaire où chaque nom de molécule est construit sur la base d'un schéma conceptuel. La reconnaissance et l'identification des entités nommées ont souvent fait appel aux techniques classiques du TAL (Traitement Automatique des Langues) basées sur l'utilisation de dictionnaires **[Taquechi, Collier, 2002].**

Une application de l'extraction d'information est la réalisation de résumé de documents. Cette tâche consiste à extraire, à partir d'un document, les phrases les plus caractéristiques du document. Le résumé automatique peut aussi être vu comme un problème de catégorisation (classement) ou chaque phrase d'un document doit être étiquetée selon les étiquettes pertinentes et non pertinent **[Hir 02].**

4.3 Le processus de Text Mining

Le processus de text mining est composé de plusieurs étapes que nous décomposons ici. Dans un premier temps, une collecte de documents doit être effectuée (c-à-dire : extraire les documents sur lesquels on va effectuer les traitements). Dans un second temps, le système doit faire un prétraitement sur ces documents textuels plus une extraction de termes (tous les documents textuels sont parcourus et pour chacun d'entre eux, une partie des termes (mot ou groupe de mots) est extraite (les termes les plus importants). Après la phase d'extraction de termes, les documents vont être représentés sous forme permettant de faire la classification facilement (généralement sous forme d'une matrice document*termes ou terme*terme). L'étape suivante est la classification des documents, que ce soit une classification supervisée ou non supervisée le but et de classer les documents selon le contenu dans des classes homogènes. Les classes obtenues doivent être évaluées et validées suivant certaines règles, et finalement interprétées visualisées.

a-La collecte des documents

La première étape de text mining est de collecter les documents. Parfois l'ensemble des documents peut être extrêmement important, on peut utiliser alors les méthodes d'échantillonnages pour sélectionner un ensemble gérable de documents pertinents **[Wei 05].**

Pour la recherche et le développement des technique de text mining, des données plus génériques peuvent être nécessaire (corpus). Souvent le corpus $Reuters^2$ est utilisé, mais il y a beaucoup d'autres corpus disponibles, utilisables qui peuvent être plus appropriés pour quelques études. Une autre ressource à considérer est le world wide web, qui permet de construire une collection de pages à partir d'un site particulier (Yahoo, Google …). Etant donnée la taille du web, les collections construites de cette façon peuvent être énormes, et le problème principal de cette approche pour construire une collection de documents c'est que la qualité de ces données est douteuse.**[Wei 05].**

b- L'identification des unités et des domaines d'information

Cette étape consiste à identifier, à partir des documents à analyser, les unités d'information (i.e. les traits descriptifs qui serviront d'ancrage à l'analyse des segments de documents) et les domaines d'information (i.e. les segments des documents).

L'identification des unités d'information consiste à déterminer et à extraire les éléments sur la base desquels les différents segments du corpus à analyser seront comparés. Dans le cas des données textuelles, ces unités d'information peuvent prendre différentes formes. Elles peuvent être des mots simples, des mots composés, des phrases, des n-gram etc. Cette étape repose sur des décisions théoriques importantes.

Le fait de fonder une analyse de données textuelles sur des mots ou sur le lexique d'un corpus impose d'importants questionnements sur la nature même d'un mot et, de manière plus générale, des unités d'information présentes au sein d'un texte. S'agit-il uniquement d'une suite de caractères? Un mot se définit-il par la séparation spatiale, l'identité morphologique, etc.? Ces questions fondamentales doivent nécessairement être abordées avant mêmed'envisager tout projet d'analyse et de gestion des documents textuels.

Techniquement, cette opération implique de soumettre les unités d'information à des considérations contradictoires, car elles se doivent d'être représentatives et discriminantes, tout en étant faciles à extraire. Il importe en dernière instance, de se rappeler que toutes les décisions théoriques prises par le chercheur doivent prendre en compte les buts poursuivis, et les résultats qu'il espère découvrir lors de ses recherches, car c'est à partir de ces éléments que la classification sera effectuée **[Mem 00].**

c- L'extraction de termes

Les programmes d'extraction de données terminologiques sont conçus pour trouver les termes contenus dans un texte. Ces programmes appelés extracteurs de termes ratissent un corpus et sont censés proposer à un utilisateur les termes qui s'y trouvent. Il est extrêmement difficile d'automatiser entièrement l'extraction des termes. Les extracteurs, même si de nombreuses améliorations sont apportées constamment, proposent à l'utilisateur des listes que celui-ci devra épurer ou enrichir **[L'Ho 04].**

d- La pondération des termes

Dans un corpus de documents, les termes n'ont pas le même pouvoir expressif. Il est souvent plus efficace de donner un poids plus important aux termes expressifs qu'aux autres termes. Toutefois, la problématique de cette approche réside dans la définition du critère permettant de mesurer le pouvoir d'expressivité des termes. La pondération des termes permet de mesurer l'importance d'un terme dans un document. Cette importance est souvent calculée à partir de considérations et interprétations statistiques (ou parfois linguistiques). L'objectif est de trouver les termes qui représentent le mieux le contenu d'un document.

De part son importance cruciale dans les performances des systèmes de traitement de données textuelles, la pondération des termes est devenue un domaine très actif de la recherche notamment dans le cadre de la recherche documentaire.

e- Les modèles de représentation des données

La représentation des données est l'une des principales étapes du processus de text mining. Cette représentation inclut souvent une matrice de relations entre les éléments. Dans la plupart des cas, cette matrice M est une matrice dite de similarité, c'est-à-dire que l'élément M (i. j) est une mesure de similarité entre le document i et le document j. D'autres types de matrices existent car il est possible de classer différents types de données.

Un autre type est la matrice de co-occurrence de termes des données de type textuel où l'élément M (i,j) représente le nombre de fois où l'élément i et l'élément j se retrouvent dans le même document (ou dans le même paragraphe, segment, etc.).

5 Corpus

L'ensemble des documents sur lequel porte la recherche est appelé le corpus. La notion de document est prise au sens large en recherche d'information. Elle comprend les documents textuels, les images, les graphiques, les sons ou toute combinaison de ces médias.

Pour créer un corpus, deux problèmes sont à considérer : l'homogénéité et la taille. La taille est caractérisée par le nombre de mots. A l'heure actuelle des gros corpus comprennent plusieurs centaines de millions de mots. Un corpus généralement écrit dans une langue, bien qu'il puisse être multilingue.

Les systèmes de text mining n'exécutent pas d'habitude leur algorithme d'extraction de connaissances sur des corpus non préparés ou bien pas encore prêts. Un effort considérable lors de l'extraction de connaissances est consacré à ce qui est généralement mentionné comme « le prétraitement de texte » qu'on va détailler par la suite **[Fel et San 07].**

6 La structure des documents

Le type de documents mis à disposition des utilisateurs évolue : du simple document texte Plat (les documents sont considérés comme une séquence de mots, un mot étant une séquence de caractères), on assiste aujourd'hui à la généralisation des documents structurés ou semi-structurés.

L'organisation des données se fait de plus en plus souvent dans des documents structurés réalisés au moyen de langages de structuration de documents plutôt que dans les bases de données à proprement parler. Deux langages ont tendance à s'imposer, à savoir le Standard Generalized Markup Language (SGML) qui est une norme de l'International Standardisation Organization (ISO) et l'eXtensible Markup Language (XML) qui est un dérivé plus récent de SGML **[L'Ho 04].**

Les documents structurés partagent beaucoup de points communs avec les bases de données lorsqu'ils sont utilisés pour organiser les termes. Nous énumérons ci-dessous les principales similitudes **[L'Ho 04] :**

- La description prévue est reprise systématiquement pour l'ensemble des entrées.

- Les données sont signalées et distinguées au moyen de champs.

- Toutes les données distinctes peuvent faire l'objet d'une exploitation comme une extraction ou une recherche.

La différence entre les documents structurés et les bases de données réside principalement dans le fait que c'est dans le texte lui-même et non des cellules fermées que les champs sont signalés. Ceux-ci sont marqués au moyen de balises **[L'Ho 04].**

7 Représentation des documents

Puisque les algorithmes de text mining opèrent sur les éléments ou les composants de texte, et pas les textes fondamentaux eux-mêmes, de nombreux éléments potentiels peuvent être utilisées pour représenter des textes **[Fel et San 07].** Les quatre types, suivants sont les plus utilisés généralement.

7.1 Représentation basée sur les caractères

Dans ce mode de représentation, l'accent est mis sur les séquences de caractères. Selon **[Fel et San 07]** on trouve **:**

Le caractère est l'élément de base pour la construction des éléments de plus haut niveau comme les termes ou les concepts, l'utilisation de cette représentation dans les applications de text ,mining reste toujours très limitée **[Fel et San 07].** L'un des modèles les plus connus est le modèle N-Gram **[Fel et San 07].** Dans ce modèle, le document est représenté par un ensemble de termes dont chaque terme est composé de N caractères (pour un alphabet de 26 lettres on obtient $26^2 = 626$ bigrammes ou $26^3 = 16\ 276$ trigrammes). On construit un modèle à partir de données d'apprentissage pour déduire la probabilité d'occurrence de telle suite de N-Gram dans un langage **[Chi et al. 00].**

7.2. Représentation basée sur les mots

Un mot est considéré comme une chaine de caractères comprise entre deux séparateurs (espace, virgule…). La sémantique de cette représentation est très basique **[Fel et San 07].**

7.3. Représentation basée sur les termes

Les termes sont des unités de forme et de contenu qui appartiennent au système d'une langue déterminée **[Cab 98]**. Selon **[Fel 87],** le contenu représente une notion définie dans un certain domaine du savoir.

Un terme peut être un mot unique ou un ensemble de mots (mot composé) **[Fel et San 07].** Dans le cas d'un mot composé, un terme n'est plus un mot unique mais un groupe de mots composé de un ou plusieurs mots. L'objectif de cette représentation est de tenir compte de l'aspect linguistique des termes. Par exemple, le groupe de mots pomme de terre désigne un sens particulier qui peut être perdu si on le décompose. Il est possible d'ajouter une information supplémentaire qui est la fréquence d'occurrences de chaque terme. Ces termes sont sélectionnés directement à partir du corpus ou un document à l'aide des méthodes d'extraction de termes. Ces méthodes peuvent convertir un ensemble de documents enun ensemble de termes normalisés (c'est-à-dire lemmatisé) **[Fel et San 07].** Ce niveau de représentation de document contient des termes qui sont obligatoirement contenus dans un document.

7.4 Représentation basée sur les concepts

Cette représentation tente de modéliser le sens induit par le document. Les modèles basés sur cette représentation utilisent des concepts pour désigner les différents sens présents dans le document. Les modèles sont, d'une part, plus riches et précis au niveau informationnel que les modèles morphologiques, et d'autre part, indépendants de la langue. L'inconvénient majeur est essentiellement l'extraction des concepts. Plus précisément, le problème réside dans la définition des concepts et dans la définition des relations et d'associations entre concepts et termes linguistiques. Les deux approches pour tenter de répondre au problème sont soit d'utiliser des connaissances linguistiques a priori, soit d'extraire les concepts statistiquement **[Fel et San 07].**

Contrairement à l'ensemble de termes, l'ensemble de concepts peut contenir des mots pas nécessairement trouvés dans le texte d'origine **[Fel et San 07].**

8 Classification automatique

La classification est un processus qui permet d'organiser des données en classes cohérentes ou homogènes. Elle s'applique apriori, sur n'importe quel type de données.

Les méthodes de classification ont donc un objectif précis : former des classes cohérentes (ou homogènes) et bien séparées. La cohérent veut dire que les éléments appartenant à une classe partagent de nombreuses caractéristiques communes et donc se ressemblent fortement. La séparation signifie que deux classes ne se ressemblent pas, c'est-à-dire qu'elles ne partagent pas les mêmes caractéristiques.

D'une manière générale, les problèmes de classification s'attachent à déterminer des procédures permettant d'associer une classe à un objet (individu). Ces problèmes se déclinent essentiellement en deux variantes. La classification dite :

- **supervisée** : souvent connue sous le terme de « classification »
- **Non supervisée** : souvent connue sous le terme «clustering».

9 Définition du clustering

Le clustering (classification non supervisée) consiste à regrouper un ensemble de données sous un ensemble de classes ou groupes appelés ≪cluster≫, c'est-à-dire que les objets sont groupés dans des classes homogènes disjointes. Ces classes ne sont pas prédéfinies par l'analyste mais découvertes au cours de l'opération. Pour faire ressortir les ensembles de documents, il faut maximiser l'homogénéité interne des classes et la dispersion entre elles **[Fel et San 07] [Wei et al. 05].**

10 Types de clustering

Il existe plusieurs types de clustering qu'on peut résumer dans ce qui suit :

Le clustering « $flou$ » (ou fuzzy clustering) fait référence aux algorithmes proposant des flus en sortie. Ce type de représentation est très riche mais peu exploitable sans post-traitement. L'approche de clustering flou la plus célèbre est l'algorithme des k-moyennes flou (ou Fuzzy-c-means) **[Did 84] [Mac 67].**

Les méthodes dites de clustering dur (ou hard clustering) pour lesquelles chaque objet est affecté à un unique groupe final. Les principaux algorithmes dans ce domaine sont les algorithmes hiérarchiques ou de partitionnement **[Did 84] [Mac 67].**

Comparativement aux clustering durs et flous, il existe relativement peu d'algorithmes de regroupement avec recouvrement (douce). La principale méthode générale est celle des « $cassesempiétantes$ »ou « $pyramides$ » appelé aussi « over lapping », dans laquelle un document appartient à un ou plusieurs clusters (mais pas tous) **[Did 84] [Mac 67].**

11 Etapes de clustering

Pour mettre en œuvre des méthodes de classification, il faut faire un choix d'un mode de représentation des documents, car il n'existe actuellement aucune méthode de clustering capable de représenter directement des données non structurées (textes). Ensuite, il est nécessaire de choisir une mesure de similarité et enfin, de choisir un algorithme de classification non supervisée **[Tuf 07] .**

* définir une représentation des documents.

* définir la mesure de similarité entre documents. C'est-à-dire la distance entre deux documents suivant le domaine d'application.

* la phase à proprement dite de clustering consiste alors à regrouper les documents similaires entre eux suivant l'algorithme choisi.

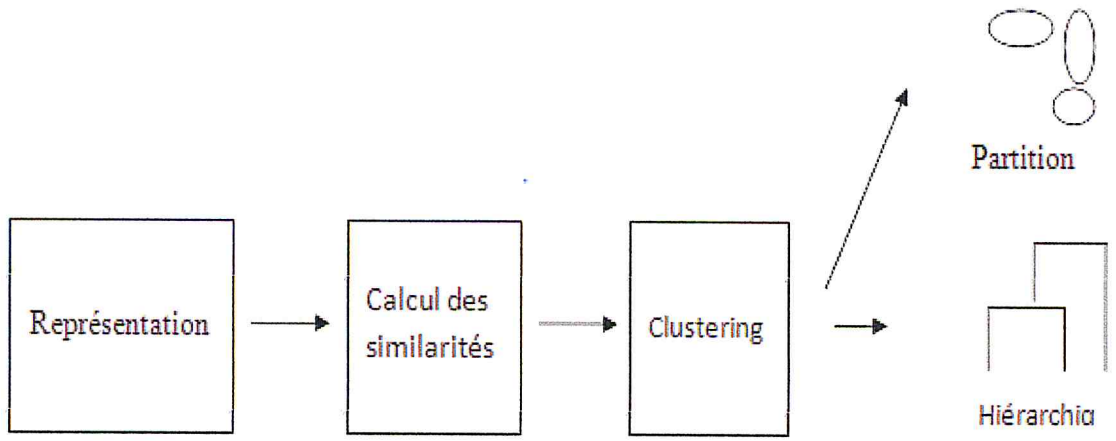


Figure 1.3 : schéma générale de la classification

12 Mesure de similarité et de distance

12.1 Mesure de similarité [Los 88]

Des textes qui se ressemblent contiennent les mêmes mots ou des mots qui apparaissent dans des contextes similaires.

Si on se place dans un espace vectoriel, des documents similaires correspondent à des vecteurs proches. Regrouper des vecteurs proches c'est trouver les vecteurs qui ont des directions semblables ou dont les extrémités sont proches.

Pour comparer deux documents, on regarde l'intersection de leurs lexiques spécifiques.

Les deux indices de similarité les plus connus sont : l'indice de Jaccard et le cosinus.

12.1.1 Indice de Jaccard

L'indice de similarité est le nombre de mots communs divisé par le nombre total de mots moins le nombre de mots commun **[Los 88]**.

$$S_f = \frac{m_c}{m_1 + m_2 - m_c}$$

m_c= nombre de mots en communs.

m_1= taille du lexique du document D1.

m_2=Taille du lexique du document D2.

$m_1 + m_2$= lexiquem_1 + lexiquem_2

Les vecteurs utilisés dans ce calcul de la similarité avec l'indice de Jaccard se fondent sur la présence/absence des mots.

12.1.2 Mesure de cosinus

Pour cette mesure, on utilise la représentation vectorielle complète, c'est-à-dire avec la fréquence des mots. Deux documents sont similaires si leurs vecteurs sont confondus.

Si deux documents ne sont pas similaires, leurs vecteurs forment un angle α dont le cosinus vaut **[Los 88]**

$$\cos\alpha = \cos(V_1,V_3) = \frac{V_1 \cdot V_3}{\|V_1\|\|V_3\|}$$

Exemple :

V1= $[a_1,a_2]$,

V3= $[c_1,c_2]$,

$$\cos(V_1,V_3) = \frac{a_1c_1 + a_2c_2}{\sqrt{a_1^2 + a_2^2}\sqrt{c_1^2 + c_2^2}}$$

12.2 Mesure de distance

Il s'agit de mesurer une distance séparant deux documents (ou vecteurs). Si deux documents (ou vecteurs) sont séparés par une faible distance alors ils sont similaires.

❖ Distance euclidienne **[Los 88]**

$$D(o_i, o_k) = \sqrt{\sum_{j=1}^{m}(o_{ij} - o_{kj})^2}$$

❖ Distance de Manhattan **[Los 88]**

$$D(o_i, o_k) = \sum_{j=1}^{m}|o_{ij} - o_{jk}|$$

13 But du Clustering

Le clustering (regroupement) des documents vise à mettre les documents similaires ensemble. En ce faisant, on veut atteindre un des buts suivants:

- Le nombre de clusters, par rapport au nombre de documents, est beaucoup plus petit. Ainsi, on peut accélérer le processus de recherche.

- Si un document est pertinent à une requête, alors les documents similaires ont plus de chance d'être pertinents aussi. Ainsi, le clustering peut être aussi vu comme un moyen d'expansion.

- Finalement, les réponses du système peuvent être regroupées, plutôt qu'être mises dans une liste individuellement. L'avantage de cette présentation de résultats est que l'utilisateur peut avoir une idée globale des résultats que le système a trouvés assez rapidement.

Avec le progrès rapide du matériel informatique, le premier avantage semble beaucoup moins important maintenant. Les deux autres restent toujours d'actualité.

14 Conclusion

Le clustering de documents est un domaine d'étude en plein essor en raison de la quantité d'information qui transite, notamment sur internet, et la valeur stratégique qu'elle revêt.

Le clustering est une tâche appliquée dans la vie courante. C'est un sujet de recherche actif qui se place au cœur du data mining, ce qui justifie pleinement l'intérêt qui lui est porté. Il a pour but l'organisation d'un ensemble de données (par exemple des documents) en classes homogènes. Le clustering recouvre l'ensemble des méthodes permettant la construction automatique de telles classifications.

Les méthodes de clustering se sont initialement développées d'un point de vue heuristique autour de méthodes optimisant des critères métriques. Ainsi les deux algorithmes les plus couramment utilisés sont, d'une part, l'algorithme des centres mobiles (ou k-means) pour la recherche de partition, et d'autre part, l'algorithme de classification hiérarchique pour la recherche de hiérarchies.

Dans le chapitre suivant, nous décrivons l'algorithme k-means ainsi que ses variantes.

Chapitre 2 : Algorithme k-means et ses variantes

1Introduction :

L'objectif du clustering (ou segmentation) est le suivant : on dispose de données non étiquetées et on souhaite les regrouper par données ressemblantes. Cette manière de définir intuitivement l'objectif de la segmentation cache la difficulté de formaliser la notion de ressemblance entre deux données.

Soit un ensemble X de N données décrites chacune par leur P attributs. La segmentation consiste à créer une décomposition de cet ensemble en groupes tel que :

Critère 1 : les données appartenant au même groupe se ressemblent.

Critère 2 : les données appartenant à deux groupes différents soient peu ressemblantes.

Avant de poursuivre, notons que le problème de segmentation optimale en *K* groupes est un problème NP complet **[Phi 06].** Il faut donc chercher des algorithmes calculant une bonne partition, sans espérer d'être sûr de trouver la meilleure par rapport aux critères considérés.

2Définition :

La segmentation est l'opération qui consiste à regrouper les individus d'une population en un nombre limité de groupes. Les segments ont deux propriétés : d'une part, ils sont prédéfinis, mais découverts automatiquement au cours de l'opération ; d'autre part, les segments regroupent les individus ayant des caractéristiques similaires (homogénéité interne) et séparent les individus ayant des caractéristiques différentes (homogénéité externe). Ces homogénéités peuvent être calculées par des critères tels que l'inertie intra-classe **[Ste 02].**

Comme la classification, la segmentation consiste à repartir les individus en groupes. Toutefois, cette répartition n'est pas effectuée en fonction d'un critère particulier et ne vise pas à rassembler les individus possédant la même valeur pour ce critère.

La définition de segments naturels est délicate car ils sont loin d'être toujours évidents. Les segments n'apparaissent pas toujours aussi naturels à posteriori, et peuvent d'ailleurs différer selon l'algorithme choisi pour les calculer.

Typiquement, un partitionnement sera jugé satisfaisant si on obtient en sortie de la méthode des groupes d'objets homogènes (c'est-à-dire possédant des objets les plus similaires possibles) et qui sont le plus hétérogènes possible entre eux. On réalise rapidement que

l'obtention d'une « bonne partition » du jeu de données (ensembles d'objets) n'est possible qu'à partir du moment où l'ensemble des objets de départ n'est pas « trop brut ». Dans le cas contraire, des procédures de traitement des données doivent être appliquées avant le lancement de toute méthode de clustering (recherche de corrélation dans les données, études statistiques des différentes variables,…). Souvent, la réussite de l'approche est étroitement liée à la phase de traitement des données.

3Algorithme de clustering

Il existe plusieurs familles d'algorithmes de clustering : les algorithmes conduisant directement à des partitions comme la méthode de classification autour de centres mobiles (cas particulier de la technique des nuées dynamiques ou des k-means),les algorithmes hiérarchiques ascendants (ou "agglomératifs") qui procèdent à la construction des classes par agglomérations successives des objets deux à deux fournissant une hiérarchie de partitions des objets, et les algorithmes descendants(ou "divisifs") qui procèdent par dichotomies successives des objets .

Il existe aussi d'autres algorithmes comme par exemple les algorithmes basés sur la densité qui utilisent des notions de connectivité et de densité ainsi que les algorithmes de grille basés sur une structure à multi-niveaux de granularité.

4Clustering par partitionnement

Les méthodes de partitionnement sont largement utilisées en raison de leur efficacité (rapidité du temps de calcul).La caractéristique principale de ce type d'algorithmes est que le nombre de clusters à identifier doit être fourni comme paramètre d'entrée.

Les méthodes de partitionnement, ou de classification directe, qui produisent de simples découpages ou partions de la population étudiée sont adaptées aux très grands ensembles de données(plusieurs milliers d'objets à classer).

Les algorithmes de partitionnement sont généralement itératifs, c'est-à-dire que plusieurs passes sont nécessaires pour obtenir une convergence de l'algorithme. La partition finale dépend de la partition initiale.

Dans ce type de clustering, la représentation d'un cluster C est définie comme étant la moyenne des éléments présents dans C, c'est le cas de centroïde ; un deuxième mode de

représentation des clusters consiste à prendre *k* individus parmi tous les individus d'un cluster. Ces *k* éléments sont centraux vis-à-vis de la classe, c'est-à-dire qu'ils sont proches du centre de la classe **[Cha 04].**

Il existe donc deux ensembles de méthodes de partitionnement :

Le premier regroupe les méthodes K-centroïdes telles que la méthode k-means, les méthodes fondées sur k-means ainsi que la méthode des centres mobiles ; la représentation d'un cluster C est définie comme étant la moyenne des éléments présents dans C. Plus précisément, un centroïde est un vecteur de termes pondérés dans lequel chaque composante correspond à la moyenne arithmétique des composantes d_i correspondantes de tous les vecteurs d'individus présents dans C comme le montre la (figure 2.1).

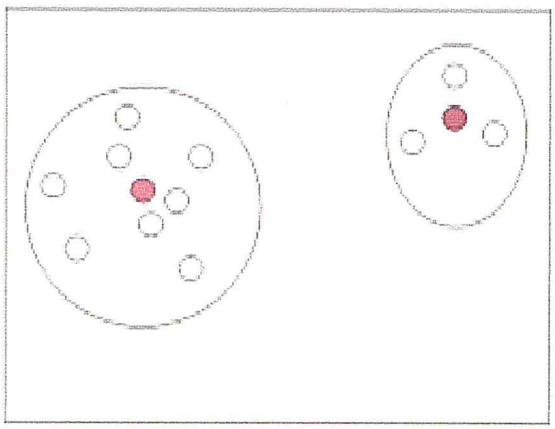


Figure 2.1 : Exemple de centroïde [Nik 06]

Le second regroupe les méthodes K-médoide telle que la méthode des nuées dynamiques comme le montre la (figure 2.2)**. [Nik 06].**

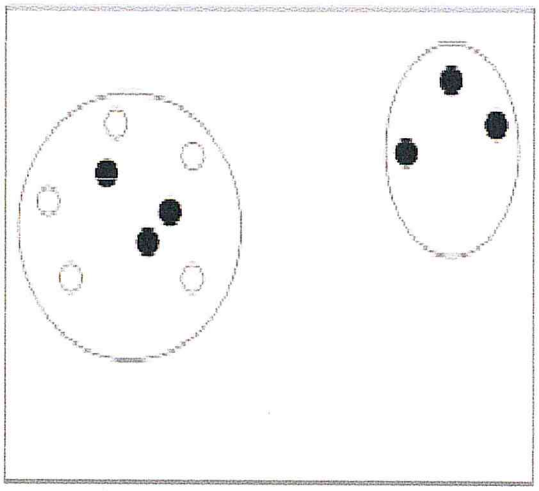


Figure 2.2 : Exemple de médoide [Nik 06].

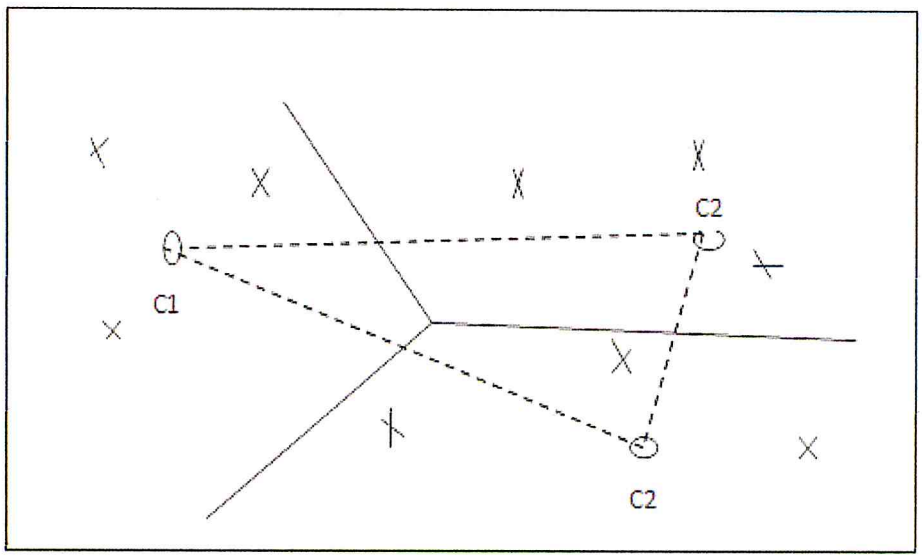


Figure2.3 : exemple d'algorithme de classification à partir de centres. c1 c2 c3 sont les centres de gravité des 3 groupes de point correspondant [Leb et al 82].

5La méthode K-Means :

La méthode k-means(ou des k moyennes) a été proposée par MacQueen **[Mac 67]**. C'est un algorithme simple qui peut être utilisé sur un grand volume de données. Il s'agit de regrouper les objets autour de K centres de gravité en affectant chacun des objets au centre le plus proche. Le nouveau centre de gravité est recalculé à chaque itération et l'opération d'affectation est recommencée jusqu'à ce que les objets ne changent plus de classe, c'est-à-dire qu'il y a stabilité **[Lab 03].**

La méthode k-means est une méthode de partition non hiérarchique ; elle est relativement efficace à cause de sa complexité O(t k n) : n est un ensemble d'objets, k est le nombre de clusters, et t est l'ensemble d'itérations, on a normalement k, t << n.

L'algorithme k-means est très simple, mais il souffre de certaines faiblesses. Le problème le plus pertinent est que l'algorithme a comme paramètre d'entrée le nombre de clusters construits, donc on doit spécifier k (nombre de clusters) à chaque fois, car l'exécution multiple permet de donner des clusters différents relativement à chacune. D'un autre coté k-means n'est pas applicable en présence d'attributs qui ne sont pas du type intervalle, car on a un calcul de moyenne. Une autre faiblesse est que les clusters sont construits par rapport à des objets inexistants (les milieux) ; en plus, l'algorithme k-means ne peut pas découvrir les groupes non convexes. En résultat nous pouvons dire que le choix des premiers clusters peut avoir comme conséquence un taux de convergence faible, ou la convergence de clustering soit optimale. Comme solution le choix des bons clusters peut être amélioré en utilisant les résultats d'une autre méthode.

La méthode k-means est améliorée et redéfinie afin d'être utilisée dans les différents types de clustering.

5.1 Schéma de l'algorithme K-means :

L'algorithme consiste à grouper les points selon un critère bien déterminé. L'entrée de l'algorithme est le nombre *k* de groupes (ou clusters). Une fois le nombre de groupes saisi, l'algorithme choisit arbitrairement *k* points comme centres «initiaux » des *k* groupes.

L'étape suivante consiste à calculer la distance entre chaque individu (point) et les *k* centres ; la plus petite distance est retenue pour inclure cet individu dans le groupe ayant le centre le plus proche (figure 2.4).

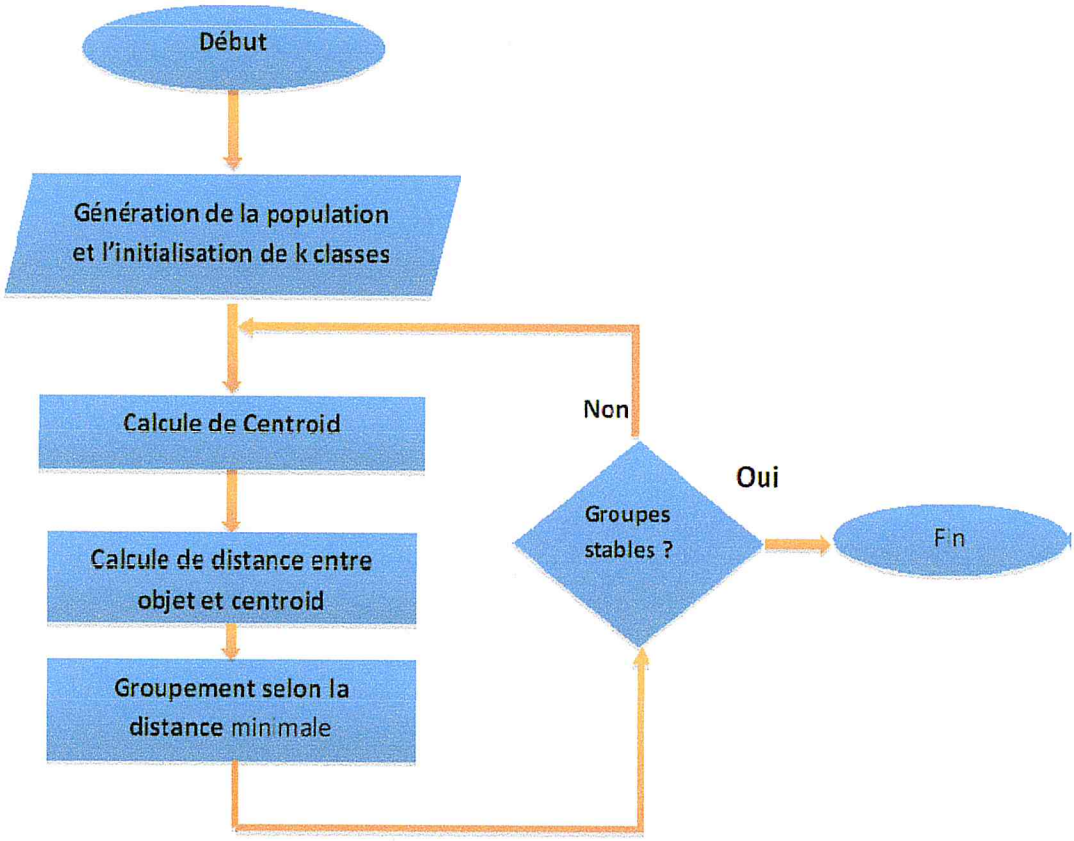


Figure 2.4 : Organigramme de K-means

5.2 L'algorithme K-means [Cle 04]

Entrées : *k* le nombre de clusters désiré, d une mesure de dissimilarité sur l'ensemble des objets à traiter X

Sortie : Une partition C= $\{C_1, \ldots, C_k\}$

Etape 0 :

1. Initialisation par tirage aléatoire dans X, de k centres $x^*_{1,0},\ldots,x^*_{k,0}$

2. Constitution d'une partition initiale $C_0 = \{C_1, \ldots, C_K\}$ par allocation de chaque objet $x_i \in X$ au centre le plus proche :

$$C1= \{x_i \in X \mid d(x_i, x^*_{l,0}) = \min d(x_i,x^*_{h,0})\}\ h=1,\ldots,k$$

3. Calcul des centroïdes des k classes obtenues $x^*_{1,1}, \ldots, x^*_{k,1}$

Etape t :

4. Constitution d'une nouvelle partition Ct = {C1,…, C_k} par allocation de chaque objet $x_i \in X$ au centre le plus proche :

$$Cl= \{x_i \in X \mid d(x_i,x^*_{l,t}) = \min d\ (x_i, x^*_{h,t})\}\ h=1,\ldots k$$

5. Calcul des centroïdes des k classes obtenues $x^*_{1,t+1}, \ldots, x^*_{k,t+1}$

6. Répéter les étapes 4 et 5 tant que des changements s'opèrent d'un schéma C_t à un schéma C_{t+1} ou jusqu'à un nombre d'itérations donné

7. Retourner la partition finale C_{finale}

5.3 Exemple :

Nous allons présenter dans cette partie une simulation de déroulement de l'exécution de l'algorithme de k-means.

Pour cela nous disposons d'un ensemble données (individus) non étiquetées placées sur le plan, qui sont représentées par des losanges.

Pour la première étape trois classes sont été choisis arbitrairement et ils sont représentés par des petits cercles noirs (k_1, k2, k3).

Etape 1 : Choisir 3 centres de classes (au hasard) (Figure 2.5)

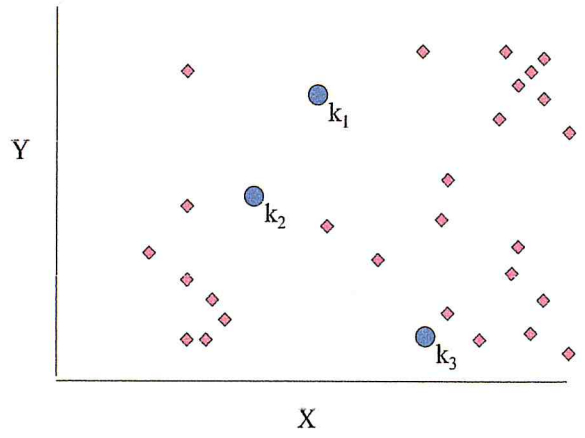


Figure 2.5 : la première étape de l'exemple k-means [Mic 05]

Etape2: La deuxième phase consiste à affecter chaque point au centre le plus proche ; sur la figure 2.6, chaque point porte la couleur du cluster auquel il est affecté.

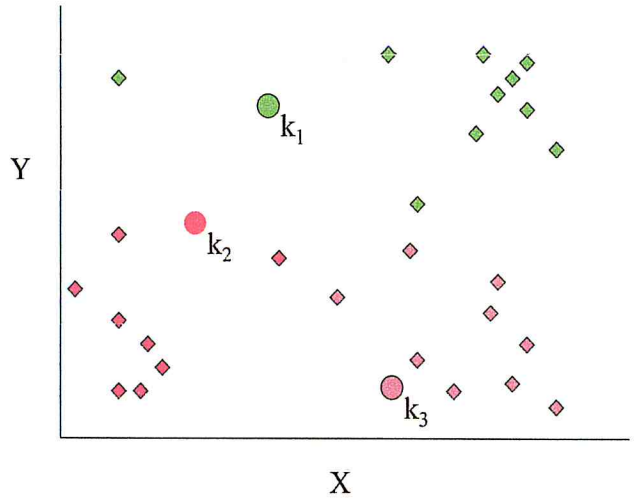


Figure 2.6 : la deuxième étape de l'exemple k-means [Mic 05]

Etape3:la troisième phase consiste à recalculer les nouveaux centres de clusters par la moyenne de chaque classe définie auparavant. La figure 2.7 montre le déplacement des anciens centres vers les nouveaux centres comme indiqué par les flèches.

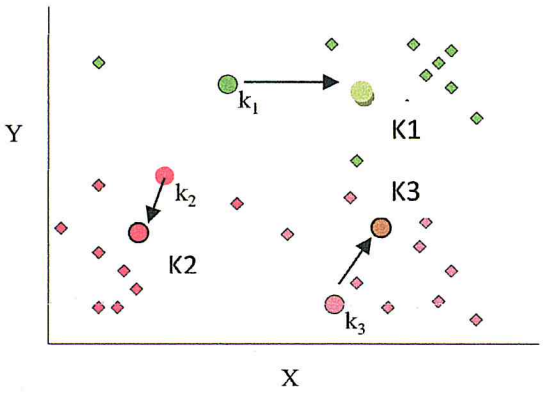


Figure 2.7 : la troisième étape de l'exemple k-means [Mic 05]

Etape4:la quatrième phase concerne la réaffectation des points vers les nouveaux centres. Mais une question pertinente se pose, c'est quels sont les points qui changent de classes.(Figure 2.8).

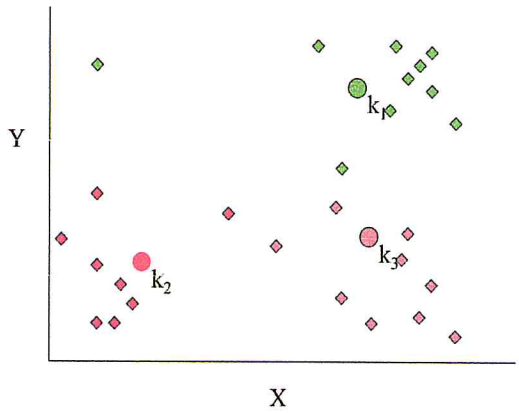


Figure 2.8 : la quatrième étape de l'exemple k-means [Mic 05]

Etape5:l'étape cinq montre quels sont les points qui changent de classe. Ils sont représentés dans le schéma par le changement de leur couleur.(Figure 2.9)

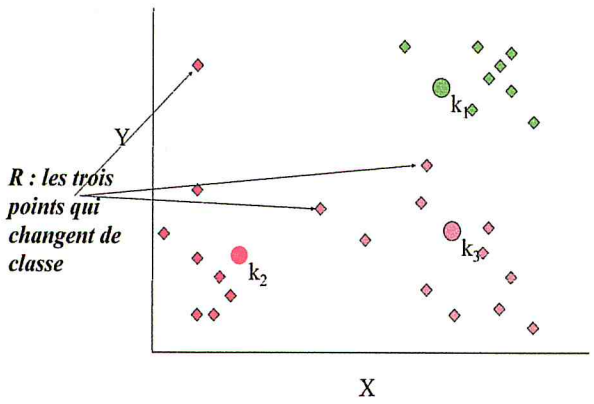


Figure 2.9 : la cinquième étape de l'exemple k-means [Mic 05]

Etape6:dans cette phase on recalcule les moyennes des classes, une première vue du positionnement des individus et centres montre clairement que les centres vont être déplacés vers le sens indiqué par les flèches. (Figure 2.10)

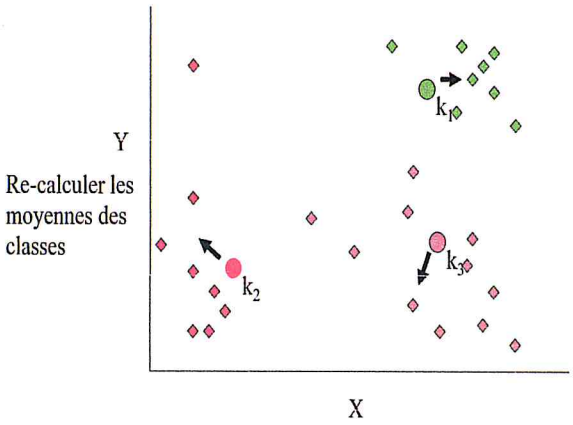


Figure 2.10 : la sixième étape de l'exemple k-means [Mic 05]

Etape7:la phase sept montre le nouveau déplacement des centres de clusters, et l'algorithme s'arrête a ce point, car les objets sont affectés aux bonnes classe, et les centres ne subissent aucun déplacement à partir de leur emplacement actuel (Figure 2.11).

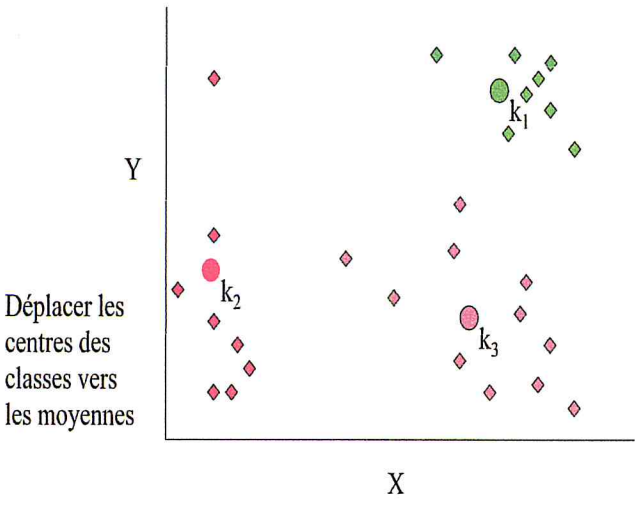


Figure 2.11 : la dernière étape de l'exemple k-means [Mic 05]

5.4Propriétés de l'algorithme :

K-means est un algorithme itératif dont la complexité en temps de calcul est généralement linéaire.

Lors de l'initialisation de l'algorithme, on prend K points dans l'espace de données au hasard. La qualité de la segmentation obtenue dépend du choix des centres initiaux. Pour contrer ce problème, on exécute plusieurs fois l'algorithme en prenant à chaque fois des centres initialisés différemment. On compare les segmentations obtenues à chaque itération et on retient celle dont l'inertie intra-classe est la plus faible. En général, un certain nombre de données se trouvent toujours regroupées ensembles, alors que d'autres ne le sont pas ; on peut considérer que les premières indiquent nettement des regroupements alors que les secondes correspondent à des donnes éventuellement atypiques ou des données bruitées.

5.5 Choix du nombre de centres :

Le critère d'inertie utilisé par K-means est lié au nombre de classes et donc au nombre de centres ; le critère d'inertie vaut zéro si chaque individu de l'ensemble est présent dans une classe singleton. Cela permet de trouver la meilleure partition en optimisant le critère d'inertie. Le choix du nombre de centres est donc primordial pour déterminer la « meilleure » partition finale. Il existe différentes approches pour tenter de résoudre ce problème récurrent :

- K est connu ;
- On impose des contraintes en ce qui concerne les classes en limitant par exemple le nombre d'individus par classe;
- On effectue plusieurs classifications avec des valeurs différentes de k (croissant en général) et on détermine la partition qui minimise le critère d'inertie.

5.6 Avantages et inconvénients de l'algorithme k-means :

L'algorithme K-means possède les avantages suivants **:[Cle 04]**

- Simple
- Compréhensible
- Complexité des calculs en O (k*n)
- Applicable à des données de grande taille

Ses inconvénients sont : **[Cle 04]**

- Le nombre de classes doit être fixé au départ
- Pas de détection des données bruitées
- La qualité de clustering dépend des centroïdes initiaux, l'optimum local calculé est parfois très différent de la valeur globale
- Difficulté pour déterminer automatiquement le nombre de classes dans les données.

5.7Domaines d'application :

L'algorithme K-means est utilisé dans diverses applications :

- En marketing : segmentation du marché en découvrant des groupes de clients distincts à partir de bases de données d'achats.

- Les assurances : identification de groupes d'assurés distincts associés à un nombre important de déclarations.

- La planification de villes : identification de groupes d'habitations suivant le type d'habitation, valeur, localisation géographique, …

- En médecine : localisation de tumeurs dans le cerveau à partir d'un nuage de points du cerveau fourni par le neurologue ; identification des points définissant une tumeur.

- Environnement : identification des zones terrestres similaires (en termes d'utilisation) dans une base de données d'observation de la terre.

6Variantes de l'algorithme k-means :

6.1 Fuzzy C-means :

Les données sont liées à chaque groupe à l'aide d'une fonction d'appartenance, qui représente le comportement de cet algorithme flou. Pour ce faire, nous devons tout simplement de construire une matrice appropriée nommé U dont les facteurs sont des nombres

compris entre 0 et 1, et représente le degré d'appartenance entre les centres de données et des clusters.

Fuzzy c-means (FCM) est un procédé de classification qui permet une partie des données d'appartenir à deux ou plusieurs pôles. Cette méthode (développé par Dunn en 1973 et amélioré par Bezdek en 1981) Il est basé sur la minimisation de la fonction objective suivante:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 \qquad 1 \leq m < \infty$$

où m est un nombre réel supérieur à 1, u_{ij} est le degré d'appartenance de x_i dans le groupe j ,x_i est la $i$ème dimension d des données mesurées, c_j est le centre dimension d de la grappe, et $||*||$ est une norme exprimant la similitude entre les données mesurées et le centre.

Partitionnement logique floue est réalisée par une méthode itérative d'optimisation de la fonction objectif indiqué ci-dessus, avec la mise à jour de l'adhésion u_{ij} et les centres de cluster c_j par :

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

Cette itération s'arrête lorsque $\max_{ij} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \right\} < \varepsilon$, où ε un critère de rupture est comprise entre 0 et 1, tandis que k sont les étapes d'itération. Cette procédure converge vers un minimum local ou d'un point de selle J_m.

L'algorithme se compose des étapes suivantes:

Etape1 :

- Initialiser U = [u_{ij}] matrice, $U^{(0)}$

Etape 2 :

- À k-étape: calculer les centres de vecteurs $C^{(k)} = [c_j]$ avec $U^{(k)}$
-

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

Etape3 :

- Mise à jour de $U^{(k)}$, $U^{(k+1)}$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

Si $||U^{(k+1)} - U^{(k)}|| < \varepsilon$ puis sur STOP, sinon retourner à l'étape 2.

6.2 La méthode K-harmonic-means

L'algorithme K-Harmonic means (KHM) **[Zha 00]** est similaire à K-means uniquement dans le sens ou cette approche est une méthode itérative basée sur les centres.

Le but de cette approche est de pallier le problème de l'initialisation des centres, que l'on rencontre pour les méthodes k-means.
Cet algorithme diffère notamment de k-means par le critère d'optimisation.

En effet, ce critère est fondé sur la moyenne harmonique de la distance de chaque document avec touts les centres **[Cha 04] :**

$$KHM(x,c) = \sum_{i=1}^{n} \frac{k}{\sum_{j=1}^{k} \frac{1}{\|xi-cj\|^p}}$$

Cette fonction d'objectivité donne un bon score (c'est-à-dire un poids faible) pour tout élément ce trouve proche d'au moins un centre. Cette fonction se comporte comme la fonction min de k-means. Ceci est la propriété voulue de cette fonction pour mesurer la qualité des classes retrouvées. P est un paramètre de l'algorithme **[Cha 04].**

6.2.1 Algorithme k-harmonic-Means

Entrée : entrée dans xi de n objet, nombre de cluster K et le paramètre t(t>=2).
Sortie : K partition.

Etape 1 :
Déclarer une matrice U de taille n×k

Etape 2 :

Générer aléatoirement k objet et associe chaque cluster un objet(Mettre centroide initiale)

En calcule la fonction objective :

$$KHM(x,c) = \sum_{i=1}^{n} \frac{k}{\sum_{j=1}^{k} \frac{1}{\|xi-cj\|^p}}$$

Etape 3 :
Remplir la matrice U ;

$$Ui,j = \frac{\|xi - cj\|^{-t-2}}{\sum_{j=1}^{k} \|xi, cj\|^{-t-2}}$$

Etape 4 :

Calculer les nouveaux centres

$$Ck = \frac{\sum_{i=1}^{n} \frac{1}{b_{i,k}^{p+2} \left(\sum_{i=1}^{k} \frac{1}{b_{i,j}^{n}}\right)^{2}} xi}{\sum_{i=1}^{n} \frac{1}{b_{i,k}^{p+2} \left(\sum_{i=1}^{k} \frac{1}{b_{i,j}^{n}}\right)^{2}}}$$

Et $b_{i,k}$ la distance entre objet i et le centre de gravité k

Etape 5 : répéter l'étape 3 et 4 jusqu'à la valeur de KHM est stable ;

Etape 6 : associe chaque objet i avec classe k par rapport la plus grand valeur de $I_{i,j}$.

6.3 La méthode Bisecting k-means

La méthode Bisecting k-means est une méthode de partition (k-means) où on utilise une hiérarchie (ACH). C'est une méthode hiérarchique divisifs.

6.3 .1 Le principe :

1. Choisir un cluster à diviser : il y'a plusieurs manières de sélectionner le cluster à décomposer, mais l'expérience prouve qu'il n'y a aucune différence significative. Habituellement, on choisit le plus grand cluster.

2. Trouver 2sous clusters en utilisant 2 means : utiliser l'algorithme k-mens pour subdiviser le cluster choisi ;

3. Répéter l'étape 1 pour un nombre constante de fois.

4. Répéter les trois étapes ci-dessous jusqu'à ce que le nombre désiré de cluster soit atteint.

La méthode bisecting k-Means est une méthode dérivée de k-means. Elle maintient les mêmes étapes dans la construction des clusters mais elle tend à produire des clusters de taille uniforme contrairement a k-means, dont la taille des clusters est largement différente ; on sait que la taille d'un cluster est petite, ces clusters sont de haute qualité, mais cela ne contribue pas beaucoup à la mesure globale de qualité du « clustering ».

6.4 Spherical K-means:

La méthode Spherical K-means est un algorithme simple qui peut être utilisé sur un grand volume de données. Il s'agit de regrouper les objets autour de K centres de gravité en affectant chacun des objets au centre le plus proche. Le nouveau centre de gravité est recalculé à chaque itération et l'opération d'affectation est recommencée jusqu'à ce que les objets ne changent plus de classe, c'est-à-dire qu'il y a stabilité.

L'algorithme Spherical K-means est similaire à K-means et diffère de ce dernier par le fait que la similarité cosinus est utilisée au lieu de la distance euclidienne.

L'algorithme consiste à grouper les points selon un critère bien déterminé. L'entrée de l'algorithme est le nombre k de groupes (ou clusters). Une fois le nombre de groupes saisi, l'algorithme choisit arbitrairement k points comme centres «initiaux » des k groupes.

L'étape suivante consiste à calculer la similarité du cosinus entre chaque individu (point) et les k centres ; la plus grande valeur de similarité est retenue pour inclure cet individu dans le groupe ayant le centre le plus proche.

6.4.1 Algorithme de Sphérical K-means:

Entrées : k le nombre de clusters désiré, *cos* une mesure de similarité sur l'ensemble des objets à traiter X

Sortie : Une partition $C= \{C_1, \ldots, C_k\}$

Etape 0 :

1. Initialisation par tirage aléatoire dans X, de k centres $x^*_{1,0},\ldots,x^*_{k,0}$

2. Constitution d'une partition initiale $C_0 = \{C_1, \ldots, C_K\}$ par allocation de chaque objet $x_i \in X$ au centre le plus proche :

$$C1=\{x_i \in X \mid \cos(x_i, x^*_{l,0}) = \max \cos(x_i, x^*_{h,0})\} \; h=1,\ldots,k$$

3. Calcul des centroïdes des k classes obtenues $x^*_{1,1}, \ldots, x^*_{k,1}$

Etape t :

4. Constitution d'une nouvelle partition $Ct = \{C1,\ldots, C_k\}$ par allocation de chaque objet $x_i \in X$ au centre le plus proche :

$$Cl= \{x_i \in X \mid \cos (x_i , x^*_{l,t}) = \max \cos (x_i, x^*_{h,t})\} \; h=1,\ldots k$$

5. Calcul des centroïdes des k classes obtenues $x^*_{1,t+1}, \ldots, x^*_{k,t+1}$

6. Répéter les étapes 4 et 5 tant que des changements s'opèrent d'un schéma C_t à un schéma C_{t+1} ou jusqu'à un nombre d'itérations donné

7. Retourner la partition finale C_{finale}.

7 Conclusion :

Comme le clustering est le regroupement des données similaires, plusieurs méthodes et approches sont utilisées afin d'améliorer la qualité des groupes (clusters), car une bonne méthode de regroupement doit garantir le mieux.

Les méthodes de clustering par partitionnement dont lesquelles les algorithmes utilisés sont basés sur la construction de plusieurs partitions, puis leurs évaluations selon certains critères. Ces méthodes sont largement utilisées a cause de leur efficacité, car elles sont préférables si l'efficacité est importante, ou si on a un bon nombre de données à traiter.

La méthode la plus connue dans cette catégorie est « k-means », c'est une méthode qui sera utilisée à cause de sa souplesse et efficacité, mais elle souffre de certaines faiblesses. Pour cela, elle est améliorée et modifier pour augmenter la qualité de clustering. Par conséquence, ils apparaissent plusieurs variantes de la méthode k-mens.

Chapitre 3 : Evaluation et comparaison

1. Introduction :

Dans ce chapitre, nous proposons une étude expérimentale dans laquelle nous évaluons et comparons les différentes versions de l'algorithme K-means, Nous utilisons pour cela des jeux de données réels et deux critères d'évaluation.
Nous avons implémenté K-means et ses variantes sous Matlab car c'est un langage adapté à la manipulation de données représentées sous forme de matrices comme c'est le cas des données textuelles.

2. MATLAB :

MATLAB est un langage haute performance pour le calcul scientifique. Il intègre la possibilité de calcul, de visualisation et de programmation dans un environnement très simple d'emploi. Les résultats sont exprimés sous une forme mathématique standard. Matlab est utilisé dans un grand nombre d'applications tel que :

-calcul scientifique

-développement d'algorithmes

-Acquisition de données

-Modélisation et simulation

-Analyse de données, exploration et visualisation

-Graphisme scientifique

-Développement d'applications, interface graphique (gui).

MATLAB est un système interactif dont la brique de base est un tableau dont la taille n'est pas nécessairement connue. Ceci permet de résoudre des problèmes, en particulier ceux qui ont une formulation matricielle, en un minimum de temps (contrairement aux langages de bas niveau comme le C ou le fortran). Le nom de MATLAB est un résumé de "Matrix Laboratory". MATLAB a été à l'origine développé pour avoir un accès simple et rapide aux projets EISPACK et LINPACK. Aujourd'hui, MATLAB intègre les librairies LAPACK et BLAS, incorporant ainsi les dernières techniques pour le calcul matriciel.

Dans l'enseignement universitaire, MATLAB s'est imposé comme un standard pour l'apprentissage de l'algorithmique scientifique. Dans l'industrie, il est l'outil de choix pour une productivité accrue en recherche et développement.
MATLAB peut aussi être enrichi à l'aide de Toolbox (boites à outils) pour des problèmes spécifiques.

3. Les données utilisées :

Citeseer et Cora sont des corpus d'articles scientifiques dans le domaine de l'informatique **[URL 1].** Chaque document de Cora appartient à l'une des thématiques suivantes :réseaux de neurones, algorithmes génétiques, apprentissage par renforcement, théorie de l'apprentissage, apprentissage de règles, méthodes probabilistes, et raisonnement par cas. Les documents de Citeseer appartiennent à cinq thématiques : agents, bases de données, recherche d'information, apprentissage, et interaction homme-machine.
Le tableau suivant résumé les caractéristiques des deux jeux de données utilisés.

Les données	Nombre de documents	Nombre de termes
cora	**2708**	**3996**
citeseer	**2994**	**1928**

Tableau3.1- Propriétés des corpus utilisés pour les expérimentations

4. Algorithmes comparés :

Lors de nos expérimentations, nous avons comparé les six algorithmes suivants :

K-means (version classique avec distance euclidienne), Spherical k-means, harmonic k-means, bisecting k-means, spherical-bisecting (c'est un algorithme bisecting avec la similarité de cosunis), et harmonic-spherical (c'est un algorithme sphérique utiluse la moyenne harmonique pour calculer les centres ou lieu de la moyenne arithmitique).

5. Pondération des termes :

5.1Pondération locale :

L'indexation automatique a montré que l'occurrence simple d'un mot ne peut pas indiquer le thème, la signification ou le but d'un texte. Ainsi des programmes ont commencé à compter

les mots dans les textes, espérant que la fréquence de terme indiquerait mieux ce qui est important dans un message. La pondération locale permet de mesurer la représentativité locale d'un terme. Elle prend en compte les informations locales du terme qui ne dépendent que du document donné, et indique l'importance du terme dans ce document. Voici la formule pour le facteur TF :

$$TF_{ij}=n_{ij}/n_j$$

Où n_{ij} est le nombre d'occurrence du terme i dans le document j et n_j est le nombre total de termes dans le document j.

5.2Pondération globale :

Contrairement à la pondération locale, la pondération globale s'intéresse aux informations concernant les termes et dépend de la collection de documents.

Elle indique la représentativité globale du terme dans l'ensemble des documents de la collection. Ainsi, une telle pondération prenant en compte l'importance d'un terme dans toute la collection améliore les performances dans le cadre de la recherche d'information **[Kar 07]**. Un terme qui apparait souvent dans la base documentaire ne doit pas avoir le même impact qu'un terme moins fréquent. Un poids plus important doit être donné aux termes qui apparaissent le moins fréquemment dans la collection, car ces termes ont un pouvoir discriminatoire plus important que ceux apparaissant dans de nombreux documents de la collection. Par conséquent, un facteur désigné par Idf peut être exprimé par la formule suivante :

$$Idf = \log (N/n_i)$$

où n_i est le nombre de documents contenant le terme i et N est le nombre total de documents dans la collection.

5.3Pondération locale et globale :

TF-IDF est une mesure statistique permettant d'évaluer l'importance d'un mot par rapport à un document extrait d'une collection. Le poids augmente proportionnellement en fonction du nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans la collection **[Kar 07]**.

Du fait de cette double pondération (locale et globale), les fonctions de pondération sont souvent référencées sous le nom de Tf-Idf. Cette technique de pondération a été utilisée au début par le modèle vectoriel classique. Une formule Tf-Idf combine donc les deux critères vus précédemment :

- L'importance du terme pour un document (par Tf),
- Le pouvoir de discrimination de ce terme (par Idf).

Ainsi, une valeur Tf-Idf élevée pour un terme signifie que ce terme est important dans le document et en plus,qu'il apparait peu dans les autres documents de la collection. Le terme correspond donc à une caractéristique importante et unique d'un document. Voici la formule obtenue par la combinaison de Tf-Idf :

$$\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \cdot \mathrm{idf}_i$$

6. Mesures d'évaluation :

Il existe dans la littérature plusieurs critères d'évaluation de clustering. Ces mesures d'évaluation sont de deux types : les mesures internes et les mesures externes.**[Chi 10]**
Les mesures internes sont propres au modèle, nous pouvons citer la distance euclidienne ou la vraisemblance des données. Les mesures dites externes font appel à des informations supplémentaires concernant la catégorie de chaque objet à classer. Là encore, une panoplie de mesures existe, nous citerons par exemple : la F-mesure, l'information mutuelle normalisée (NMI, Normalized Mutual Information), l'indice de Rand, la variation d'information, etc. Les deux premières sont les plus utilisées c'est pour nous les avons retenues pour nos expérimentations.
La NMI est une mesure issue de la théorie de l'information permettant de comparer deux partitions (ou clusterings). Avec cette mesure, on considère chaque partition comme une distribution de probabilité et on calcule la quantité d'information commune aux deux distributions. Strehl **[Str 02]** propose de normaliser ce critère afin que sa valeur soit comprise entre 0 et 1. Une valeur 1 indique que les deux partitions sont identiques.
Voici la formule pour la mesure de NMI **[Str 02]:**

$$\text{NMI (A,B)}=\frac{2}{n}\sum_{l=1}^{k(a)}\sum_{h=1}^{k(h)} n * log_{k(a)*k(b)}\ \left(\frac{a_l^{(H)} n}{a^{(H)} n_l}\right)$$

La F-mesure est une mesure très connue dans le domaine de la recherche d'information.
Elle est égale à la moyenne harmonique entre la précision et le rappel. Elle est toujours comprise entre 0 et 1. Le tableau ci-dessous montre comment calculer la F-mesure.

Nom	Formule	Description
Précision	$P = \frac{vp}{vp + fp}$	Proportion desolutions trouvées qui sont pertinentes. Mesure la capacité du système à refuser les solutions non-pertinentes
Rappel	$R = \frac{vp}{vp + fn}$	Proportion des solutions pertinentes qui sont trouvées. Mesure la capacité du système à donner toutes les solutions pertinentes.
F-mesure	$F = \frac{2PR}{P + R}$	Moyenne harmonique de la précision et du rappel. Mesure la capacité du système à donner toutes les solutions pertinentes et à refuser les autres

Tableau3.2. Les différentes étapes pour calculer la F-mesure

7. Evaluation de l'effet de la pondération:

Nous avons appliqués les algorithmes comparés en utilisant diverses méthodes de pondération de la matrice termes-documents. Pour l'arrêt des algorithmes, plusieurs critères d'arrêt peuvent être considérés : stabilité du vecteur d'appartenance, stabilité de la fonction objectif, etc. Nous avons choisi ici une méthode simple qui consiste à fixer le nombre d'itérations maximal de l'algorithme Nous avons fixé ce nombre à 100.

Pour le calcul des mesures d'évaluation, nous exécutons chaque algorithme 10 fois puis la moyenne et la déviation standard sont calculées pour chaque algorithme.

7.1. Résultats avec la collection Cora :

7.1.1Evaluation des algorithmes sans utiliser de pondération:

D'après le tableau ci-dessous, nous remarquons que l'algorithme Spherical k-means est le meilleur car la valeur de la NMI est la plus grand par rapport aux autres algorithmes ; il est suivi de près par la version qui procède par division qui des donne des résultats proches. L'algorithme harmonic-Spherical qui n'identifie aucun clustering (NMI égale à 0).

Les Algorithme	La F-mesure	La NMI
k-means	**0.3759 ± 0.0353**	**0.0779 ± 0.0371**
Spherical-k-means	**0.4411 ± 0.0142**	**0.2553 ± 0.0316**
Harmonic--k-means	**0.4329 ± 0.0237**	**0.0537 ± 0.0230**
Bisecting-k-means	**0.4298 ± 0.0387**	**0.0572 ± 0.0273**
Harmonic-spherical-kmeans	**0.4640**	**0**
Algorithme spherical-Bisecting-k-means	**0.4463 ± 0.0160**	**0.2298 ± 0.0213**

Tableau3.3-Evaluation de l'algorithme K-means et ses variantes sans pondération en utilisant la collection Cora

7.1.2Evaluation des algorithmes en utilisant la pondération TF :

Les Algorithme	La F-mesure	La NMI
k-means	**0.3941 ± 0.0107**	**0.0877 ± 0.0062**
Spherical-k-means	**0.4224 ± 0.0038**	**0.2651 ± 0.0226**
Harmonic--k-means	**0.4640**	**0**
Bisecting-k-means	**0.4523 ± 0.0004**	**0.0374 ± 0.0051**
Harmonic-spherical-kmeans	**0.4640**	**0**
Algorithme spherical-Bisecting-k-means	**0.4297 ± 0.0294**	**0.2134 ± 0.0047**

Tableau3.4-Evaluation de l'algorithme K-means et ses variantes avec TF en utilisant la collection Cora.

Le tableau ci-dessus montre que l'algorithme Spherical k-means est le meilleur, suivi encore une fois par la version qui procède par bissection. Les versions utilisant la moyenne harmonique pour le calcul des nouveaux centroïdes donnent de très mauvais résultats quelle que soit la mesure de distance employée.

Nous remarquons aussi que la pondération TF a amélioré de manière infime les résultats de l'algorithme Sphérical-k-means puisque la valeur de la NMI est passée de 0.2553 sans pondération à 0.2651 avec la pondération TF.

7.1.3Evaluation des algorithmes en utilisant la pondération IDF :

Les résultats du tableau ci-dessous montrent que Spherical k-means et Sphérical-bisecting-k-means sont les meilleurs car les valeurs de NMI sont largement supérieures par rapport aux valeurs obtenues avec les autres algorithmes. Même avec cette pondération, les versions basées sur la moyenne harmonique semblent être inadaptées pour le clustering de textes.

Nous remarquons que la pondération IDF permet d'améliorer significativement les résultats de clustering par rapport à la pondération TF. Pour Sphérical-k-means, la valeur de la NMI est passée de 0.26 avec la pondération TF à 0.36 avec la pondération IDF.

Les Algorithme	La F-mesure	La NMI
k-means	**0.3947 ± 0.0649**	**0.0988 ± 0.0681**
Spherical-k-means	**0.5481 ± 0.0671**	**0.3630 ± 0.0544**
Harmonic--k-means	**0.4640**	**0**
Bisecting-k-means	**0.4790 ± 0.0220**	**0.0840 ± 0.0695**
Harmonic-spherical-kmeans	**0.4640**	**0**
Algorithme spherical-Bisecting-k-means	**0.5318 ±0.0518**	**0.3406 ± 0.0455**

Tableau3.5-Evaluation de l'algorithme K-means et ses variantes avec IDF en utilisant la collection Cora.

7.1.4Evaluation des algorithmes en utilisant la pondération TF-IDF :

Les Algorithme	La F-mesure	La NMI
k-means	**0.4347 ± 0.0630**	**0.2288 ± 0.0672**
Spherical-k-means	**0.5500 ± 0.0207**	**0.3513 ± 0.0234**
Harmonic--k-means	**0.4640**	**0**
Bisecting-k-means	**0.5319 ± 0.0245**	**0.2537 ± 0.0239**
Harmonic-spherical-kmeans	**0.4640**	**0**
Algorithme spherical-Bisecting-k-means	**0.5617 ± 0.0420**	**0.3721 ± 0.0261**

Tableau3.6-Evaluation de l'algorithme K-means et ses variantes avec TF-IDF en utilisant la collection Cora

D'après le tableau ci-dessus, l'algorithme Sphérical-bisecting-k-means est le meilleur car la valeur de la NMI estla plus grande par rapport aux autres algorithmes, et l'algorithme harmonic-spherical et harmonic-k-means sont mauvais car les valeurs de la NMI sont égales à 0. Nous remarquons que la pondération TF-IDF améliore les résultats de tous les algorithmes y compris ceux de l'algorithme K-means standard

Ce tableau montre aussi Spherical-bisecting-k-means combiné à la pondération TF-IDF permet d'obtenir les meilleurs résultats avec une NMI égale à 0.37 et une F-mesure égale à 0.56.

7.2 Résultats avec la collection Citeseer :

7.2.1 Evaluation des algorithmes sans utiliser de pondération :

Les Algorithme	La F-mesure	La NMI
k-means	**0.4001 ± 0.0388**	**0.1769 ± 0.0514**
Spherical-k-means	**0.5739 ± 0.0415**	**0.2674 ± 0.0286**
Harmonic--k-means	**0.3705 ± 0.0045**	**0.0412 ± 0.0383**
Bisecting-k-means	**0.3979 ± 0.0482**	**0.0840 ± 0.0960**
Harmonic-spherical-kmeans	**0.3750**	**0**
Algorithme spherical-Bisecting-k-means	**0.5932 ± 0.0295**	**0.2831 ± 0.0227**

Tableau3.7-Evaluation de l'algorithme K-means et ses variantes sans pondération en utilisant la collection Citeseer

D'après le tableau ci-dessus, on peut dire que l'algorithme Spherical-bisecting-k-means est le meilleur car la valeur de F-mesure est plus grande par rapport aux autres algorithmes ; le plus faible algorithme est harmonic-Sphérical-k-means.

7.2.2Evaluation des algorithmes en utilisant la pondération TF :

Les Algorithme	La F-mesure	La NMI
k-means	**0.4477 ±0.0214**	**0.2281 ± 0.0179**
Spherical-k-means	**0.5761 ± 0.0497**	**0.2701 ± 0.0327**
Harmonic--k-means	**0.3853 ± 0.0468**	**0.0541 ± 0.0795**
Bisecting-k-means	**0.4309 ± 0.0379**	**0.1400 ± 0.0638**
Harmonic-spherical-kmeans	**0.3750**	**0**
Algorithme spherical-Bisecting-k-means	**0.6141 ± 0.0233**	**0.2916 ± 0.0119**

Tableau3.8-Evaluation de l'algorithme K-means et ses variantes avec TFen utilisant la collection Citeseer.

D'après ce tableau, on peut dire que l'algorithme Spherical-bisecting-k-means est le meilleur. On remarque car la pondération TF n'a pas beaucoup amélioré les résultats de Sphérical K-means et de Sphérical Bisecting K-means mais qu'elle a sensiblement amélioré les résultats de K-means et bisecting k-means.

7.2.3Evaluation des algorithmes en utilisant la pondération IDF :

Les Algorithme	La F-mesure	La NMI
k-means	**0.4667 ± 0.0243**	**0.2140 ± 0.0291**
Spherical-k-means	**0.6001 ±0.0606**	**0.3198 ± 0.0479**
Harmonic--k-means	**0.3739 ± 0.0016**	**0.0088 ± 0.0134**
Bisecting-k-means	**0.4204 ± 0.0522**	**0.1056 ± 0.0932**
Harmonic-spherical-kmeans	**0.3750**	**0**
Algorithme spherical-Bisecting-k-means	**0.6392 ± 0.0423**	**0.3415 ± 0.0241**

Tableau3.9-Evaluation de l'algorithme K-means et ses variantes avec IDF en utilisant la collection Citeseer

Le tableau montre que la pondération IDF a un effet très positif sur les algorithmes Sphérical-k-means et Sphérical-bisecting-k-means que la pondération TF. On observe aussi que la pondération IDF améliore légèrement les algorithmes K-means et bisecting K-means mais avec un degré moindre que TF.

7.2.4Evaluation des algorithmes en utilisant la pondération TF-IDF :

Les Algorithme	La F-mesure	La NMI
k-means	**0.5797 ± 0.0671**	**0.3170 ± 0.0598**
Spherical-k-means	**0.6390 ± 0.0564**	**0.3593 ± 0.0327**
Harmonic--k-means	**0.3750**	**0**
Bisecting-k-means	**0.5155 ± 0.0280**	**0.2437 ± 0.0401**
Harmonic-spherical-kmeans	**0.3750**	**0**
Algorithme spherical-Bisecting-k-means	**0.6442 ± 0.0299**	**0.3598 ± 0.0175**

Tableau3.10-Evaluation de l'algorithme K-means et ses variantes avec TF-en utilisant la collection Citeseer

Les résultats du tableau ci-dessous montrent clairement que la pondération TF-IDF améliore sensiblement les performances des algorithmes Sphérical K-means, bisecting Sphérical K-means et même celles de l'algorithme K-means qui passe de 0.17 pour la NMI sans pondération à 0.31 avec la pondération TF-IDF. L'algorithme harmonic K-means donne encore une fois de très mauvais résultats même avec la pondération TF-IDF.

8. L'indexation sémantique latente « LSI »

Nous étudions dans cette section l'effet de la technique LSI sur les différentes variantes de l'algorithme K-means.

8.1 Définition

L'indexation sémantique latente *(Latent Semantic Indexing LSI)* **[Dum 93] [Dum 94]** est une technique d'indexation s'inspirant de la *LSA (Latent Semantic Analysis)* développée au début des années 90 **[Dee et al 90].** Ce modèle fut conçu comme une nouvelle approche de l'indexation et de la récupération automatique d'informations dans des bases de données constituées de documents textuels. Son but est d'améliorer les techniques d'indexation textuelles et ainsi de permettre le rapprochement sémantique d'un certain nombre de documents au travers des mots les composant. En effet, la LSA permet de résoudre les problèmes de synonymie et de polysémie.

La LSI est une technique d'analyse statistique qui vise à décrire de manière économique les cooccurrences de termes qui surviennent au sein d'un ensemble de documents, puis d'en déduire des proximités sémantiques entre termes. En effet, deux mots peuvent être considérés sémantiquement proches s'ils sont utilisés dans des contextes similaires. Le contexte d'un mot est ici défini comme l'ensemble des mots qui apparaissent conjointement avec lui. Ainsi, les mots « écolier » et « lycéen » sont considérés sémantiquement proches car ils apparaissent tous deux avec des mots comme « classe », « cours », « examen », etc.

En effet, grâce à cette analyse statistique, le sens de chaque mot est caractérisé par un vecteur dans un espace de grande dimension. L'angle formé par deux vecteurs correspondra àla proximité sémantique de ces mots, sans toutefois que celle-ci soit déterminée.

Comme première étape, cette analyse consiste à construire une matrice d'occurrences Documents x Termes (Fig.3.1). Ainsi, pour la représentation d'un ensemble de N documents par M termes, nous aurons une matrice A de tailleN xM, dont chaque ligne est associée à un document et contient le nombre d'occurrence des M termes dans le document.

A=	t1	t2	t3	.....tm
D1				
D2				
D3				
|				
Dn				

Figure3.1 -Matrice Documents-Termes

Une fois la matrice remplie, l'étape suivante consiste à réduire ses dimensions par le biais d'une décomposition en valeurs singulières *(Singular Value Décomposition SVD)*. Celle-cipermet de décomposer la matrice *A* en un produit de trois autres matrices :

$$A = USV^T (1)$$

où *U* est une matrice orthogonale de taille M x N (UU^T=IM) de description des termes,
V est une matrice orthogonale de taille N x N (VV^T=IN) de description des documents et *S* est une matrice diagonale de taille N x N.

Les colonnes de *U* et *V* sont connues sous le nom de vecteurs singuliers respectivement droits et gauches de *A*. Les éléments diagonaux de S sont appelés valeurs singulières de *A* : ce sont les racines carrées non nulles des N valeurs propres de AA^T.
A partir d'un certain nombre *k*, on s'aperçoit de l'existence de valeurs singulières très faibles et qui peuvent être négligées dans la matrice. De ce fait, il est possible de prendre l'approximation A_k suivante :

$$\mathbf{A_k = U_k S_k V^T_k}$$

Et pour l'obtenir il suffit de :

- Ranger les éléments de *S* par ordre décroissant et réarranger les colonnes correspondantes de *U* et *V* de façon à garder l'égalité (1).
- Garder les k premières valeurs de *S* et les colonnes et lignes correspondantes de

V^T et U respectivement.

Note : Il n'y a pas de règle établie pour la détermination de k, le critère de sélection est expérimental et dépend du corpus de documents utilisé. Instinctivement, nous pouvons supposer que plus k est faible, plus nous éliminons du bruit dans les données, mais que parallèlement nous perdons de l'information : *il faut prendre grand soin lors du choix de ce nombre*.

Cette réduction va nous permettre de ne garder que les k termes les plus significatifs, et afin de calculer la similarité documents-documents, les documents sont représentés dans un espace vectoriel de dimensions k(Fig.3.2). Les colonnes du produit de matrices $S_k x V^T_k$ sont les coordonnées des vecteurs documents

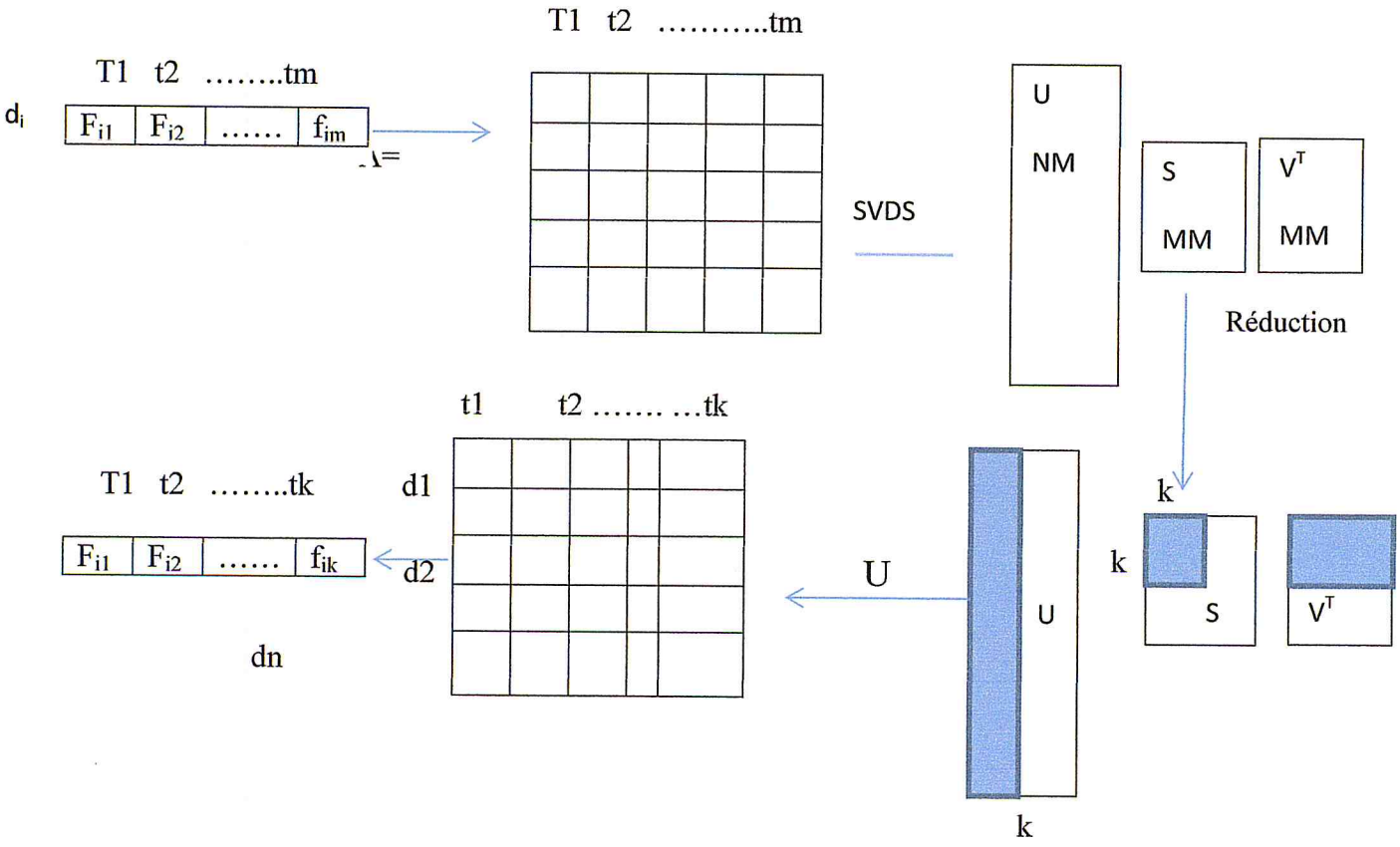


Figure3.2 : Démarche suivie pour l'indexation des documents avec la LSI.

8.2Exemple de la LSI :

Afin d'illustrer le principe de la LSI, considérons les trois "documents" suivants :

Doc1 : "Bouteflika débute la campagne présidentielle FLN"
Doc2 : "Bouteflika affronte Benflis"
Doc3 : "Il y a des tracteurs en campagne"

Ils sont constitués de l'ensemble des mots suivants {Bouteflika, débute, campagne, Présidentielle, FLN, affronte, Benflis, tracteurs} après retrait de l'ensemble des mots vides {la, a, été, pour, il, y, des, en}.

On obtient ainsi la matrice Documents-Termes suivante :

	Bouteflika	débute	campagne	présidentielle	FLN	affronte	Benflis	tracteurs
DOC1	1	1	1	1	1	0	0	0
DOC2	1	0	0	0	0	1	1	0
DOC3	0	0	1	0	0	0	0	1

Donc X=[1 1 1 1 1 0 0 0 ; 1 0 0 0 0 1 1 0 ; 0 0 1 0 0 0 0 1]

En appliquant la SVD sur cette matrice, on obtient :

[U S V]=SVDS(X) :

U =

-0.9064 -0.2482

-0.3419 0.9064

-0.2482 -0.3419

S =

2.3772 0

0 1.6511

V =

-0.5251 0.3986

-0.3813 -0.1504

-0.4857 -0.3574

-0.3813 -0.1504

-0.3813 -0.1504

-0.1438 0.5489

-0.1438 0.5489

-0.1044 -0.2071

8.3 LSI et matrices creuses :

On appelle matrice creuse (en anglais "sparse Matrix") une matrice comportant une forte proportion de coefficients nuls. De nombreux problèmes issus de la physique conduisent à l'analyse de systèmes linéaires à matrice creuse. L'intérêt de telles matrices résulte non seulement de la réduction de la place mémoire (on ne stocke pas les zéros) mais aussi de la réduction du nombre d'opérations (on n'effectuera pas les opérations portant sur les zéros).

Par défaut dans MATLAB une matrice est considérée comme *pleine* (ou *full* en anglais), c'est-à-dire que tous ses coefficients sont mémorisés. Si *M* est une matrice, la commande ***sparse(M)*** permet d'obtenir la même matrice mais stockée sous la forme sparse. Si l'on a une matrice stockée sous la forme sparse, on peut obtenir la même matrice stockée sous la forme ordinaire par la commande ***full***.

MATLAB dispose également de la commande *svds* (SVD Sparse) l'équivalent de la commande *svd* mais pour les matrices creuses ; elle est beaucoup plus rapide que la commande standard. C'est cette commande que l'on utilisera par la suite lors de nos expérimentations car les matrices documents-termes sont connues pour être très creuses.

8.4 Evaluation de l'effet de la LSI :

8.4.1 Evaluation en utilisant la collection Cora :

Le tableau ci-dessous donne les résultats obtenus en appliquant l'algorithme Sphérical K-means, en utilisant la LSI et sans pondération. On remarque que la LSI permet d'améliorer légèrement les performances de cet algorithme et que plus le degré de la technique LSI est grand, plus les résultats de F-mesure et la NMI sont meilleurs.

Algorithme Spherical-k-means		
F-mesure	NMI	Dim. de LSI (k)
0.4101± 0.0050	**0.1917** ± 0.0073	**5**
0.4298 ± 0.0029	**0.2313** ± 0.0008	**10**
0.4343 ± 0.0114	**0.2375** ± 0.0194	**20**
0.4303 ± 0.0052	**0.2430**± 0.0246	**50**
0.4556 ± 0.0589	**0.2608** ± 0.0534	**100**
0.4623 ± 0.0097	**0.2713** ±0.0063	**200**

Tableau3.11-Evaluation de l'algorithme Spherical-k-means, avec LSI et sans pondération en utilisant la collection Cora

Algorithme Spherical-k-means		
F-mesure	NMI	Dim. de LSI (k)
0.5470 ± 0.0118	**0.3718** ± 0.0032	**5**
0.5771 ± 0.0383	**0.3779** ± 0.0242	**10**
0.6028 ± 0.0248	**0.4138** ± 0.0150	**20**
0.5949 ± 0.0538	**0.3975** ± 0.0395	**50**
0.5825 ± 0.0399	**0.3962** ± 0.0282	**100**
0.5654 ± 0.0222	**0.3950** ± 0.0230	**200**

Tableau3.12-Evaluation de l'algorithme spherical-k-means avec LSI et pondération TF-IDF en utilisant la collection Cora

Le tableau 3.12 donne les résultats obtenus en appliquant l'algorithme Sphérical K-means, en utilisant la LSI et la pondération TF-IDF. Cette fois l'amélioration apportée par la LSI est plus importante puisque la NMI passe de 0.35 sans elle à 0.41 avec.

Les tableaux ci-dessous (3.13 et 3.14) donnent respectivement les résultats obtenus en appliquant l'algorithme K-means en utilisant la LSI sans pondération et les résultats obtenus en utilisant la LSI avec la pondération TFIDF. D'après le tableau 3.13, nous remarquons (pour k=100) que la valeur de a NMI est égale à 0.15 et qu'avec l'utilisation de la pondération TF-IDF (tableau 3.14) la valeur de la NMI est égal à 0.31, soit une augmentation de plus de 50%.

Algorithme k-means		
F-mesure	NMI	Dim. de LSI (k)
0.3784 ± 0.0057	**0.1559** ± 0.0215	**100**
0.3609 ± 0.0114	**0.1497** ± 0.0331	**200**

Tableau3.13-Evaluation de l'algorithme K-means, avec LSI et sans pondération en utilisant la collection Cora

Algorithme k-means		
F-mesure	NMI	degré
0.5115 ± 0.0659	**0.3154** ± 0.0495	**100**
0.4518 ± 0.0389	**0.3058** ± 0.0863	**200**

Tableau3.14-Evaluation de l'algorithme K-means, avec LSI et pondération TF-IDF en utilisant la collection Cora

8.4.2 Evaluation en utilisant la collection Citeseer :

Les tableaux3.15 et 3.16 indiquent respectivement les résultats obtenus en appliquant l'algorithme Sphérical K-means, en utilisant la LSI sans pondération et en utilisant la LSI avec la pondération TF-IDF. Là-aussi on remarque que la LSI combinée à a pondération TF-IDF permet d'améliorer les résultats et que, plus le degré de la technique LSI est grand, plus les résultats de F-mesure et la NMI sont meilleurs.

Algorithme Spherical-k-means		
F-mesure	NMI	degré
0.5409 ± 0.0554	**0.2447** ± 0.0314	**5**
0.5728 ± 0.0089	**0.2699** ± 0.0047	**10**
0.5611 ± 0.0340	**0.2646** ± 0.0291	**20**
0.5816 ± 0.0304	**0.2743** ± 0.0179	**50**
0.5855± 0.0224	**0.2748** ± 0.0199	**100**
0.5884± 0.0327	**0.2757** ± 0.0264	**200**

Tableau3.15-Evaluation de l'algorithme Spherical-k-means, avec LSI et sans pondération en utilisant la collection Citeseer

Algorithme Spherical-k-means		
F-mesure	NMI	degré
0.6860 ± 0.0001	**0.3759** ± 0.0001	**5**
0.6590 ± 0.0413	**0.3643** ± 0.0387	**10**
0.6613 ± 0.0495	**0.3727** ± 0.0317	**20**
0.6802 ± 0.0431	**0.3818** ± 0.0396	**50**
0.6919 ± 0.0365	**0.3916** ± 0.0322	**100**
0.7142 ± 0.0053	**0.4144** ± 0.0085	**200**

Tableau3.16-Evaluation de l'algorithme Spherical-k-means, avec LSI et pondération TF-IDF en utilisant la collection Citeseer

Les deux tableaux ci-dessous indiquent les valeurs de la NMI et de la F-lesure obtenues en appliquant respectivement l'algorithme K-means en utilisant la LSI sans pondération et en utilisant la LSI avec la pondération TF-IDF.

Nous remarquons que duo LSI-pondération TFIDF permet encore une fois d'améliorer les résultats mais que ces résultats restent inférieurs à ceux obtenus avec la version sphérique de K-means.

Algorithme k-means		
F-mesure	NMI	degré
0.4217 ± 0.0309	**0.1908** ± 0.0483	**100**
0.4277 ± 0.0421	**0.1920** ± 0.0445	**200**

Tableau3.17-Evaluation de l'algorithme k-means, avec LSI, sans TF_IDF et avec le graphe Citeseer.

Algorithme k-means		
F-mesure	NMI	degré
0.5069 ± 0.0485	**0.2654** ± 0.0470	**100**
0.5369 ± 0.0772	**0.2857** ± 0.0620	**200**

Tableau3.18-Evaluation de l'algorithme k-means, avec LSI, avec TF_IDF et avec le graphe Citeseer.

8.5 Evaluation de la convergence :

Les figures ci-dessous représentent les courbes de la fonction objective en fonction du nombre d'itérations de l'algorithme avec les collections Cora et Citeseer.

Convergence de Spherical-k-means :

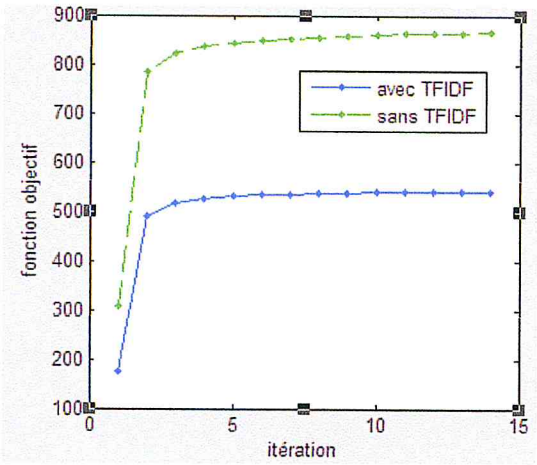


Figure 3.3 L'algorithme Spherical-k-means avec la collection Cora.

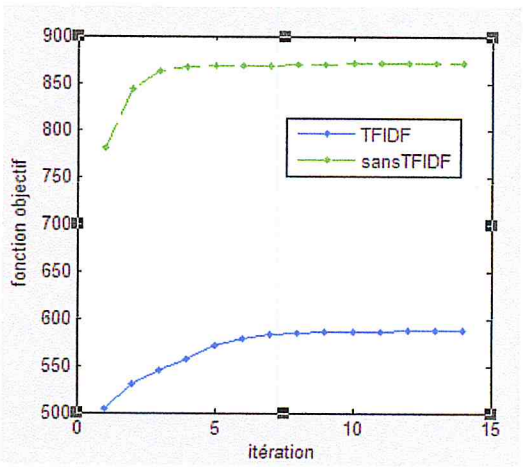


Figure 3.4 L'algorithme Spherical –k-means avec la collection Citeseer.

Nous remarquons que l'algorithme Sphérical-k-means avec la pondération TF_IDF converge plus vite avec les deux collections.

Convergence de spherical-k-means avec la pondération TF-IDF et la technique LSI

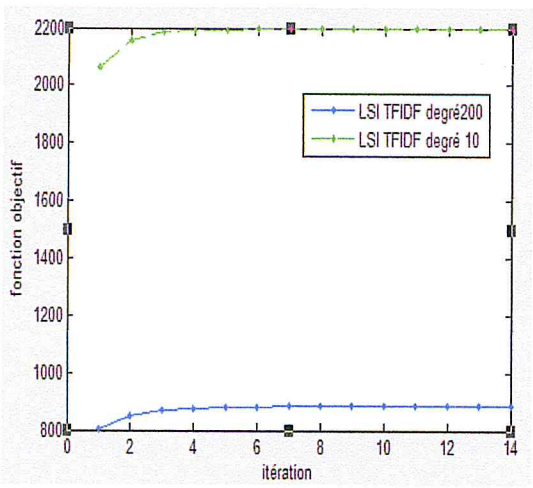


Figure 3.5algorithme Sphérical-k-means avec la pondération TF_IDF et la LSI avec la collection Cora

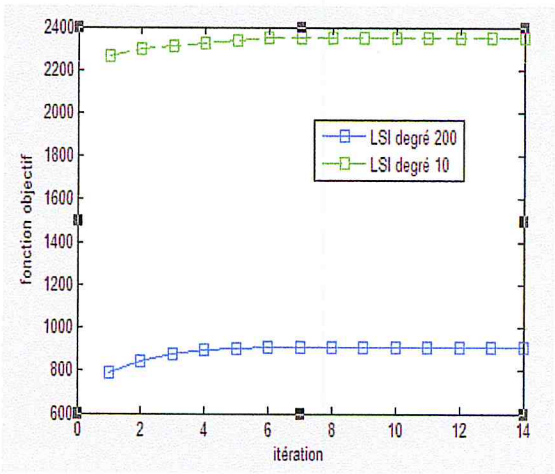


Figure 3.6 algorithme Sphérical-k-means avec la pondération TF_IDF et la LSI avec la collection Citeseer

Les figure ci-dessus montrent que la fonction objectif de l'algorithme Sphérical-k-means avec la pondération TF-IDF et LSI de degré 200 converge plus vite c'est-à-dire en un nombre d'itérations plus petit.

Convergence de k-means :

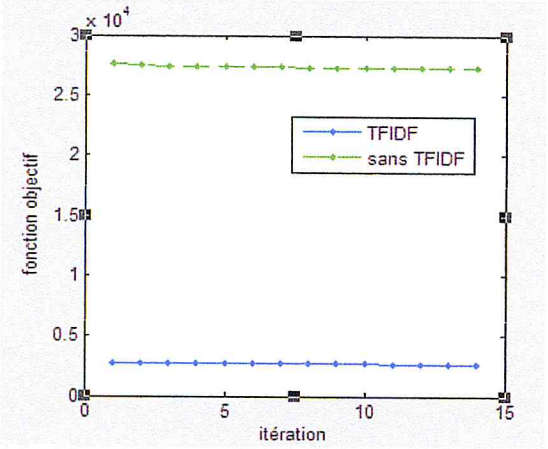


Figure 3.7L'algorithme k-means avec la collection Cora

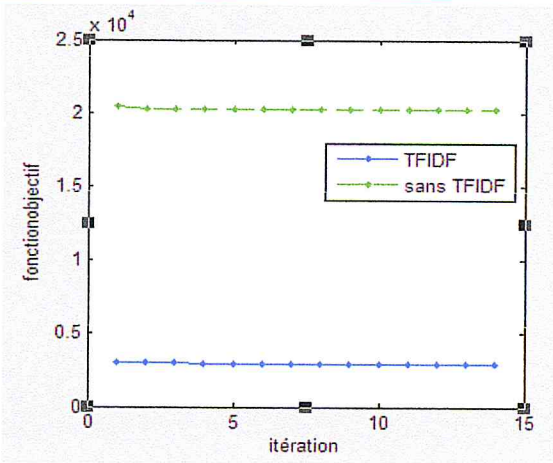


Figure 3.8L'algorithme k-means avec la collection Citeseer

La fonction objective de l'algorithme k-means avec la pondération TF-IDF converge aussi plus vite que sans la pondération.

Convergence de k-means avec la pondération TF-IDF et la technique LSI

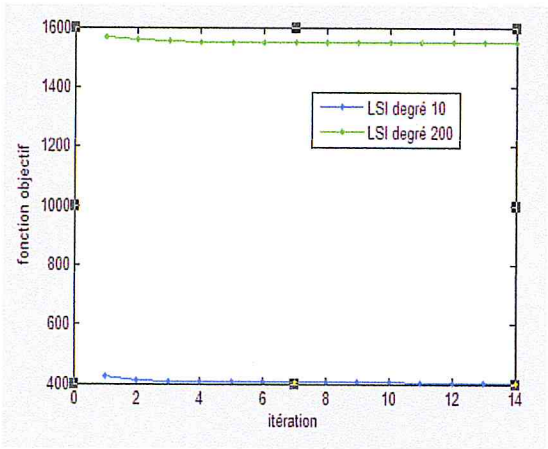


Figure 3.9 algorithme k-means avec la pondération TF_IDF et la LSI avec la collection Cora.

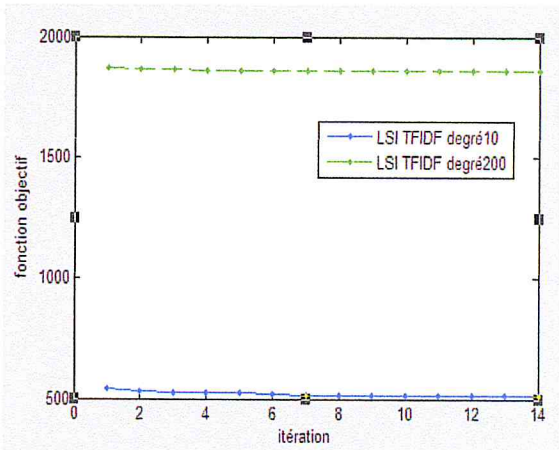


Figure 3.10 algorithme k-means avec la pondération TF_IDF et la LSI avec la collection Citeseer.

Conclusion générale

Nous avons abordé dans ce travail la problématique de clustering qui est connue aussi sous le nom de classification non supervisée. Il s'agit d'un domaine en plein essor dans lequel de nouvelles approches sont régulièrement publiées. Mais malgré le nombre important de techniques de clustering existantes, l'algorithme K-means, proposé à la fin des années 60 par MacQueen, reste sans doute l'une des méthodes les plus populaires et les plus utilisées par la communauté de data mining. Cependant, un des principaux inconvénients de l'algorithme K-means est le nombre de ses variantes qui ont été proposées par les chercheurs. Un utilisateur inexpérimenté souhaitant utilisé cet algorithme doit par exemple choisir une mesure de distance à utiliser ou encore une technique de normalisation des données pour que l'algorithme puisse effectuer le clustering correctement.

Nous nous sommes intéressés dans le cadre de ce travail l'analyse et la comparaison des différentes variantes existantes de l'algorithme K-means pour le clustering de documents textuels. Les textes représentent en effet un type particulier de données qui nécessitent des algorithmes adaptés à leurs propriétés.

Parmi les nombreuses variantes de l'algorithme K-means, nous avons retenu uniquement celles ayant une complexité linéaire par rapport au nombre de documents. Nous avons ainsi analysé les versions suivantes : K-means classique, K-means sphérique, K-means harmonique et Bisecting K-means. D'autres techniques tel que Fuzzy C-means ont été écartées en raison de leur inadéquation à l'analyse de données de dimensionnalité élevée tel le cas des données textuelles.

Pour notre étude comparative, nous avons utilisé deux corpus textuels et deux mesures d'évaluation. Nous avons analysé l'effet des méthodes de normalisation de la matrice termes-documents sur les performances des algorithmes. Nous avons aussi étudié l'effet de la techniques LSI (Latent Semantic Indexing) sur la qualité de clustering. La LSI étant une technique permettant de trouver des "liens" cachés entre les documents qui ne sont pas visibles dans l'espace vectoriel.

Les résultats obtenus sont sans appel : l'algorithme K-means sphérique combiné avec la pondération TFIDF est le meilleur par rapport à toutes les autres variantes, avec un léger avantage pour la version qui procède par bissection. Certains algorithmes comme le K-means harmonique qui sont adaptés à certains types de données se sont avérés complètement inadaptés à l'analyse de textes. Quant à la LSI, les résultats ont montré qu'elle permettait dans certains cas d'améliorer considérablement la qualité de clustering ainsi que la vitesse de convergence.

Bien que de nombreux aspects relatifs au clustering de textes avec K-means aient abordés dans ce travail, certains points qui nous semblent intéressants pourraient aussi être envisagés à l'avenir. Il serait par exemple intéressant de développer une technique permettant de déterminer de manière automatique les centroïdes initiaux car ceux-ci déterminent la qualité du clustering final. Nous avons en effet constaté lors de nos expérimentations que K-means (et toutes ses variantes) était très sensible à l'étape d'initialisation. Une autre direction de recherche concerne l'identification du nombre optimal de clusters K ; dans des applications réelles il est très difficile pour un utilisateur de connaitre ou de choisir ce paramètre.

Bibliographie

[Adj et Him 05] N.Adjeneg, M.Himeur «le clustering de document hiérarchique sur le web» mémoire ingéniorat, Université Saad Dehleb Blida, ref.MIG-004-068,2005.

[Aga 05] Bruno Agard, Andrew Kusiak, « Exploration des bases de données industrielles à l'aide du Data Mining –Perspectives », 9ème colloque national AIP PRIMECA, Avril 2005.

[Bae et Rib 00] R. Baeza-Yates & B. Ribeiro-Neto « Modern information retrieval». Ed. ACM Press, 2000.

[Bil 08] R. Bilisoly «practical text mining with perl» Ed JHON WILEY 2008.

[Bra et al. 98] S.P. Bradley, U.M. Fayyad & Reina "Scaling clustering algorithms to large data bases" in knowledge Discovery and data mining, page 9-15.1988.

[Cab 98] **M. T.** Cabré « La terminologie: théorie, méthode, application» les presses de l'université de Ottawa, 1998.

[Cha 04] C. Charles «une méthode d'aide à la navigation fondé sur Ω-means, algorithme de classification non supervisé. Application sur un corpus jurdique Français» thèse doctorat, l'Ecole des Mines de Paris, France, 2004

[Chi et al. 00] **K.** Chibout, J. Mariani, N. Masson & F. Néel« Ressources et évaluation en ingénierie des langages » Ed Duculot, 2000.

[Chi 10] N. Chikhi « Calcul de centralité et identification de structures de communautés dans les graphes de documents », Thèse Doctorat l'Université Paul Sabatier - Toulouse III, 2010.

[Cle 04] G. Cleuziou «une méthode de la classification non-supervisé pour l'apprentissage de règle et de la recherche d'information» thèse présentée pour doctorat de l'université d'Orléans, 2004

[Del 02] Jean-Michel Delorme propose par Mme Maguelome Teisseire «L'apport de la feuille de donnée dans l'analyse de textes »

[Did 84] E. Diday« une représentation visuelle des classes empiétantes: Les pyramides InRuiro » Analyse des données (vol 52), pages 475-526, 1986.

[Dum 93] S. T. Dumais. LSI meets TREC : A status report. In Proceedings of the 1st Text Retrieval Conference, pages 137–152, 1993.

[Dum 94] S. T. Dumais. Latent Semantic Indexing (lsi). In Proceedings of TREC-3, 1994.

[Fay 96] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy (Eds.); Advances in Knowledge Discovery and Data Mining; MIT Press; 1996.

[Fel 87] H. Felber« Manuel de terminologie » U.N.E.S.C.O, Centre international d'information pour la terminologie (Infoterm) Paris, 1987.

[Fel et San 07] R. Feldman & J. Sanger« The text mining handbook Advanced Approaches in Analyzing Unstructured Data » CAMBRIDGE UNIVERSITY PRESS 2007.

[Hir et al. 02] T. Hirao, H. Isozaki, E.Maeda & Y. Matsumoto « Extracting Important Sentences with Support Vector Machines » In COLING, p. 342-348 2002.

[Kar 07] S. Karbasi «Pondération des termes en Recherche d'Information: Modèle de pondération basé sur le rang des termes dans les documents **»,** Thèse Doctorat, l'Université Paul Sabatier - Toulouse III, 2007 ».

[L'Ho 04] M.C.L'Homme« La terminologie: principes et techniques » les Presses de l'université de Montréal, 2004.

[Lab 03] N. Labroche, «modélisation du système de reconnaissance chimique des fournis pour le problème de la classification non-supervisé: application à la mesure d'audience sur Internet», thèse doctorat, Université de Tours, 2003

[Lef 98] René Lefébure, Gilles Venturi, « Le Data Mining » Edition EYROLLES, deuxième tirage 1998.

[Lin 97] Data Mining: techniques appliquées au marketing, à la vente et au service client; Michael J.A. Berry & Gordon Linoff 1997.

[Los 88] R. M. Losee « Text Retrieval and Filtering Analytic Models of Performances Kluwer Academic Publishers, 1998 ».

[Mac 67] J. MacQueen «some methods for classification and analysis of multivariate observations» Proceedings of the 5thBerkeley Symposium on Mathematical statistics and probability, vol. 1, University of California Press, p.281-297, 1967.

[Mat] www. math.mcmaster.ca.

[Maz 04] Said MAZIZ et Lamri SIAD, « AntMining », Mémoire pour obtenir le grade d'Ingénieur d'état en Informatique de l'Institut National d'Informatique, Oued-Smar, Alger, Algérie. 2004.

[Mem 00] D.Memmi «Le model vectoriel pour le traitement de documents» Grenoble: Cahiers Leibniz, no 2004-14.

[Mic 05] Michel Jardinio, fouille de données dans le corpus de texte, 2005.

[Mor] http : //morgon.univlyon2.fr/Introduction_au_datamining_cours.htm

[Nak 07]. Nakache, Jean-Pierre: "A propos de l'ouvrage : Data Mining et statistique décisionnelle, de Stéphane Tufféry", *05-2007*

[Nik 06] Nicolas Beck, Application de méthodes de clustering traditionnelles et extension au cadre multicritère, mémoire d'ingéniorat en électromécanique, Bruxelles 2006.

[Phi 06] Philipe preux, fouille de données -note de cours. Université lille, 2006.

[STE 2002] Stéfaine Tuffery «Data Mining et scoring-base de donneé et gestion de la relion client Dunod » Paris 2002.

[Str 02] A.Strehl «Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining ». The University of Texas at Austin May 2002.

[Sup] www .si.suplec.fr

[Taq et Col 02] K. Taquechi & N. Collier« Use of support vector machines in extended named entity » In Proceedings of CoNLL-2002, pages 119-125. Taipei, Taiwan, 2002.

[Tuf 07] S. Tuffery «data mining et statistique décisionnelle» Ed. Technip, 2007.

[Wei et al. 05] S. Weiss, N.Indurkhya, T.Zhang, & F.Damerau « text mining predictive methods for analyzing unstructured information » Ed. Springer Science 2005.

[Zha et al. 05] Y. Zhang, J. X. Yu and J. Hou. Web Communities: Analysis and
American Society for Information Science, vol 41, pp. 391-407, 1990.

[Ref –Top 10 data mining algorithms].

[Url 1] : http://www.cs.umd.edu/~sen/lbc-proj/LBC.html.