جامعة سيدي محمد بن عبد الله بفاس
UNIVERSITE SIDI MOHAMED BEN ABDALLAH DE FES
كلية العلوم ظهر المهراز فاس
FACULTE DES SCIENCES DHAR EL MEHRAZ FES

N° d'Ordre : **/2020**

Formation Doctorale : Sciences et Technologies de l'Information et de Communication

Discipline : Informatique

Spécialité : Informatique

Laboratoire : Laboratoire d'Informatique, Signaux, Automatique et Cognitivisme

THESE DE DOCTORAT

Présentée par

Mme Awatif KARIM

Modèles intelligents à base des réseaux de neurones pour la classification des documents textuels

Soutenue le 12 / 04 / 2021 devant le jury composé de :

Pr. Omar EL BEQQALI	**Faculté des Sciences Dhar El Mehraz-Fès**	**Président**
Pr. Mohamed SABBANE	**Faculté des Sciences Meknes**	**Rapporteur**
Pr. El Habib NFAOUI	**Faculté des Sciences Dhar El Mehraz-Fès**	**Rapporteur**
Pr. Majda FIKRI	**École Nationale de Commerce et de Gestion-Agadir**	**Rapporteur**
Pr. Khalid HADDOUCH	**Ecole Nationale des Sciences Appliquées-Fès**	**Examinateur**
Pr. Mohammed Chakib SOSSE ALAOUI	**Centre Régional des Métiers de l'Education et de la Formation CRMEF-Fès**	**Examinateur**
Pr. Chakir LOQMAN	**Faculté des Sciences Dhar El Mehraz-Fès**	**Co-Directeur de thèse**
Pr. Jaouad BOUMHIDI	**Faculté des Sciences Dhar El Mehraz-Fès**	**Directeur de thèse**

Année universitaire : 2020-2021

Résumé de la thèse

Le nombre de données non structurées forme environ 90% de données disponible sur le web et sur les supports de stockage. Or la gestion de ce nombre énorme de documents numériques, qui ne cesse d'évoluer à chaque instant, nécessite les techniques d'intelligence artificielle pour pouvoir classifier et gérer automatiquement ces données de haute dimension. Nous proposons des nouveaux modèles intelligents à base des réseaux de neurones artificiels.

Nous avons utilisé dans la première contribution de la présente thèse, d'une part la théorie des graphes ; précisément le concept de l'ensemble stable maximum (MSSP) pour modéliser le problème de clustering de texte, et d'autre part les réseaux de Hopfield continu comme réseaux de neurones, pour détecter automatiquement le nombre de clusters et les centres initiaux du corpus proposé. Ces derniers seront les paramètres de base de K-Means dans la deuxième contribution. Notre approche a prouvé son efficacité et sa performance en termes de qualité de clustering et de temps d'exécution pour les grands ensembles de données.

Dans la dernière partie, nous avons réalisé une synthèse sur la structure et les implémentations techniques des systèmes de classification de textes basés sur l'apprentissage profond. L'objectif est d'examiner l'impact de nombreuses représentations de mots par rapport au plongement de mots contextuelle (BERT) sur la réalisation de classification de textes.

Mots-clés : Apprentissage non supervisé, Analyse de données, Classification, Document clustering, Text mining, Nombre de clusters, K-Means, Ensemble stable maximum, Réseaux de neurones, Réseau de Hopfield, Apprentissage profond, Représentation de texte.

Abstract

The number of unstructured data forms about 90% of the data available on the web and on storage media. Managing this huge number of digital documents, which is constantly evolving, requires artificial intelligence techniques to automatically classify and manage this high-dimensional data. We propose new intelligent models based on artificial neural networks.

In the first contribution of this thesis, we used graph theory; specifically, the concept of maximum stable set (MSSP) to model the text clustering problem, and continuous Hopfield networks as neural networks, to automatically detect the number of clusters and the initial centers of the proposed corpus. These will be the basic parameters of K-Means in the second

contribution. Our approach has proven its efficiency and performance in terms of clustering quality and execution time for large data sets.

In the last part, we performed a synthesis of the structure and technical implementations of text classification systems based on deep learning. The objective is to examine the impact of many traditional word representations versus contextual word embedding (BERT) on the realization of text classification.

Key Words: Unsupervised learning, Data analysis, Classification, Document clustering, Text mining, Number of clusters, K-Means, Maximum stable set Problem, Neural networks, Hopfield network, Deep learning, Text representation.

AVANT PROPOS

Ce travail s'inscrit dans le cadre d'une thèse de doctorat préparé au sein du laboratoire d'Informatique, Signaux, Automatique et Cognitivisme de la Faculté des Sciences Dhar El Mahraz de l'Université Sidi Mohamed Ben Abdellah de Fès (Maroc). Cette thèse a été dirigée par JAOUAD BOUMHIDI Professeur de l'Enseignement Supérieur à la Faculté des Sciences Dhar El Mahraz de Fès et CHAKIR LOQMAN Professeur de l'Enseignement Supérieur à la Faculté des Sciences Dhar El Mahraz de Fès.

Les travaux de cette thèse ont fait l'objet de publications dans des journaux internationaux et des communications à des congrès nationaux et internationaux.

REMERCIEMENTS

Tout d'abord, je tiens à remercier vivement l'encadrant de ma thèse, Monsieur Jaouad Boumhidi, Professeur de la faculté des sciences Dhar El Mehraz, pour avoir proposé le sujet de cette thèse. Je le remercie surtout du temps qu'il a consacré à guider et orienter mes travaux de recherche, ainsi que de la confiance qu'il m'a accordée durant la préparation de cette thése.

Je remercie vivement le co-directeur de ma thèse, Monsieur Chakir Loqman, Professeur de la faculté des sciences Dhar El Mehraz, pour les conseils, les orientations stimulants que j'ai eu l'honneur de recevoir de sa part lors de nos réunions. Je le remercie surtout pour les efforts déployés pour que cette thèse aboutisse dans les meilleurs délais.

Je remercie Mr. Omar EL BEQQALI, professeur de la faculté des sciences Dhar El Mehraz de Fès. Je le remercie de m'avoir fait l'honneur de présider le jury de ma thèse, malgré ses nombreuses occupations, pour assister à la soutenance.

Mes vifs remerciements s'adressent à Mr. Mohamed SABBANE professeur de la faculté des sciences Meknès, Mr. El Habib NFAOUI, professeur de la faculté des sciences Dhar El Mehraz de Fès qui ont bien voulu accepter d'évaluer le présent travail et pour avoir accepté de rapporter ma thèse malgré leurs préoccupations. Je les suis reconnaissante pour la lecture très attentive du manuscrit, pour leurs questions et remarques constructives, qui m'ont permis d'améliorer certaines parties de mon manuscrit.

Je tiens à remercier aussi Mme Majda FIKRI, professeur de l'école nationale de commerce et de gestion d'Agadir, pour le temps précieux qu'il m'a accordé en acceptant d'être mon rapporteur et par les remarques très utiles qu'il a fait et qui m'ont permis d'enrichir ce manuscrit, ainsi que ses critiques qui m'ont inspiré plusieurs idées pour des prochains travaux.

Mes remerciements vont aussi à Mr.Khalid HADDOUCH, professeur de l'école nationale des sciences appliquées de Fès, et Mr. Mohammed Chakib SOSSE ALAOUI, professeur du centre régional des métiers de l'éducation et de la formation de Fès, d'avoir consacré de ces précieux instants à la lecture de cette thèse pour examiner ce travail.

Je voudrais également remercier tous mes amis et les doctorants du notre groupe de recherche, pour leurs conseils et leurs aides.

Finalement, je remercie mes parents pour leurs encouragements et leur assistance qui m'ont permis de faire cette thèse dans de bonnes conditions. Je remercie chaudement mes chers frères et ma sœur unique Jamila. Je remercie mon cher époux Youssef pour son soutien quotidien indéfectible et son enthousiasme contagieux à l'égard de mes travaux. Je remercie aussi bien mes petites filles Zaynab et Nada pour leur amour et leur support.

Que toute la Famille KARIM et la Famille HAMI trouve ici mes sincères remerciements pour leur épaulement et encouragement continu.

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GENERALE

De nos jours, l'analyse des données est devenue un atout courant qui constitue un cadre de richesse des organismes socio-économiques. Les informations accessibles et les documents électroniques sont devenus pour ces organisations un facteur de compétitivité et de création de valeur. Notamment sur le web les masses de données textuelles aujourd'hui disponibles engendrent un problème difficile lié à leur traitement automatique. Le problème aujourd'hui n'est plus d'accéder aux informations mais de caractériser ces dernières et de déterminer l'information utile ou d'extraire les connaissances considérées désormais comme un capital qui a une valeur économique et un statut stratégique pour l'entreprise. Bien entendu, en fonction des besoins de l'utilisateur cette information pourra être utilisée de différentes manières : filtrage de documents, classification de documents, etc. Par exemple la société Amazon a toujours besoin de connaître les préférences et les avis des clients sur les produits exposés sur le site web en fonction de leurs comportements. Par conséquent, de nombreuses techniques d'apprentissage automatique ont été développées pour analyser les grands ensembles de données. Dans cette thèse, nous nous intéressons particulièrement aux techniques d'apprentissage non supervisé qui permettent d'organiser efficacement des ensembles de données de tailles importantes sans connaissance à priori sur les données traitées [Biernat et Lutz, 2016].

Dans le clustering, l'un des problèmes les plus difficiles à résoudre est la détermination du nombre de clusters dans un ensemble de données, qui est considéré comme un paramètre d'entrée de base pour la plupart des algorithmes de clustering [Dumont et al., 2018]. Ces algorithmes ont besoin d'un nombre de clusters spécifié à l'avance par l'utilisateur, puis ils sélectionnent à l'avance les centres initiaux de façon aléatoire, ce qui affecte considérablement la qualité des résultats de regroupements. Celle-là dépend fortement du nombre de classes et du choix aléatoire des centres initiaux. Spécialement quand le jeu de données est grand et qu'on n'ait pas à priori des hypothèses sur les données [Ashour et Fyfe, 2014]. Pour pallier à cette carence, nous proposons une méthode de détection automatique du nombre de clusters et des centres initiaux, qui sont les paramètres d'entrée de l'algorithme classique de K-Means. L'approche proposée est exécutée avant d'entamer les méthodes de regroupement, ce qui signifie que notre approche est indépendante de toute méthode de clustering qui commence par k centres, qu'elle est efficace pour les grands ensembles de données et qu'elle est très optimisée en termes de temps.

Les travaux de recherche présentés dans cette thèse se situent dans le cadre de l'intelligence artificielle, la recherche d'information et l'analyse de données, notamment la classification des documents basée sur l'apprentissage non supervisé. Dans ce contexte, nous proposons diverses améliorations de classification basées sur des réseaux de neurones artificiels et des modèles mathématiques. Ces modèles portent sur des paramètres qui déterminent le nombre de clusters ainsi que les centres initiaux nécessaires à toute méthode de clustering.

Ce document comporte deux parties :

La première partie est composée de trois chapitres consacrés à la présentation des méthodes de résolution des problèmes de classification non supervisée.

La deuxième partie de cette thèse est consacrée à l'étude et à la résolution des problèmes de détermination du nombre optimal de classes lors du regroupement non supervisé. Elle est composée de deux chapitres.

Le premier chapitre est consacré à l'état de l'art de la Catégorisation de Textes (CT). Nous présentons au début une définition détaillée de catégorisation de textes. Puis nous passons en revue tout le processus d'un système de CT y inclut la représentation des documents, les calculs de pondérations pour faire ressortir les attributs potentiellement discriminants, les principales techniques appliquées pour réduire la taille du vocabulaire pris en compte, les principaux algorithmes d'apprentissage ayant fait leurs preuves dans le domaine de CT et les métriques d'évaluation utilisées pour comparer entre eux

Le deuxième chapitre a pour objet de présenter les notions élémentaires et les termes relatifs à la description de l'apprentissage non supervisé. Nous y exposerons tout d'abord les types de classifications non supervisées. Nous commençons par les méthodes hiérarchiques et les méthodes de partitionnement. Ensuite, les étapes du clustering à savoir le prétraitement des données, le choix de l'algorithme puis les critères d'évaluation ou les indices de validité du clustering ainsi que leurs performances. A la fin, nous mettons l'accent sur le problème le plus difficile dans la classification non supervisée concernant la détermination du nombre de classes à retenir pour une base de données.

Les masses de données textuelles aujourd'hui disponibles engendrent un problème difficile lié à leur traitement automatique. Des méthodes de Fouille de Textes (FT) et de Traitement Automatique du Langage (TAL) peuvent en partie répondre à une telle problématique. Elles consistent à modéliser puis mettre en œuvre des méthodologies appliquées aux données

textuelles afin d'en déterminer le sens et/ou découvrir des connaissances nouvelles. La plupart des méthodologies proposées s'appuient sur des approches linguistiques et/ou statistiques.

Le troisième chapitre constitue les composantes principales de notre modèle intelligent. Il est dédié à la présentation des bases principales de la théorie de graphes pour définir ce qu'on appelle l'ensemble stable ou indépendant dans un graphe. Après, nous décrivons les différentes approches pouvant être employées pour résoudre le problème de l'ensemble stable de taille maximale, tout en distinguant les différentes formulations du problème de l'ensemble stable maximal en des programmes quadratiques. Nous insistons notamment sur les méthodes des réseaux de neurones artificiels en particulier le réseau neuronal de Hopfield continu. Nous terminons ce chapitre par la conception et le fonctionnement de ce réseau et surtout en précisant le lien et la formulation des problèmes d'optimisations sous forme de l'énergie du réseau de Hopfield continu en donnant les poids et les seuils associés.

Dans **le quatrième chapitre**, nous proposons un modèle intelligent à base des réseaux de neurones pour la détection automatique du nombre de clusters et des centres initiaux lors des algorithmes de regroupement [Karim et al., 2018], spécialement les paramètres d'entrée de l'algorithme classique de K-Means. La résolution de ce problème se fait en quatre étapes :

- La première consiste à construire un graphe permettant de déterminer un ensemble stable de taille maximale,
- La deuxième modélise le problème de détermination du nombre de clusters comme un problème d'ensemble stable maximum (MSSP), ce dernier peut être modélisé en termes d'un problème quadratique à variable binaires 0-1,
- La troisième étape consiste à appliquer le réseau neuronal de Hopfield continu pour résoudre le programme quadratique obtenu. Par conséquent, la fonction d'énergie généralisée associée au réseau de Hopfield continu et une procédure de paramétrage appropriée concernant le problème de MSSP sont données,
- La quatrième étape concerne l'amélioration de la solution obtenue comme paramètre d'entrée dans l'algorithme de K-Means. La résolution du modèle proposé par les réseaux de Hopfield continus fournit une bonne initialisation aux centres de l'algorithme de K-Means et le nombre adéquat des classes dans la base de données.

La seconde partie sera accordée à la validation des performances du modèle proposé dans la première partie de ce chapitre. La validation portera sur la notion des indices des critères internes, externes et relatifs. Les performances des résultats obtenus par cette méthode seront comparées avec les résultats obtenus par l'algorithme de K-Means.

Dans **le dernier chapitre**, on propose une étude comparative pour examiner l'impact de nombreuses représentations de mots (BOW, plongement de mots traditionnel (GloVe, Word2Vec) et plongement de mots contextuel (BERT) ainsi que des approches de classification (apprentissage profond par rapport aux méthodes traditionnelles d'apprentissage automatique) sur la réalisation de classification de textes [Karim et al., 2021].

Nous terminons ce mémoire de thèse par une conclusion générale, où nous suggérons aussi des perspectives et développements futurs pour la suite de nos travaux de recherche.

PARTIE I : APPRENTISSAGE NON SUPERVISE ET FOUILLE DE TEXTES

CHAPITRE I : FOUILLE DE TEXTES

1 Introduction

Avec le nombre important et en croissance rapide des documents disponibles sous format numérique, tant sur internet que sur les systèmes d'information d'entreprises, la fouille de textes est devenue un domaine de recherche important et nécessaire. La fouille de textes en anglais appelée Text Mining est une spécialisation de la fouille de données, et fait partie du domaine de l'intelligence artificielle. L'objectif commun de toutes les recherches sur la fouille de texte est le passage de grands volumes de textes à de la connaissance, afin d'aider les experts à enrichir ses modèles de connaissances ou à effectuer toute autre tâche de raisonnement. La fouille de textes est apparue dans la deuxième moitié des années 90, dont l'usage des termes Extraction de connaissances à partir de bases de données (ECBD) se précise par [Feldman et Dagan, 1995]. L'ECBD désigne alors la transmission de données brutes à des connaissances alors que la fouille de données n'est qu'une étape de l'ECBD où un modèle est construit [Fayyad et al., 1996].

En fouille de textes comme en fouille de données, tout est quantifiable, les différentes solutions envisagées peuvent être évaluées et comparées. De ce fait, elle vise à tirer le meilleur profit possible pour créer des programmes capables de prendre des décisions pertinentes. Par exemple, dans les banques et les assurances pour l'attribution d'un crédit, la médecine pour effectuer un diagnostic ou évaluer l'efficacité d'un médicament ou encore dans le marketing pour cibler la clientèle.

La fouille de textes représente l'ensemble des technologies et méthodes destinées au traitement automatique de gros volumes de contenus textuels numériques (des textes, des données, des sons, des images ou d'autres éléments, ou une combinaison de ceux-ci) en langage naturel, en vue d'extraire et de structurer le contenu dans la perspective d'analyse rapide et découverte d'informations cachées ou prise automatique de décision. La fouille de textes est le processus de génération d'informations de haute qualité. Ces informations sont typiquement dérivées par la conception de modèles basés sur l'apprentissage de modèles statistiques [Aggarwal et Zhai, 2012], [Cohen et Hunter, 2008]. L'évaluation et l'interprétation de la qualité d'un tel programme se mesurera à sa capacité à s'approcher d'une solution de référence validée par l'être humain.

La catégorisation de texte est une tâche importante de la fouille de données textuelles. Elle consiste à structurer et caractériser les textes d'entrée en vue de l'insérer dans une base de données structurée, on peut citer le regroupement des messages d'actualités de plusieurs fournisseurs d'informations en fonction du sujet.

Une extension spéciale de la fouille de données textuelles est la fouille de données web. Il s'agit d'informations textuelles collectées en exploitant le Web pour découvrir des modèles à partir de l'analyse des opinions des navigateurs [Srivastava et al., 2005]. La fouille de web est répartie en trois types essentiels, à savoir la fouille d'utilisation web, la fouille de contenu web et la fouille de structure web [Han et al., 2011].

Nous identifions les tâches typiques et les applications de la fouille de données textuelles les plus connues : la tâche du clustering ou de la classification de documents au profit de la recherche d'information et de l'extraction de concepts et de connaissances, qui a été massivement utilisée pour la navigation sur le web [Cuttung et al., 1992], le résumé ou la synthèse automatique de textes [Kummamuru et al., 2004] et la recherche distribuée [Xu et Croft, 1999]. D'autres applications habituelles comme, les systèmes Questions/Réponses, la traduction automatique, la détection de spam dans les messages électroniques, etc.

Ce chapitre est organisé de la façon suivante. Dans la section 2, nous présentons un processus général de l'extraction de connaissances à partir des textes. La section 3 présente des méthodes de réduction de la dimension après la phase de pondération des termes. La section 4 aborde les différents modes de représentations de textes. Enfin, nous présentons les principales propriétés de la similarité entre documents textuels, et quelques mesures permettant de calculer quantitativement ces similarités.

2 Processus de la fouille de textes

L'interprétation et la classification automatique des données textuelles sont particulièrement des tâches compliquées [Witten et Frank, 2005], c'est pourquoi la fouille de textes doit passer par plusieurs étapes brièvement décrites ci-dessous.

2.1 Les étapes du processus de la fouille de textes

Toutes ces étapes ne s'emploient pas toujours ensemble et le choix des étapes nécessaires dépend de l'objectif et du contexte de la problématique. A partir des collections de données textuelles jusqu'à l'évaluation et l'interprétation des solutions envisagées, le processus de text

mining construit un modèle précis qui fait un prétraitement et / ou réduit les données considérées en vue de faire une analyse pertinente des données textuelles (Figure **I.1**).

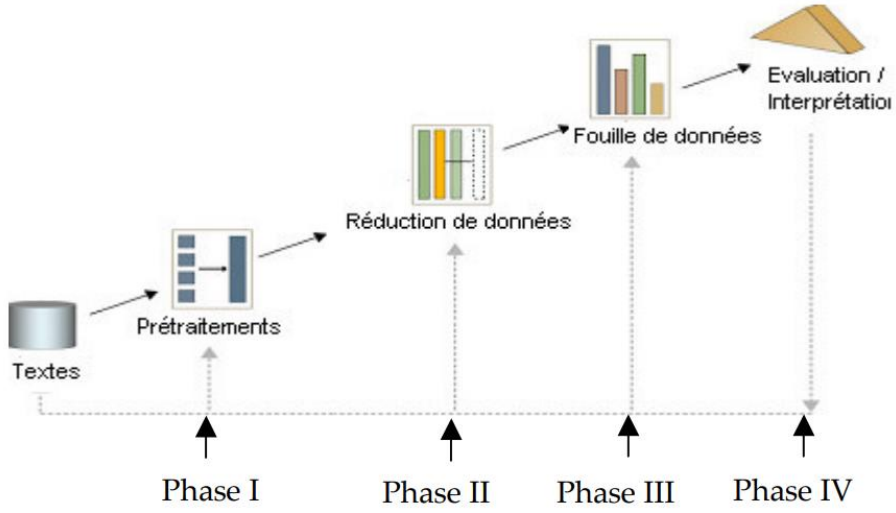


Figure I.1. Processus d'extraction de connaissances à partir de documents textuels

Certaines étapes (telles que la collecte ou le prétraitement de données) sont toujours présents dans la fouille de texte et dépendent de la nature des données.

2.2 Prétraitement des textes

Il y a plusieurs tâches de prétraitement à effectuer avant l'application des algorithmes d'exploration de données. Nous présentons ici les prétraitements qui peuvent être employés seuls ou encore combinés entre eux. Il s'agit notamment des tâches suivantes :

-Collecte de données textuelles : cette étape consiste à récupérer le contenu de plusieurs sources identifiées et à extraire le texte du contenu obtenu. Par exemple, les pages Web contiennent également d'autres données, telles que des balises HTML, des menus et parfois des scripts JavaScript, etc. Ces données doivent généralement être séparées de texte à analyser.

-Décomposer le texte en éléments ou unités de mots appelés "tokens". Nous divisons notre texte en mots en utilisant les espaces comme délimiteur, pour le convertir ensuite en un sac de mots (en anglais appelé Bag-Of-Words). Par exemple, la segmentation du document texte ou de la phrase suivante :

« L'approche proposée est exécutée avant le regroupement. »

Donne le sac de mots suivant :

« L », « ’ », « approche », « proposée », « est », « exécutée », « avant », « le », « regroupement », «. ».

-Éliminer tout symbole qui ne correspond pas à une lettre de l'alphabet (points, virgules, traits d'union, chiffres, etc). Cette opération est motivée par le fait que ces caractères ne sont pas liés au contenu des documents et ne change rien au sens s’ils sont omis et par conséquent ils peuvent être négligés.

Le sac de mots précédent devient :

« L », « approche », « proposée », « est », « exécutée », « avant », « le », « regroupement ».

-Supprimer des mots-vides qui n’ont pas de sens significatif et qui souvent sont trop fréquents dans des textes, par exemple (articles, prépositions, mots grammaticaux...). Dans ce sens, on peut créer un dictionnaire de mots vides pour un domaine spécifié appelé en anglais "stop-words".

On obtient pour le sac de mots utilisé :

« approche », « proposée », « est », « exécutée », « regroupement ».

-La racinisation appelé en anglais "stemming" est un procédé de transformation des termes en leur radical ou racine (suppression des préfixes et des suffixes des éléments) comme par exemple : national, nationalité et nationalisation sont remplacées par leur racine « national » et les verbes conjugués par leurs infinitifs. Le "stemming" n'a pas d'impact sur la masse des mots, mais réduit de 30% en moyenne la taille du document. Nous avons utilisé l’algorithme de Porter pour remédier à cette étape [Porter, 1980].

Le sac de mots devient :

« approche », « proposer », « être », « exécuter », « regroupement ».

-Approche morphosyntaxique : Il comprend une analyse morphologique et une analyse syntaxique [Brill, 1992]. Notons que ces deux analyses sont précédées par certains prétraitements (traitement des ponctuations, majuscules, codages et formats). Une analyse morphologique peut être considérée comme un automate qui traite isolément chaque forme d’un texte en lui associant des traits informationnels ou des propriétés [Fay-Varnier et al., 1991]. L’analyse syntaxique permet de segmenter les textes en propositions. Chaque proposition est formée de couples (entrée lexicale, catégorie). Les seules ambiguïtés qui demeurent sont internes à une catégorie. Les résultats de l’analyse des propositions sont des arborescences de

structures syntaxiques attestées par la langue [Habert et al., 1997]. A la fin, l'étiquetage morphosyntaxique associe à chaque mot d'une phrase sa catégorie morphologique (genre, nombre) et syntaxique (nom, adjectif, verbe, etc.).

Rappelons le document de l'exemple :

« L'approche proposée est exécutée avant le regroupement. »

Une analyse morphosyntaxique de ce document donne :

« L » est un article défini féminin singulier.

« approche» est un nom féminin.

« proposée » est un verbe au participe passé à valeur d'adjectif.

« est » est un verbe à l'indicatif présent au 3éme personne du singulier.

« exécutée » est un verbe au participe passée.

« avant » est une préposition.

« le » est un article défini masculin singulier.

« regroupement» est un nom ou substantif masculin singulier.

«. » est une ponctuation forte désigne fin de phrase.

-Ensuite nous avons la lemmatisation qui est une réduction des mots à leur forme canonique. Elle consiste à remplacer chaque mot du document par son synset (synonyme dans la base lexicale). On utilise WordNet comme une base de données lexicale [Miller, 1995], pour avoir des descripteurs moins sensibles au bruit et des meilleures similarités entre documents.

Revenons à la phrase traitée:

Le lemme « **regroupement** » donne 2 formes fléchies : regroupement, regroupements.

Le lemme « **exécuter** » donne plusieurs formes : exécute, est exécutée, exécutant, exécuterai, etc.

Après la phase de prétraitement, un texte devient une succession de « termes » dans la suite, et on passe à la phase de représentation vectorielle de textes.

3 Pondération des termes

Afin de représenter quantitativement les termes dans les documents, il faut décider comment pondérer chaque terme de ce vecteur. Nombreuses études ont été axées sur les calculs des poids et les pondérations dans un corpus de documents [Singhal, 1997], [Lee, 1995], [Buckley et al., 1992], [Salton et Buckley, 1998].

Deux mesures quantifiables combinées en produit sont fréquemment utilisées pour la pondération des termes à savoir :

- La valeur TF (TermFrequency) qui décrit la fréquence d'apparition d'un terme dans un document par rapport à tous les autres termes. Elle est définie par :

$$TF_{ij} = \frac{n_{ij}}{|d_j|}$$

  Avec n_{ij} le nombre d'occurrences du terme **i** dans le document j et $|d_j|$ la longueur du document d_j.

- La valeur IDF (Inverse Document Frequency) qui mesure la signification d'un terme en fonction de son utilisation dans l'ensemble des documents. Elle est calculée comme suit :

$$IDF_i = \log\left(\frac{t}{t_i}\right)$$

  Avec t le nombre total de documents et t_i le nombre de documents contenant le terme i.

- La multiplication de ces deux dernières valeurs donne naissance à une méthode de pondération appelée TFIDF, qui est très utilisée en recherche d'information et en particulier dans la fouille de textes. Cette mesure statistique permet de représenter quantitativement à la fois la fréquence relative d'un terme dans un document et la fréquence du terme dans le corpus. L'expression du TF-IDF fréquemment utilisée pour ce coefficient est :

$$TFIDF = TF_{ij} \times IDF_i$$

  Cette méthode permet alors d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus.

4 Réduction des dimensions

Après la phase de la pondération des termes, la phase de la réduction des dimensions est primordiale vue son intérêt, d'une part, elle permet d'écarter les termes non pertinents d'un point de vue statistique par exemple la suppression des mots plus fréquents et des mots très rares (nombre d'occurrences< 3). D'autre part, elle permet d'éviter le sur-apprentissage afin d'améliorer l'efficacité des algorithmes d'apprentissage ayant des difficultés à gérer un espace de représentation important.

La réduction des dimensions est un problème crucial pour la catégorisation de textes et l'apprentissage en général. Cependant, il existe des techniques de la réduction des dimensions pour choisir les mots utiles pour discriminer entre documents pertinents et non pertinents, on cite par exemple :

- La statistique du Chi-2 qui détermine les termes les plus caractéristiques de chaque catégorie.

- L'indexation sémantique latente.

4.1 La statistique de Chi-2

Les tests statistiques sont des techniques largement utilisées pour expliquer la dépendance entre les variables. La statistique du Chi-2 mesure l'écart à l'indépendance entre une caractéristique t et une classe C_j. Cette mesure notée χ^2, permet l'association d'un mot à une classe, autrement dit permet d'identifier pour chaque classe les mots et les caractéristiques des textes les plus représentatifs. C'est une mesure statistique bien connue, elle s'adapte bien à la sélection d'attributs, car elle évalue le manque d'indépendance entre un mot et une classe.

La formule suivante compare les proportions de documents contenant le terme (t_k) ou ne contenant pas ce terme $\overline{(t_k)}$, dans la classe (C_j) ou dans les documents des autres classes $\overline{(C_j)}$.

$$\chi^2(t_k, C_j) = \frac{n[P(t_k, C_j)P(\overline{t_k}, \overline{C_j}) - P(\overline{t_k}, C_j)P(t_k, \overline{C_j})]^2}{P(t_k)P(\overline{t_k}) - P(C_j)P(\overline{C_j})}$$

Où :

$P(t_k, C_j)$ (respectivement $P(t_k, \overline{C_j})$) : représente la probabilité des documents contenant le terme t_k(respectivement non) dans la catégorie C_j.

$P(\overline{t_k}, C_j)$ (respectivement $P(\overline{t_k}, \overline{C_j})$) : représente la probabilité des documents qui ne contiennent pas le terme (t_k)(respectivement non) dans la catégorie C_j.

Ces probabilités sont exactement les cardinaux des documents où figurent le terme t (ou non) dans une catégorisation choisie C (ou non plus) par rapport au nombre total de documents n.

Deux configurations ou scores sont ensuite possibles pour évaluer l'indépendance de t_k par rapport à l'ensemble des classes C_j :

$$\begin{cases} \chi^2(t_k) = \sum_j P(C_j)\,\chi^2(t_k, C_j) & \text{(Moyenne Pondérée)} \\ \chi^2_{max}(t_k) = \max_j \chi^2(t_k, C_j) & \text{(Maximum)} \end{cases}$$

Par conséquent, les mots sélectionnés sont ceux qui sont les moins indépendants des classes correspondent aux valeurs élevées pour ces mesures.

Cette technique de réduction de dimensions est la plus utilisée dans le domaine de la catégorisation de textes (classification supervisé). Elle a comme inconvénient qu'elle ignore les relations entre les termes, car parfois un terme qui n'est pas utile par lui-même peut être utile dans son contexte.

4.2 Indexation sémantique latente (LSI)

LSI est une méthode mathématique permettant d'extraire et d'identifier les relations contextuelles cachées entre les mots dans le domaine de traitement du langage naturel (NLP). Cette approche a été proposée par [Deerwester et al., 1990], et utilisée pour effectuer de la recherche d'informations puisque les termes apparaissant ensemble sont projetés sur la même dimension. Il s'agit d'une approche vectorielle qui permet la caractérisation du sens des mots en regroupant les termes co-occurrents (similaires) dans les mêmes dimensions, et produisant des relations mot à mot bien corrélées avec la similarité sémantique [Landauer et al.,1998]. Par rapport à d'autres applications de recherche d'informations, LSI peut en tirer une méthode précise pour extraire et représenter le sens du texte. La méthode pratique est décomposée de la manière suivante :

4.2.1 Construction de la matrice des occurrences

Soit X la matrice documents-termes (Figure **I.2**) dans laquelle chacune des m lignes correspond à un document et chacune des n colonnes à un terme, où l'élément (i, j) est la pondération de type TF-IDF.

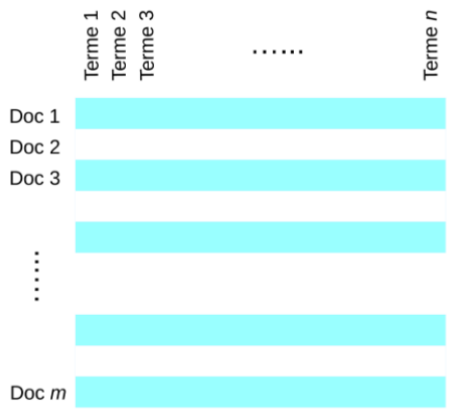


Figure I.2 **Matrice terme-document**

4.2.2 Décomposition en valeurs singulières

On effectue alors une décomposition en valeurs singulières de la matrice X, qui donne deux matrices orthogonales U et V et une matrice diagonale S, tels que :

$$X = USV^t$$

4.2.3 Espace des concepts

Lorsqu'on sélectionne les k plus grandes valeurs singulières, ainsi que les vecteurs singuliers correspondants dans U et V, on obtient une approximation de rang k de la matrice des occurrences.

5 Représentation du texte en TALN

La représentation vectorielle du texte est destinée à promouvoir l'application de méthodes de fouille de texte. Il existe plusieurs techniques différentes, démarrant avec le modèle classique de Salton [Salton et McGill, 1986], puis les méthodes de réduction de dimensionnalité comme l'analyse sémantique latente [Deerwester et al., 1990], arrivant à la représentation compacte de type Word2Vec [Mikolov et al., 2013a], [Mikolov et al., 2013b].

5.1 Représentation en sac de mots

Le sac de mots ne donne aucune information linguistique, ce n'est qu'un vecteur de la taille du corpus étudié, dans lequel on affecte selon la présence du mot dans le document, 1 à chacun des mots présents dans le texte, 0 sinon. Ce type de sac est appelé sac de mots binaire. Le sac de mots avec fréquence détermine de manière quantitative la représentativité d'un terme dans le

corpus. Aussi, il contient le nombre d'occurrences de chaque mot dans le document ce qui correspond à la fréquence du terme ou le critère TF.

5.2 Représentation des textes avec les racines lexicales

Contrairement au modèle précédent (Représentation en sac de mots), chaque flexion est considérée comme descripteur différent et une dimension de plus. La représentation lexicale cherche à résoudre cette difficulté en considérant uniquement la racine des mots plutôt que les mots entiers. Plusieurs algorithmes ont été proposés pour substituer les mots par leur racine : le plus connu pour la langue anglaise est l'algorithme de Porter stemmer [Porter, 1980].

5.3 Représentation à base des lemmes

La représentation à base des lemmes consiste à remplacer les mots en leurs formes canoniques en utilisant l'analyse grammaticale. Grâce à la recherche des racines de mots, la lemmatisation sert à remplacer les verbes par leurs formes infinitives et les noms par leurs formes au singulier.

L'algorithme nommé TreeTagger, a été développé en utilisant l'arbre de décision et un dictionnaire spécifique à chaque langue. Il a montré son efficacité pour les langues anglaise, française, allemande et italienne.

Même si la substitution des mots par leurs racine ou leur lemmes réduit l'espace de représentation d'une part, qui est en général assez grande. Ce qui rend la plupart des algorithmes de classification difficiles à utiliser [Kim et al., 2008]. D'autre part, cette substitution peut augmenter l'ambiguïté et perd d'information sur le contexte puisqu'elle ignore les liens sémantiques entre les mots au sein d'une phrase. Par exemple, le mot "actions" indique souvent des actions des firmes, mais sa racine "action" peut être loin de ce concept si elle est employée dans la phrase : "le domaine d'action du gouvernement". Ce qui rend nécessaire d'utiliser une représentation conceptuelle des textes.

5.4 Représentation par concepts

Cette représentation appelée en anglais « Bag Of concepts », a pour objectif d'améliorer la pertinence de la représentation des documents textes. Elle permet de représenter et d'intégrer des aspects sémantiques d'un lexique, contrairement à la représentation en « sac de mots ». Par ailleurs, des meilleurs résultats sont obtenus par extraction de concepts, avec écarts faibles entre ces diverses représentations.

Selon [Réhel, 2005] l'avantage principal de cette représentation est de réduire relativement la dimension de l'espace car chaque document texte contient des concepts dont l'expression pouvant contenir un ou plusieurs mots. Les chaînes de Markov ont été utilisées dans un modèle probabiliste pour intégrer les sens des mots [Besançon et al., 2001]. La représentation par concepts a été développée avec succès dans le domaine médical [Pouliquen, 2002]. Cependant, l'inconvénient majeur de cette représentation est qu'il n'existe pas de ressources sémantiques pour toutes les langues.

La représentation par concepts se base sur le formalisme vectoriel et des outils de statistiques bien adaptés pour représenter les documents textuels. L'annotation de ces documents se fait via les éléments du vecteur, qui sont associés directement à des termes en vue d'améliorer leur rattachement à des concepts ou ceux que l'on trouve à proximité d'autres.

Pour obtenir une telle représentation des documents, il est nécessaire de créer des pochettes ou des sacs de concepts pour chacun des documents. Pour cela, il faut projeter les termes dans une ressource sémantique telle que WordNet, en vue d'avoir de groupes de synonymes des mots.

6 Similarité entre documents

Le domaine de l'identification de la similarité a été considéré comme un sujet de recherche fortement recommandé dans plusieurs domaines d'application comme le traitement automatique du langage naturel, la bio-informatique, le web services ou l'extraction de connaissances à partir de données textuelles [Latrache et al., 2014]. Dans chacun de ces domaines, les similarités sont utilisées pour différents traitements. Il existe deux grands types de similarités : La similarité syntaxique et la similarité sémantique.

6.1 Similarité syntaxique

La similarité syntaxique est une métrique qui mesure la similarité ou la dissimilarité entre deux chaînes de caractères. Par exemple, les chaînes de caractères "Cheval" et "Cheveu" peuvent être considérées comme très proches, alors qu'ils sont très différents. Parmi les mesures syntaxiques les plus connues, nous trouvons : la distance euclidienne, la similarité cosinus, la formule de Salton et l'indice de Jaccard, etc. Ces trois dernières formules se basent principalement sur le nombre de propriétés communes ou non entre les documents. D'autres mesures de distances existantes dans la littérature comme la distance de Levenshtein et le coefficient de Dice sont présentés en détail ainsi que leurs propriétés dans [Rajman et Lebart, 1998], [Morin, 1999].

6.1.1 Distance Euclidienne

Il s'agit de la mesure de distance entre deux vecteurs dans un espace euclidien de dimension finie n, elle est la plus classique entre deux vecteurs documents x_i et x_j, et elle est calculée comme suit :

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$$

6.1.2 Similarité Cosinus

La similarité cosinus est fréquemment utilisée dans la mesure de comparer deux documents textuels. Il s'agit d'une mesure qui calcule le cosinus de l'angle entre deux vecteurs documents x_i et x_j. La formule est définie par :

$$\cos(x_i, x_j) = \frac{< x_i, x_j >}{\|x_i\| \times \|x_j\|}$$

Où $< x_i, x_j >$ représente le produit scalaire des vecteurs x_i et x_j, $\|x_i\|$ et $\|x_j\|$ représentent respectivement les normes de x_i et x_j.

Par conséquent, la valeur du cosinus est de 1 lorsque les documents sont identiques et de 0 lorsqu'ils sont bien séparés et n'ont rien en commun. En pratique le cosinus est intéressant car plus l'angle est petit, plus son cosinus est proche de 1. Or, un petit angle signifie que les deux documents ont des proportions similaires.

6.1.3 Indice de Jaccard

L'indice de Jaccard ou coefficient de Jaccard est le rapport entre la cardinalité (la taille) de l'intersection des ensembles considérés et la cardinalité de leur union. Il permet d'évaluer la similarité entre les documents textes. Les documents d_1 et d_2 sont représentés, non pas comme des vecteurs, mais comme des ensembles de termes. Cette mesure est plutôt utilisée dans le cas de valeurs booléennes, sous la forme simplifiée suivante :

$$d_{jaccard}(d_1, d_2) = \frac{card(d_1 \cap d_2)}{card(d_1 \cup d_2)}$$

Par conséquent, la similarité obtenue par l'indice de Jaccard $d_{jaccard}(d_1, d_2)$ est une valeur du segment $[0,1]$.

6.2 Similarité sémantique

La similarité sémantique est un concept selon lequel un ensemble de documents ou de termes se voient attribuer une métrique basée sur la ressemblance de leur signification ou contenu sémantique. Il est très utile de prendre en compte des informations lexicaux ou sémantiques sur les unités textuelles pour avoir des profils plus signifiants. L'analyse sémantique s'intéresse au sens des phrases considérées par la signification des mots (unités) plutôt qu'à leur rôle syntaxique.

Dans la littérature, plusieurs travaux sur la mesure de similarité sémantique ont été développés dans différents contextes. De tels systèmes, basés sur l'analyse sémantique, sont préoccupés d'associer une sémantique au contenu.

Une étude comparative de cinq distances sémantiques [Budanitsky et Hirst, 2001], a permis d'obtenir des meilleurs résultats avec la mesure présentée dans [Jiang et Conrath, 1997] qui se focalise sur la probabilité d'apparition d'une classe. Le travail de [Zweigenbaum, 1997] a abordé les différents moyens de représentation sémantique par des graphes tout en pondérant les relations sémantiques reliant les unités mots. L'analyse sémantique a permis l'extraction des informations à partir des bases de données textuelles [Jacquemin et Zweigenbaum, 2000], [Gabrilovich et Markovitch, 2007]. Une distinction opérationnelle du contenu de texte à des catégories d'informations comme les objectifs, les méthodes, les résultats et les conclusions a été mis dans [Ruch et al., 2003].

Par ailleurs, la question de la distance sémantique a été développée de manière globale, en utilisant la notion de voisinage sémantique d'une entité de cluster [Rodriguez & Egenhofer, 2003]. Plusieurs mesures ont été présentées dans [Jin et Mobasher, 2003], en particulier une distance basée sur la corrélation entre les classes.

L'idée générale de la similarité consiste à déterminer la sémantique d'un mot en consultant les autres mots apparait à ses côtés au sein d'une même phrase. Ce sont les cooccurrences qui deviennent des éléments pertinents pour la détermination du sens [Lewis et Croft, 1990]. Une manière de le faire est d'utiliser des vecteurs pour représenter le sens des mots, et d'utiliser ensuite des mesures de similarité vectorielles. Il faut construire un ensemble de vecteurs pour chaque mot dans le dictionnaire utilisé. Les vecteurs sont définis dans un espace vectoriel orthogonal à n dimensions où chaque base se voit attribuer un mot de vocabulaire unique (chaque entrée du dictionnaire à une base dans l'espace vectoriel). Pour chaque mot du dictionnaire, on détermine un vecteur dans cet espace, où la composante du vecteur pour chaque

base est le nombre d'occurrences du mot dans la base qui le représente où il apparaît dans le contexte du mot pour lequel un vecteur a été construit.

7 Conclusion

Dans ce chapitre, nous avons présenté quelques définitions et concepts de base de la fouille de textes. Nous avons mis l'accent sur le prétraitement des textes utilisé ainsi que les pondérations des mots. Puis, des méthodes de réduction de la dimension après la phase de quantification ont été présentées. Nous avons ensuite analysé certaines représentations de textes permettant aussi l'amélioration de la réduction de la dimension et le traitement des données. Enfin, nous avons traité et présenté des propriétés générales de la similarité entre documents textuels et quelques mesures permettant de calculer quantitativement ces similarités.

La tâche de classification et de représentation des données textuels présentent un cadre commun largement partagé dans le domaine de la fouille de textes. Les méthodes employées pour réaliser ces tâches sont très diverses, et trouvent leurs fondements dans plusieurs disciplines comme l'intelligence artificielle, les statistiques ou encore l'optimisation.

CHAPITRE II : APPRENTISSAGE NON SUPERVISE

1 Introduction

L'apprentissage non supervisé (Unsupervised Learning) est un processus d'apprentissage automatique où les objets sont regroupés selon des caractéristiques similaires sans connaître les étiquettes des données. Contrairement à l'apprentissage supervisé (Supervised Learning) qui tente de trouver un modèle depuis des données étiquetées.

Les méthodes de classification automatique non supervisé (aussi appelées méthodes de clustering) sont basées sur la notion d'apprentissage non supervisé. Ces méthodes ont pour but de regrouper des individus en classes homogènes en fonction de l'analyse des caractéristiques qui décrivent ces individus. La mise en cluster consiste à séparer ou à diviser un ensemble de données en un certain nombre de groupes, de sorte que les ensembles de données appartenant aux mêmes groupes se ressemblent davantage que ceux des autres groupes. Les problèmes de classification automatique ont été traités à travers plusieurs ouvrages [Larose, 2014], [Wu, 2012], [Han et al., 2012]. De nombreuses méthodes de clustering sont disponibles, parmi lesquels deux groupes principaux : méthodes hiérarchiques et les méthodes de partitionnement.

Les méthodes de classification automatique ont apporté une aide précieuse, notamment par leurs applications en exploitant les informations et les données de plusieurs domaines pour la recherche d'information, la documentation ou l'aide à la décision. La classification non supervisée est utilisée dans différents domaines d'application on cite par exemple : la médecine, le traitement d'images [Lefèvre, 2002], l'éducation, text mining [Yin et al., 2012], [Bovo et al., 2013], etc.

L'expression non supervisée fait référence au fait qu'aucun superviseur ou label est utilisé pour savoir à quelle classe appartient un objet. En conséquence, le nombre de classes existant dans un ensemble des objets est à priori inconnu. De ce fait, l'un des problèmes les plus difficiles à propos des méthodes de classification non supervisée concerne le choix du nombre de classes à retenir. Pour pallier cet écueil, il existe des artifices permettant d'approcher le bon nombre de classes, ceux-ci seront discutés par la suite. Une fois le nombre de classes choisi, le prochain pas dans le processus de classification consiste à choisir l'algorithme de clustering adéquat

selon le type, la taille et d'autres critères, le dernier pas c'est évaluer la qualité de la partition obtenue par la méthode de clustering choisie.

Dans ce chapitre, nous présentons les méthodes de clustering les plus connus en détails, parmi lesquels deux groupes principaux : les méthodes hiérarchiques et les méthodes de partitionnement. Nous rappelons dans la section 3, les étapes du clustering à savoir le prétraitement des données, le choix de l'algorithme puis les critères d'évaluation des classes obtenus. La section 4 présente un tour d'horizon sur les indices de validité du clustering ainsi que ses performances. Au dernier paragraphe, nous mettons l'accent sur le problème le plus difficile dans la classification non supervisée concernant la détermination du nombre de classes à retenir pour une base de données.

2 Méthodes de clustering

Deux catégories principales de regroupement, connues sous le nom de regroupement par partitionnement et de regroupement hiérarchique, sont envisagées dans la littérature. La taxonomie des différents algorithmes de regroupement est représentée dans la figure (**II.1**).

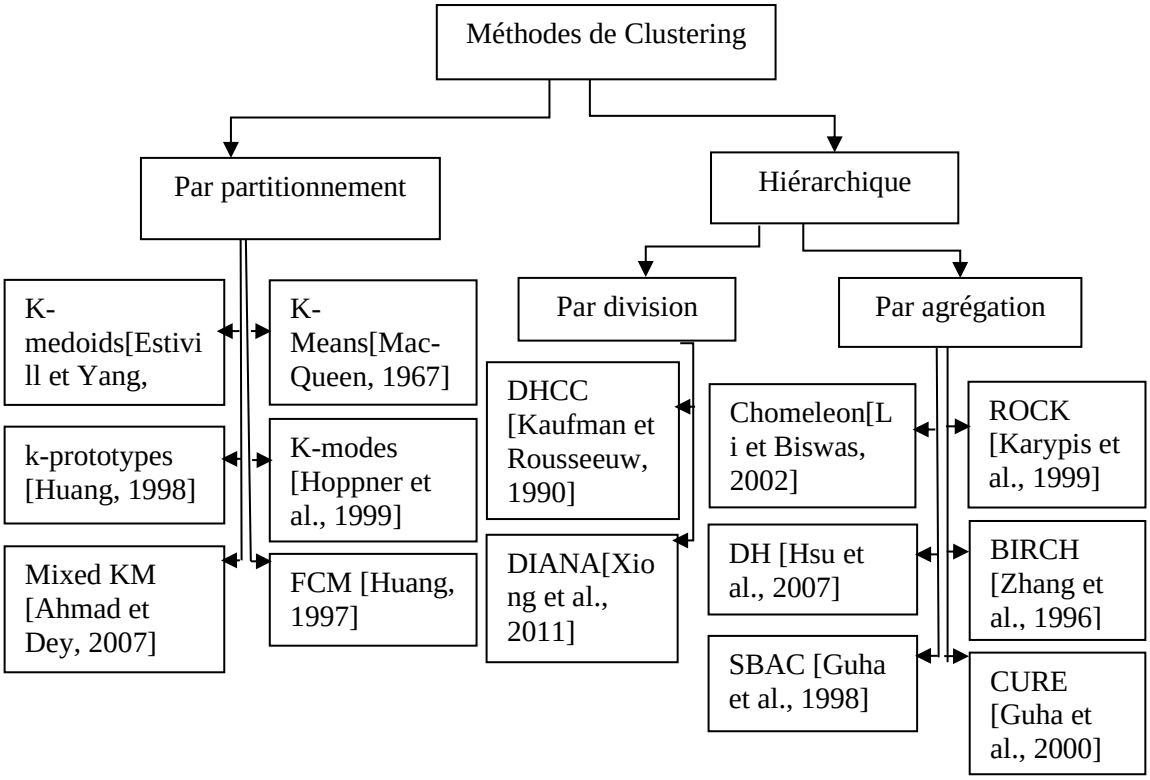


Figure II.1. **Taxonomie de méthodes de clustering**

2.1 Regroupement hiérarchique

La classification hiérarchique est une famille de techniques qui génèrent des suites de clusters emboîtés les uns dans les autres. Ces méthodes de regroupement hiérarchique visent à afficher la hiérarchie des échantillons de données à l'aide d'un dendrogramme, ce qui permet de créer une structure de cluster arborescente. Les techniques de classification existantes dans cette famille peuvent être classées dans deux grands groupes : Classification Ascendante Hiérarchique (CAH) ou par agrégation et Classification Descendante Hiérarchique (CDH) ou par division.

L'algorithme de clustering hiérarchique agglomératif génère un groupe imbriqué de clusters organisés sous forme d'arborescence. Ensuite, les deux clusters les plus proches sont agrégés dans un nouveau cluster combiné. Finalement, tous les enregistrements sont combinés en un seul énorme cluster. Les méthodes de clustering par division commencent avec toutes les données dans un grand cluster, les données les plus dissemblables étant scindées de manière récursive, en un cluster séparé, jusqu'à ce que chaque donnée représente son propre cluster [Karol et Mangat, 2013], [Kothari et Pitts, 1999].

Cependant, une question fondamentale dans le regroupement hiérarchique est : comment mesurer la similarité entre deux groupes ? Un certain nombre de méthodes courantes d'agrégation existantes pour chaque couple de clusters, nous citons ci-après [Aggarwal et Reddy, 2013] :

- Agrégation selon le lien minimum (single linkage) : définit la distance entre deux clusters A et B par le minimum des distances entre un élément du cluster A et un élément du cluster B.

$$d(A, B) = \min_{x \in A, y \in B}\{d(x, y)\}$$

- Agrégation selon le lien maximum (complete linkage) : définit la distance entre deux clusters A et B par le maximum des distances entre un élément du cluster A et un élément du clusterB.

$$d(A, B) = \max_{x \in A, y \in B}\{d(x, y)\}$$

- Agrégation selon le lien moyen (average linkage) : définit la distance entre deux clusters A et B par la moyenne des distances entre un élément du cluster A et un élément du cluster B.

$$d(A,B) = moy_{x \in A, y \in B}\{d(x,y)\}$$

- La méthode de Ward : on choisira de regrouper les deux clusters ayant la plus petite distance entre les classes (Inertie intra-classe minimum). Cette méthode tend à produire des classes plus compactes.

$$d(A,B) = \frac{n_1 \times n_2}{n_1 + n_2} d(G_1, G_2)$$

Avec $n_1$et n_2 les effectifs des deux classes A et B respectivement, G_1et G_2 leurs centres de gravité respectifs.

2.2 Regroupement par partitionnement

Dans le regroupement par partitionnement, un ensemble de données est décomposé en un ensemble de clusters disjoints. Un cluster est un ensemble d'objets tel qu'un objet dans un cluster est plus proche (plus similaire) du « centre » d'un cluster, que du centre de tout autre cluster. Chaque partition est un ensemble d'objets représenté par le centre ou le prototype de la classe, ce dernier est formé de telle manière qu'il est étroitement lié (en termes de similarité) à tous les objets de ce groupe. Les méthodes de partitionnement les plus connues comprennent K-Means et ses variantes [Ng, 2012], l'algorithme K-modes qui définit une classe par un représentant le plus central [Hastie et al., 2001], l'algorithme Possibilistic c-means est connu sous le nom de PCM, est également approprié pour démêler les clusters compacts. Le cadre ici est similaire à celui utilisé dans Fuzzy c-means (FCM) [Siddique et al.,2018]. Il y a d'autres algorithmes de partitionnement similaires au K-Means comme par exemple PAM, CLARA [17F] et CLARANS [Ng et Han, 2002].

2.2.1 K-Means

L'idée de l'algorithme K-Means est de classer un ensemble donné d'observations en un nombre k de classes disjointes, où la valeur k est fixée à l'avance. L'algorithme comporte deux phases. La première phase consiste à déterminer k centres de classes, une pour chaque classe. La deuxième phase consiste à prendre observation de l'ensemble de données et de l'affecter à la classe dont le centre de classe est le plus proche. La distance Euclidienne est généralement utilisée pour le calcul des distances deux à deux entre les observations et les centres des classes. Lorsque tous les points sont classés, la première étape est terminée et un premier regroupement est effectué. Il est alors nécessaire de recalculer les nouveaux centres de chaque classe car l'insertion de nouvelles observations au sein des classes peut entrainer un changement de

centres. Les nouveaux centres de classes correspondent aux centres de gravité de chacune des classes obtenues. Une fois que les k nouveaux centres sont déterminés, les distances entre les observations et ces centres sont calculées et certaines observations peuvent être affectées à une autre classe dont le centre est le plus proche. Ce processus est répété de manière itérative jusqu'à ce que les centres ne subissent plus de changement et soient stables. L'atteinte de cette stabilité correspond à un critère de convergence pour l'algorithme. Le pseudo-code de l'algorithme K-Means est présenté dans l'algorithme **(II.1)**.

Entrées:

D = $\{d_1, d_2, \dots, d_n\}$: un ensemble de n observations

k : le nombre de classes désirées

Sorties: A un ensemble de k classes

début

Choisir aléatoirement k centres de classes initiaux

répéter

- Affecter chaque observation d_i à la classe dont le centre de classe est le plus proche ;
- Calculer le nouveau centre de chaque classe ;

jusqu'à ce qu'un critère de convergence soit satisfait;

Algorithme II.1: K-Means

2.3 Regroupement basé sur les grilles :

L'approche de regroupement basée sur la grille consiste à diviser l'espace des nœuds en grilles. Ce type d'algorithmes diffère des autres algorithmes de regroupement par le fait de partitionner l'espace de valeurs qui entourent les points de données en différentes cellules à l'aide d'une grille. Ces valeurs statistiques accumulées permettent d'identifier les ensembles de cellules denses connectées puis lancent le clustering au niveau de cette région pour la formation des clusters. En général, un algorithme typique de regroupement basé sur une grille comprend les cinq étapes de base suivante [Shen-Yi et al., 2018] :

- Créer la structure de la grille, c'est-à-dire diviser l'espace de données en un nombre fini de cellules.
- Calculer la densité de chaque cellule.
- Trier les cellules en fonction de leur densité.
- Identifier les centres des classes.
- Traversez les cellules voisines.

La performance des méthodes appartenant à cette catégorie dépend beaucoup plus de la taille de la grille que de la taille du graphe lui-même en plus de la densité minimale des cellules de la grille. Cependant, les avantages les plus importants du clustering basé sur la grille sont la réduction significative de la complexité de calcul et le temps de traitement pour les grands ensembles de données, puisqu'il parcoure tout le graphe une seule fois pour calculer les valeurs statistiques des grilles, et sa tolérance aux valeurs aberrantes. Wave-Cluster [Jixue, 2009] et STING [Venkatkumar et Shardaben, 2016] sont des algorithmes typiques parmi d'autres de cette forme de clustering.

2.4 Regroupement basé sur les densités

Ce type d'algorithmes est basé sur les fonctions de connectivité et de densité. Ils sont assez proches du principe de plus proches voisins. Un cluster est défini comme un composant dense connecté qui peut croître dans n'importe quelle direction que mène la densité. La densité globale autour d'un nœud est étudiée pour déterminer des régions de forte densité, entourées par des régions de faible densité pour former les clusters. Certaines études intéressantes incluent DBSCAN [Tuinstra, 2016], OPTICS [Patwary et al., 2013] et DENCLUE [Idrissi et al., 2015] utilisent cette méthode pour filtrer le bruit et améliore la qualité de regroupement.

2.5 Regroupement basé sur les graphes

Ce genre d'algorithmes basé sur les graphes considère les clusters comme étant des ensembles de nœuds connectés dans un graphe. L'objectif est d'optimiser la somme des valeurs des arcs du graphe formé par la connexion des exemples entre eux. Par ailleurs, l'ajustement entre les nœuds du graphe combiné à un certain modèle mathématique prédéfini permet de déterminer le nombre de clusters. Parmi ces algorithmes on trouve EM [Ramadan et Tairi, 2015] qui utilise

un modèle basé sur la densité, ou le clustering conceptuel comme COBWEB [Satyanarayana et Acquaviva, 2014].

2.6 Cartes auto-organisatrices de Kohonen

Les cartes auto-organisatrices de Kohonen (ou Self Organizing Maps (SOM), en anglais) ont été introduites par T. Kohonen. Cette carte est un réseau de neurones artificiels d'apprentissage compétitif et non supervisé. Elle doit apprendre sans intervention externe, sans la connaissance des sorties désirées [Kohonen, 2000].

Ils sont généralement présentés comme une méthode de classification non supervisée qui généralise les méthodes du type partitionnement en introduisant la notion de voisinage entre les classes. Les classes sont représentées par des neurones disposés sur un réseau, appelé carte. Cette carte définit un voisinage a priori entre les classes en respectant une topologie préétablie. La tâche du réseau consiste à détecter la régularité et la corrélation dans les données [Kohonen, 1995], puis créer automatiquement des classes. L'apprentissage de la méthode aboutit à la concordance entre la proximité des neurones sur la carte et la proximité des individus dans l'espace des données : deux individus associés à des neurones voisins sont proches dans l'espace des données.

Des états de l'art plus détaillés sont disponibles dans la littérature. Le lecteur souhaitant une description plus avancée des méthodes pourra s'y référer [Aggarwal et Reddy, 2013], [Rencher, 2003], [Jain et al., 1999].

3 Étapes du clustering

Plus formellement, dans les problèmes du clustering, les données $D=\{x_1, x_2, \dots, x_n\}$ sont composées de n observations sans étiquettes (ou classes), et chacune des données x_i est décrite par m attributs :

$$\forall i \in \{1, \dots, n\}, \qquad x_i = \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,3} \\ \cdot \\ \cdot \\ \cdot \\ x_{i,m} \end{pmatrix}$$

L'objectif du clustering est de déterminer k clusters tel que chaque cluster regroupe des observations similaires ou les moins dispersées.

Les étapes principales du processus de classification non supervisée sont : la phase de préparation des données, puis la phase du choix de l'algorithme de clustering et à la fin la phase d'évaluation de la qualité des résultats obtenus.

3.1 Préparation des données

La phase de préparation des données est une phase primordiale dans la classification non supervisée [Celebi et al., 2013]. Les données brutes sont souvent bruitées, incohérentes et non structurées. Par conséquent, le prétraitement permet de transférer ces données dans un environnement plus facile à exploiter. Le processus de prétraitement des données comprend les phases de sélection, de nettoyage, d'intégration de transformation de réduction et de discrétisation. En général, les résultats de la fouille dépendent de la qualité de préparation des données.

3.2 Choix de l'algorithme

Le choix de l'algorithme de clustering se fait selon la nature des variables quantitatives ou qualitatives dans la base de données, et des clusters attendus : nombres, formes, densités, etc. Les critères principaux du choix de l'algorithme peuvent être : le volume de données à traiter, le temps de calcul nécessaire à la résolution du problème, la nature des données, la forme des clusters souhaités ou encore le type de sortie attendue, soit une hiérarchie de clusters, soit partition stricte, ou dendrogramme, etc.

3.3 Évaluation des clusters

Dans les méthodes de classification automatique, les chercheurs définissent généralement trois types de critères de validation selon que l'on dispose ou pas d'information à priori sur les données. Les critères d'évaluation des clusters sont de trois types : interne, externe et relatif [Theodoridis et Koutroubas, 1999].

Critères externes : nous évaluons les résultats d'un algorithme de classification en se basant sur une structure prédéfinie (connaissances externes sur les données).

Critères internes permettent de comparer différents ensembles de clusters sans aucune référence aux connaissances externes. Ces mesures internes varient d'un problème à l'autre. Elles peuvent en gros être subdivisées en deux groupes : celui qui évalue l'adéquation entre les données et la

structure attendue et les autres qui se concentrent sur la stabilité de la solution [Pacual et al., 2010].

Critères relatifs : se basent sur les deux critères de dispersions intra-clusters et inter-clusters. L'évaluation relative consiste à évaluer les résultats en comparant plusieurs schémas obtenus par une même méthode avec différents paramétrages.

Par la suite, la description des indices de ces critères sera discutée en détail dans la section suivante.

4 Techniques d'évaluation du clustering

Une évaluation de regroupement exige une mesure indépendante et fiable pour l'évaluation et la comparaison des expériences et des résultats de regroupement. La base commune des indices d'évaluations ainsi que leurs calculs sont tous basés sur un tableau (ou matrice) de contingence qui définit l'association entre deux classifications sur un même ensemble d'individus E [Aliguliyev, 2009].

Tableau II.1 - **Tableau de contingence entre la partition P et C.**

P/C	C_1	$\cdots$	C_j	$\cdots$	C_k	$\sum$
P_1	n_{11}	$\cdots$	n_{1j}	$\cdots$	n_{1k}	$n_{1.}$
$\vdots$						$\vdots$
P_i	n_{i1}	$\cdots$	n_{ij}	$\cdots$	n_{ik}	$n_{i.}$
$\vdots$						$\vdots$
P_m	n_{m1}	$\cdots$	n_{mj}	$\cdots$	n_{mk}	$n_{m.}$
$\sum$	$n_{.1}$	$\cdots$	$n_{.j}$	$\cdots$	$n_{.k}$	$n_{..} = n$

Considérons P et C les partitions résultantes de deux classifications sur un même ensemble d'individus I. La partition P est considérée la partition de référence. Le tableau II.1 représente la matrice de contingence sur ces deux partitions. n_{ij} indique le nombre d'individus présents simultanément dans le groupe P_i de la partition P et le groupe C_j de la partition C. En additionnant les valeurs des lignes ou des colonnes, on obtient les valeurs marginales $n_{i.}$ et $n_{.j}$, qui se rapportent au nombre d'objets dans les partitions P_i et C_j respectivement.

4.1 Indices de validation interne

Dans cette sous-section, nous présentons certains indices de validation du clustering de type internes. Ces indices permettent d'évaluer la qualité et la performance des clusters obtenus lors de la classification non supervisée.

4.1.1 La F-mesure

L'indice de F-mesure combine la précision et le rappel pour calculer le score. Elle correspond à la moyenne harmonique entre Rappel et Précision. Le rappel est défini par le ratio entre le nombre des exemples en commun entre deux clusters de différentes partitions et le total des exemples dans le cluster de la partition P. La précision est définie par le ratio entre le nombre des exemples en commun entre deux clusters de différentes partitions et le total des exemples dans le cluster de la partition C. À partir de la matrice de contingence, les concepts de Rappel et de Précision peuvent être définis par :

$$\mathrm{Recall}(P_i, C_j) = \frac{n_{ij}}{n_{i.}}$$

$$\mathrm{Precision}(P_i, C_j) = \frac{n_{ij}}{n_{.j}}$$

La F-mesure traditionnelle est égale à la moyenne harmonique des valeurs de précision et de rappel :

$$F(P_i, C_j) = \frac{2 \times \mathrm{Recall}(P_i, C_j) \times \mathrm{Precision}(P_i, C_j)}{\mathrm{Recall}(P_i, C_j) + \mathrm{Precision}(P_i, C_j)}$$

Au niveau de la partition, la F-mesure est calculée à partir des moyennes des valeurs maximales figurant dans les colonnes du tableau de contingence. La F-mesure prend des valeurs dans le segment [0,1].

$$F(P, C) = \frac{1}{n} \sum_{i=1}^{m} n_{i.} \max_{1 \le j \le k} F(P_i, C_j)$$

La valeur de la F-mesure est élevée pour les deux partitions P et C, si les partitions P et C sont très similaires.

4.1.2 La pureté

La pureté est l'une des principales mesures de validation permettant de déterminer la qualité des clusters. Chaque cluster est attribué à la classe qui est la plus fréquente dans le cluster. Ensuite, la pureté de cette assignation est formellement calculée comme ci-dessous :

$$\text{Purity}(P, C) = \frac{1}{n}\sum_{i=1}^{m} \max_{1 \le j \le k}(n_{ij})$$

La pureté est de 1 si chaque document possède son propre groupe.

4.1.3 L'entropie

L'entropie est une mesure de la théorie de l'information. Elle permet de lever l'incertitude d'information externe aux étiquettes des classes. L'entropie de chaque classe C_j est calculée comme suit :

$$\forall j \in \{1, .., k\}, \qquad E_j = -\sum_{i=1}^{m} p_{ij} \log(p_{ij})$$

Où p_{ij} est la probabilité que chaque membre d'un groupe C_j appartient à la classe P_i.

L'entropie totale est calculée comme la somme des entropies de chaque cluster pondéré par la taille de chaque grappe $n_{.j}$. Ensuite, l'entropie totale est formellement calculée comme ci-dessous :

$$E = \sum_{j=1}^{k} \frac{n_{.j} \times E_j}{n}$$

L'entropie est de 0 si tous les clusters sont constitués d'objets n'ayant qu'une seule étiquette de classe, nous nous attendons à ce que chaque cluster ait une entropie faible pour maintenir la qualité du clustering.

4.1.4 L'information mutuelle normalisée

L'information mutuelle normalisée (en anglais appelé Normalized Mutuelle Information (NMI)) est une normalisation du score de l'information mutuelle (IM) afin d'échelonner les résultats entre 0 (aucune information mutuelle) et 1 (corrélation parfaite). L'information mutuelle nous indique la réduction de l'entropie des étiquettes de classe que nous obtenons si nous connaissons les étiquettes de classe.

$$\text{NMI} = \frac{2 \times I(P; C)}{E(P) + E(C)}$$

Avec:

$E(P)$: Entropie de la partition P.

$E(C)$: Entropie de la partition C.

$I(P; C)$: Indice de l'information mutuelle entre les partitions P et C.

4.2 Indices de compacité et de séparation

4.2.1 L'indice Fukuyama-Sugeno

L'indice de Fukuyama-Sugeno (FS) est un indice de validité des classes, il combine compacité et séparation entre les classes. Les valeurs minimales de cet indice proposent également une bonne partition. Selon [Fukuyama et Sugeno, 1989], il est défini comme suit :

$$V_{FS} = \sum_{j=1}^{n}\sum_{i=1}^{k} u_{ij}{}^{m}(\| v_i - x_j \|^2 - \| v_i - \bar{v} \|^2)$$

Où, $X = \{x_1, \dots, x_n\}$ est un ensemble de données, $V = (v_1, \dots, v_k)$ est une matrice des centres de clusters, u_{ij} est la valeur d'appartenance du cluster x_j dans le cluster i, et l'exposant de pondération $m \geq 1$ est une constante qui influence les valeurs des membres.

4.2.2 L'indice de Xie-Beni

Xie et Beni (1991) ont proposé une mesure de validité de cluster axée sur deux propriétés : la compacité et la séparation. Il s'agit d'un rapport entre la variation totale de la partition et les centres (U, V) et la séparation des vecteurs centres. Selon [Xie et Beni, 1991], il est défini comme suit :

$$V_{XB} = \frac{\sum_{j=1}^{n}\sum_{i=1}^{k} u_{ij}{}^{2} \| v_i - x_j \|^2}{n \times \min\limits_{1\leq i,j\leq k,\ i\neq j}(\| v_i - v_j \|^2)}$$

Les valeurs minimales de cet indice proposent également une bonne partition.

4.3 Indices d'homogénéité

4.3.1 L'indice de Davies-Bouldin

Cet indice vise à identifier des ensembles de classes qui sont compactes et bien séparées. L'indice de Davies-Bouldin est défini comme suit [Davies et Bouldin, 2000]:

$$DB = \frac{1}{c}\sum_{i=1}^{c} \max_{i\#j}\left\{\frac{d(X_i) + d(X_j)}{d(c_i, c_j)}\right\}$$

Où c indique le nombre de classes, i, j sont des étiquettes de classes, puis $d(X_i)$ et $d(X_j)$ sont des échantillons des classes i et j à leurs centres respectifs, $d(c_i, c_j)$ représente la distance entre les centres des deux classes c_i et c_j .

Cet indice est la moyenne du rapport maximal entre la distance d'un point au centre de son groupe et la distance entre deux centres de groupes. On constate ainsi que ce rapport sera d'autant plus faible que les classes sont compactes et éloignées les unes des autres. Une valeur plus petite de DB indique un "meilleur" partitionnement.

4.3.2 L'indice de Silhouette

L'indice de Silhouette travaille à l'échelle microscopique, c'est à dire qu'il s'intéresse aux documents en particulier et non pas aux classes. Le but de Silhouette est de vérifier si chaque document a été bien classé.

Pour cela, et pour chaque document i de la partition, la valeur de l'indice de Silhouette est calculée :

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Où $a(i)$ représente la distance moyenne qui le sépare des autres documents de la classe à laquelle il appartient et $b(i)$ représente la distance moyenne qui le sépare des documents appartenant à la classe la plus proche.

Quand $S(i)$ est proche de 1, le document est bien classé : la distance qui le sépare de la classe la plus proche est très supérieure à celle qui le sépare de sa classe. Par contre, si $S(i)$ est proche de -1, cela veut dire que le document est mal classé. Mais si $S(i)$ est proche de 0 alors il pourrait également être classé dans la classe la plus proche.

Par ailleurs, l'indice de Silhouette de la partition est calculé à partir de la moyenne entre les indices de ses éléments. Par conséquent, la meilleure partition retenue est alors celle qui permet d'obtenir un indice de silhouette global maximal.

5 Problème de détermination du nombre de classes

La plupart des algorithmes de classification non supervisée nécessitent un nombre de classes spécifié par l'utilisateur ou des paramètres implicites de contrôle du nombre de classes à l'avance. Pour certaines applications, le nombre de classes peut être estimé en termes d'expertise de l'utilisateur ou de connaissance du domaine. Cependant, dans de nombreuses situations, pour un ensemble de données le nombre de classes est inconnu à l'avance. Il est bien connu que la surestimation ou la sous-estimation du nombre de classes affectera considérablement la qualité des résultats de la mise en cluster. Par conséquent, l'identification du nombre de classes dans

un ensemble de données (une valeur souvent appelée K) est une question fondamentale dans l'analyse des classes.

Dans le cas de la Classification Ascendante Hiérarchique (CAH), le choix du K peut se faire a posteriori de la classification, à partir du dendrogramme obtenu, en le coupant à des niveaux différents, puis en comparant les classes obtenues par les différentes coupures. Dans le cas d'une classification avec K-Means ou k-modes, le nombre de classes est à fournir en paramètre de la méthode. Pour estimer la valeur de K, de nombreuses études ont été rapportées dans la littérature [Jose et Gomez, 2016], [Kuo et al., 2012], [Das et al., 2006], [Kuncheva et Bezdek, 1997].

La recherche du bon nombre de classes se fait par essais et erreurs. Une même technique est utilisée à plusieurs reprises avec différentes valeurs de K, et pour chaque nouvelle partition obtenue, on calcule la valeur d'un critère de qualité et celle qui l'optimise (maximise ou minimise) est considérée comme étant le nombre de classes optimal et la partition associée est considérée comme la meilleure. Parmi ces indices d'évaluation du clustering on trouve : l'indice Silhouette de Rousseeuw [Rousseeuw, 1987], l'indice de Davies et Bouldin [Davies et Bouldin, 2000], l'indice de Xie et Beni [Xie et Beni, 1991] et l'indice de Fukuyam-Sugeno [Fukuyama et Sugeno, 1989].

La méthode Elbow est l'une des méthodes les plus populaires pour déterminer cette valeur optimale de k. Elle consiste à effectuer une mise en cluster de l'ensemble de données avec un nombre croissant de classes, à calculer la somme des erreurs quadratiques pour chacune d'entre elles et à les représenter sur un graphique linéaire. Si le graphique ressemble à un bras, la meilleure valeur de k se trouvera sur le "coude" [Gove, 2017].

Il existe certains critères, dérivés de différentes approches, pour déterminer le nombre optimal de classes. Nous citons le critère de Xie et Beni [Xie et Beni, 1991], qui est basé sur une mesure de la séparabilité et de la compacité des classes. Ces deux notions définissent les critères d'évaluation d'une classification. Xie et Beni proposent de choisir le k optimal qui minimise la relation entre séparabilité et compacité.

Tong et ses collaborateurs [Tong et al., 2011] ont appliqué l'algorithme Greedy pour obtenir le nombre de centres et ont obtenu un résultat final de regroupement avec un taux de précision et une stabilité satisfaisante.

Subbalakshmi et ses collaborateurs [Subbalakshmi et al., 2015] ont présenté une méthode d'indice de silhouette floue sur des données dynamiques pour trouver le nombre optimal de

classes. Le nombre optimal de classes k est celui qui maximise la silhouette moyenne sur une gamme de valeurs possibles pour k, le résultat de la mise en cluster est instable.

Selon [Karol et Mangat, 2013], le C-Moyennes flou est généré pour produire les centres initiaux, et l'optimisation des essaims de particules (particle swarm optimization (PSO) en anglais) est une méta-heuristique d'optimisation utilisée pour faire des clusters optimaux.

Song et ses collaborateurs [Song et al., 2015] ont amélioré l'algorithme de Huang Min [Huang et al., 2011] pour sélectionner les centres de regroupement initiaux en se concentrant sur la distance entre les échantillons qui ont les mêmes paramètres de densité maximale et en la comparant avec la distance moyenne de l'ensemble de données.

Shen-Yi et ses collaborateurs [Shen-Yi et al., 2018] ont initialisé K-Means par les centres initiaux produits par l'algorithme de regroupement hiérarchique, les résultats de regroupement trouvés en termes de F-mesure ne sont pas très convaincants, et l'ensemble de données provient du corpus de textes chinois.

Sherkat et ses collaborateurs [Sherkat et Velcin, 2018] ont produit des centres initiaux basés sur une méthode appelée DSKM (Deterministic Seeding K-Means). L'idée clé de la méthode proposée est de sélectionner k points de données qui sont distants les uns des autres, et qui ont en même temps une norme L1 élevée. Ces points de données sont utilisés pour initialiser l'algorithme K-Means, la méthode a donné de bons résultats car elle prend comme paramètre (K) le nombre réel de classes, ce qui signifie qu'elle est sensible au nombre de classes comme paramètre.

6 Conclusion

Les méthodes de classification non supervisée ont montré de bons résultats et largement utilisées dans différents domaines de traitement de l'information. Parmi les applications qui utilisent le processus d'apprentissage non supervisé on trouve : le traitement d'images médicales, le traitement de l'information, la classification des documents textuels, etc…

Au cours de ce chapitre, nous avons présenté les différentes méthodes de classification non supervisée. Ensuite, nous avons défini le processus de classification non supervisée et les étapes principales de ce processus qui sont : la préparation des données, le choix de l'algorithme de clustering et évaluation de la qualité des résultats obtenus. Puis, nous avons présenté les indices de validation et de comparaison de partitions obtenues sur un même ensemble d'individus.

Finalement, nous avons abordé la problématique de détermination du nombre de classes à travers la présentation de quelques indices et stratégies permettant de trouver une solution approchée.

CHAPITRE III : MODELE NEURONAL DE L'ENSEMBLE STABLE

1 Introduction

Nombreux sont les problèmes pratiques qui se présentent sous forme d'un graphe. En particulier, lorsque des conflits entre certains objets se produisent. La théorie de graphes permet de définir dans son contexte des ensembles indépendants permettant de séparer ces conflits. Pour résoudre de nombreux problèmes de classification, on est amené à chercher des ensembles indépendants permettant de répartir les données. Ces derniers servent également de modèles utiles pour les problèmes d'optimisation du monde réel. Par exemple, un ensemble indépendant est un modèle utile pour découvrir des composants indépendants ou stables dans un réseau d'objets. Cela indique que le problème de l'ensemble indépendant (ou stable) est applicable, tel que les problèmes de partitionnement, la théorie de la classification, l'économie [Serraji et al., 2016], la programmation et l'ingénierie biomédicale [Butenko, 2003]. Ainsi que les problèmes sur les flots et le problème du voyageur de commerce [Bomze et al., 1999]. En plus, de l'intérêt théorique du problème de l'ensemble stable on le retrouve dans des applications de la recherche d'information, de l'analyse de la transmission des signaux et de la vision par ordinateur.

Ce chapitre est organisé comme suit : nous commençons par une brève présentation sur les notions principales et la terminologie de la théorie de graphes pour définir l'ensemble stable dans un graphe. Dans la section 3, nous réalisons un tour d'horizon sur les méthodes proposées dans la littérature pour résoudre le problème de l'ensemble stable de taille maximum, tout en distinguant entre les méthodes approchées et les méthodes exactes. Puis, nous terminons par une large étude sur les différentes formulations du problème de l'ensemble stable maximal en des programmes quadratiques. La section 4 présente différents attributs essentiels du réseau neuronal de Hopfield continu, sa conception et son fonctionnement et surtout en précisant le lien et la formulation des problèmes d'optimisations sous forme de la fonction d'énergie du réseau de Hopfield continu en donnant les poids et les seuils associés.

2 Ensemble stable dans un graphe

Avant de parler d'un ensemble stable dans un graphe, on va rappeler la définition d'un graphe en général. Ainsi que les caractéristiques de base concernant la théorie de graphe. De manière générale, un graphe permet de représenter la structure, les relations entre les éléments d'un

ensemble. Il permet de modéliser une grande variété de problèmes : réseau de communication, réseaux routiers…etc.

2.1 Définitions élémentaires

2.1.1 Notions de base

Définition d'un graphe :

Un graphe est un couple $G = (V, E)$ où V est un ensemble et E est un sous-ensemble de $V \times V$.

Définition des objets :

Les éléments de V du graphe G sont appelés les points ou les sommets ou les nœuds du graphe.

Définition des liens :

On appelle arc du graphe orienté G tout couple (i, j) de E.

L'arc (i, i) est appelé boucle.

On appelle arête du graphe non orienté G tout sous ensemble de deux points $\{i, j\}$ où i est différent de j et le couple $(i, j) \in E$.

i et j sont appelés les extrémités de l'arête.

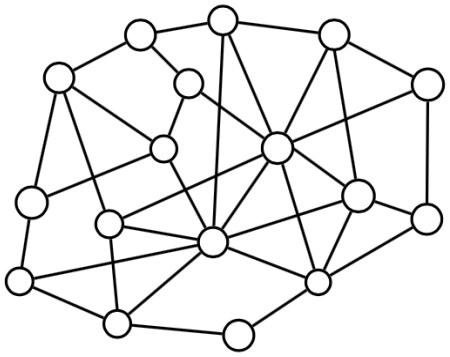


Figure III.1 **Exemple d'un graphe**

2.1.2 Terminologie

Deux sommets i et j sont **adjacents**, s'il existe une arête dont les points sont les deux extrémités.

- Graphe complet :

Un graphe complet est un graphe simple dont tous les sommets sont adjacents deux à deux.

- Sous-graphe :

$H = (Y, B)$ est un sous graphe de $G = (V, E)$ si $Y \subseteq V$ et $B \subseteq E$.

- Graphe partiel :

$H = (Y, B)$ est un graphe partiel de $G = (V, E)$ si $Y = V$ et $B \subseteq E$.

- Ordre d'un graphe :

L'ordre d'un graphe est le nombre de sommets de ce graphe : $|V|$.

- Nombre chromatique d'un graphe :

Nombre minimal de couleurs permettant de colorier les sommets d'un graphe, de telle sorte que deux sommets adjacents n'aient pas la même couleur.

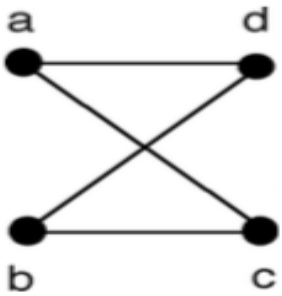
2.1.3 Matrice d'adjacence

On considère un graphe $G = (V, E)$ avec $V = \{v_1, \dots, v_n\}$. On appelle matrice d'adjacence du graphe G, la matrice $M \in \mathcal{M}_n(\mathbb{R})$ dont les coefficients m_{ij} sont définis par :

$$m_{ij} = \begin{cases} 1, & si\ (v_i, v_j) \in E \\ 0, & sinon \end{cases}$$

Cette matrice est bien symétrique à coefficients binaires.

La figure **(III.2)** représente un graphe G à quatre sommets $V = \{a, b, c, d\}$ reliant entre eux par 4 arêtes, ainsi que sa matrice d'adjacence M :



$$M = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Figure III.2 Graphe et sa matrice d'adjacence

2.2 Définitions et problèmes sur les stables

2.2.1 Définition d'un stable

Dans un graphe $G = (V, E)$, Un sous-ensemble S de points de V est un ensemble **stable** si ses points ne sont pas adjacents entre eux ou il n'existe aucune arête reliant ses points deux à deux.

Un sous-ensemble stable S est maximum si sa cardinalité est maximale.

2.2.2 Remarques

On s'intéresse aux stables lorsqu'on cherche des sous-ensembles d'objets deux à deux compatibles, sachant que le graphe G représente alors l'incompatibilité entre les objets. Ou bien vice versa.

S est un ensemble stable (également appelé ensemble indépendant) dans un graphe G si $S \subseteq V$ tel qu'aucun des deux sommets de S n'est relié par un bord, c'est-à-dire : $i, j \in S \Rightarrow \{i, j\} \notin E$.

3 Problème de l'ensemble stable maximum

Le problème de l'ensemble stable maximum (en anglais the maximum stable set problem sera noté (MSSP)) vise à trouver le plus grand ensemble stable dans un graphe G, la taille du stable maximum dans G est appelée le nombre de stabilité noté $\alpha(G)$.

La Figure **(III.3)** représente un exemple d'un ensemble stable dans un graphe G à 6 nœuds et 5 arêtes. Il est facile de voir que $S = \{1, 2, 5, 6\}$ est un stable maximum dans G de taille $\alpha(G) = 4$.

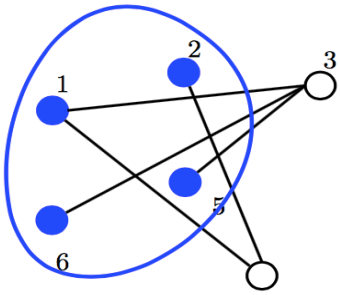


Figure III.3 **Ensemble stable maximum**

Dans la section suivante, nous présentons un aperçu sur l'état de l'art des méthodes approchées et des méthodes exactes du problème du stable maximum.

3.1 État de l'art

Dans les années soixante-dix, le premier algorithme d'énumération implicite pour le problème du stable maximum a été proposé par [Desler et Hakimi, 1970]. Cet algorithme est amélioré en d'autres versions avec des complexités inférieures.

A la fin des années quatre-vingts, une première heuristique de recherche tabou pour le problème du stable maximum a été proposé [Friden et al., 1989], sous le nom STABULUS. Elle est basé sur la stratégie de pénalisation en commençant avec un sous ensemble S non stable et en changeant un sommet de S avec un sommet de $V \setminus S$ jusqu'à que S devient stable.

Dans le même contexte des échanges des sommets entre les ensembles S et $V \setminus S$ dans un graphe. On trouve des heuristiques de recherche tabou [Ng, 2012], [Jin et Hao, 2015] qui ont prouvé le bon fonctionnement et l'efficacité sur des instances de référence pour la résolution du problème de l'ensemble stable de taille maximale.

Une liste plus détaillée des méthodes exactes pour résoudre le problème de l'ensemble stable maximum, peut être trouvée dans [Bomze et al., 1999]. La plupart de ces méthodes sont basées sur le principe général de l'algorithme de Branch and Bound (B&B) [Mannino et Sassano, 2005]. Dans la littérature, d'autres méthodes ont été discutées pour résoudre ce problème utilisant, par exemple, des formulations continues [Burer et al., 2002], la génération de colonnes [Bourjolly et al. 1999], la programmation par contraintes [Régin, 2003], [Verweij et Aardal, 1999] ou des résultats de calcul pour différentes relaxations de programmation linéaire ont été rapportés par [Gruber et Rendl, 2003].

Par ailleurs, le problème de la clique maximum est un problème qui consiste à trouver le maximum (en termes de cardinalité) d'un sous graphe complet dans un graphe G est équivalent au problème du stable maximum dans $\overline{G}$. Par conséquent, le problème de l'ensemble stable est équivalent au problème de la clique maximale dans le graphe du complément. Dans ce contexte, chaque méthode résolvant le problème de la clique maximale peut également être utilisée pour résoudre le problème de l'ensemble stable maximum [Bonomo et al., 2005], [Cogisa et Thierry, 2005], [Lehmann et al., 2006].

3.2 Modélisation et complexité

Dans la littérature, on trouve plusieurs travaux de la modélisation du problème de l'ensemble stable maximum. Dans cette sous-section, nous présentons les formulations mathématiques les

plus utilisées dans la modélisation du problème de l'ensemble stable maximum, pour une étude profonde et détaillée sur ces modélisations [Butenko, 2003]. Il est bien connu qu'il est NP-difficile de déterminer un ensemble stable maximum dans un graphe arbitraire [Hàstad, 1999]. Il est également difficile d'approcher le nombre d'ensembles stables par un algorithme temporel polynomial [Arora et Safra, 1992], [Fujisawa et al., 1995].

La formulation la plus générale du problème de l'ensemble stable de poids maximum, est exprimée par le modèle mathématique suivant [Rebennack, 2008] :

$$\begin{cases} \text{Max} \quad c^{T}x \\ \quad s.c \\ \qquad x_i + x_j \leq 1, \quad \forall (i,j) \in E \\ \qquad x \in \{0,1\}^{|V|} \end{cases}$$

Les variables binaires x_i prennent la valeur 1, si le nœud v_i se trouve dans un ensemble stable, par exemple S, et zéro sinon. Ainsi, le vecteur c désigne le vecteur de poids (positif) des nœuds.

Les contraintes $x_i + x_j \leq 1$ sont appelées les inégalités du bord et garantit, pour chaque bord$\{i,j\}$ un seul nœud final peut se trouver dans l'ensemble stable S. La formulation est assez compacte. Malheureusement, les contraintes binaires sur x rendent difficile la résolution de ce programme linéaire.

Une autre formulation qui élimine les poids du nœuds dans la formulation précédente, i.e. le vecteur $c = (1, \dots, 1)^{T}$, définie alors un problème de l'ensemble stable maximum. Ce problème peut être formulé sous forme d'un programme linéaire à contraintes quadratiques et variables binaires. La relaxation semi-définie positive (SDP) a été proposée pour la première fois par Lovász dans [Lovász, 1979] pour résoudre ce problème. Elle nous permet d'obtenir une limite supérieure à la solution en résolvant un programme semi-défini.

Le problème de l'ensemble stable peut être formulé comme suit :

$$\begin{cases} \qquad \text{Max} \quad \sum_{i=1}^{n} x_i \\ s.c \\ \quad {x_i}^2 = x_i, \quad i \in V = \{1, \dots, n\} \\ \quad x_i x_j = 0, \quad (i,j) \in E \end{cases}$$

La contrainte ${x_i}^2 = x_i$ équivaut à dire que $x_i \in \{0,1\}$ et l'ensemble stable S correspond à l'ensemble des i tel que $x_i = 1$. Notez que la contrainte $x_i x_j = 0$ garantit que S est un ensemble

stable. La fonction objective $\sum_{i=1}^{n} x_i$ compte la cardinalité de S. La résolution du problème d'optimisation est en général difficile à calculer.

Un autre angle de vue, pour le problème de l'ensemble stable maximal est exprimé comme un problème d'affectation linéaire avec une contrainte quadratique à variable binaire [Ettaouil et al., 2010] :

$$\begin{cases} \text{Min} \ -\sum_{i=1}^{n} x_i \\ s.c \\ x^t B x = 0 \\ x \in \{0,1\}^n \end{cases}$$

Avec la matrice B (matrice d'adjacence) a pour coefficients bivalents représentant la connexion entre les sommets du graphe :

$$b_{ij} = \begin{cases} 1, & si\ (i,j) \in E \\ 0, & sinon \end{cases}$$

Pour résoudre ce modèle, de nombreuses méthodes différentes sont testées et éprouvées, telles que la méthode du point intérieur, les relaxations semi-définies positives (SDP) [Ettaouil et Loqman, 2008] et les relaxations lagrangiennes [Thiongane et al., 2005].

Dans [Ettaouil et al., 2010], une nouvelle approche a été proposée pour résoudre le problème du stable maximum en utilisant un réseau de neurones artificiels appelé réseau de Hopfield continu (CHN). La méthode proposée consiste à résoudre le modèle ci-dessous via le réseau de Hopfield continu qui donne un minimum local. Puis améliorer la solution initiale en ajoutant une contrainte linéaire pour obtenir, une bonne solution au problème du stable maximum.

4 Réseau de Hopfield

Hopfield a proposé une réalisation physique possible d'un réseau de neurones. Ce modèle neuronal est inspiré des systèmes physiques comme le modèle des verres de spin d'Ising à base d'amplificateurs opérationnels [Hopfield, 1982]. Ce réseau neuronal est dit réseau de Hopfield continu en anglais Continuous Hopfield Network (CHN). Le mode d'apprentissage utilisé ici est le mode non-supervisé [Hopfield et Tank, 1985]. Cette découverte de Hopfield a permis de relancer l'intérêt et les recherches sur les réseaux de neurones artificiels [Cichocki et Unbehauen, 1993], [Wena et al.,2009], [Smith, 1999], puisqu'il permet de résoudre une grande variété des problèmes d'optimisations combinatoires et de fonctionner comme une mémoire associative non-linéaire [Joya et al., 2002], [Hopfield, 1984], [Hopfield, 1982], [Hopfield et Tank, 1985].

Dans cette section, nous présentons la conception générale du réseau de Hopfield, son architecture, et notamment le lien entre l'équation d'évolution du réseau et sa fonction d'énergie. Ensuite, nous donnons l'application essentielle dans le domaine de l'optimisation combinatoire, en faisant associer le problème de la programmation quadratique en termes de minimisation de la fonction d'énergie associée au réseau de Hopfield continu.

4.1 Conception des réseaux de Hopfield

Le modèle de base des réseaux de Hopfield électronique est construit à partir de l'interconnexion d'un ensemble de résistances, des amplificateurs non linéaires avec sortie symétrique, et d'un courant provenant de l'extérieur (Figure **III.4**).

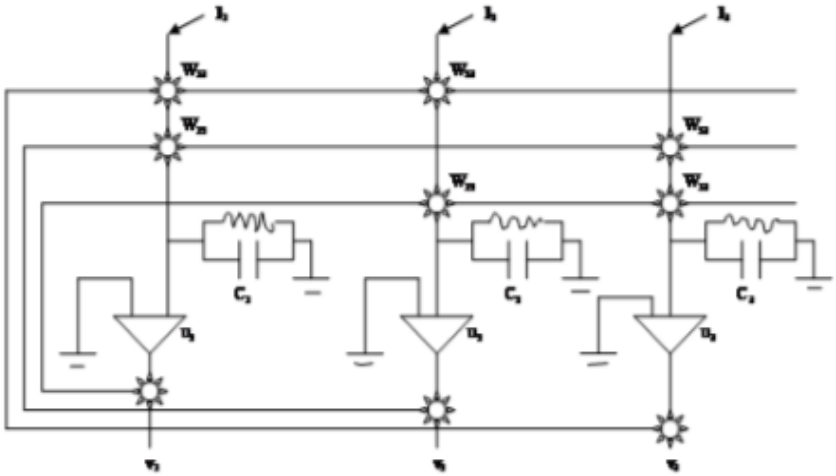


Figure III.4. **Schéma électronique du réseau de Hopfield**

Ainsi, le système des équations différentielles qui régi l'évolution dans le temps du réseau de Hopfield continu, est déterminé par l'application de la loi de conservation de Kirchhoff (KCL) à l'entrée de chaque amplificateur i :

$$C_i \frac{du_i}{dt} + \frac{u_i}{R_i} = \sum_{j=1}^{n} W_{ij} x_j + I_i \qquad \text{(III. 1)}$$

Avec :

W_{ij} est la conductance entre les amplificateurs i et j.

R_i, C_i sont respectivement la résistance et la capacité à l'entrée de l'amplificateur i.

I_i est un bruit provenant de l'extérieur.

u_i et x_i sont respectivement le voltage à l'entrée et à la sortie de l'amplificateur i.

Ces deux derniers voltages u et x à chaque amplificateur, sont liés par la fonction d'activation en tangente hyperbolique définie par :

$$\forall i \in \{1, \dots, n\}, \qquad x_i = g(u_i) = \frac{1}{2}\left(1 + \tanh\left(\frac{u_i}{u_0}\right)\right) \qquad \text{(III. 2)}$$

Avec $u_0 > 0$ est un paramètre utilisé pour contrôler le gain (ou la pente) de la fonction d'activation.

La figure **(III.5)** représente le graphe de la fonction tangente hyperbolique. Par conséquent, on observe que la fonction d'activation g est bien une fonction continue à seuil dans le segment [0,1]. En plus, cette fonction [Dreyfus et al., 2004], permet d'assurer une convergence rapide vers les sommets de l'hypercube $[0,1]^n$:

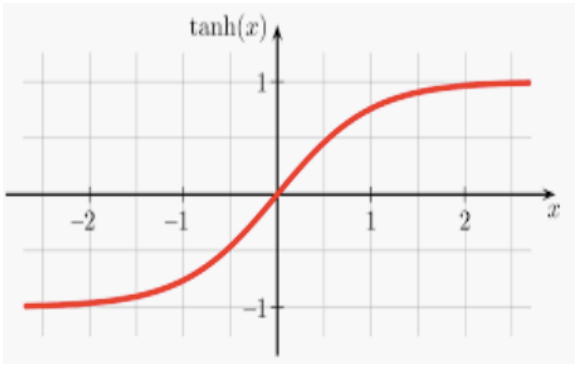


Figure III.5. **Graphe de la fonction tangente hyperbolique**

4.2 Architecture neuronale des réseaux de Hopfield

En supposant dans tout le schéma électronique du réseau de Hopfield (Figure **III.4**), que la conductance et la capacité à l'entrée de chaque amplificateur (neurone) i sont les mêmes. Le système d'équation différentielle (Eq. **III.1**) qui gouverne l'évolution du réseau de Hopfield continu devient :

$$\frac{du}{dt} = -\frac{u}{\tau} + Tx + I^b \qquad \text{(III. 3)}$$

Ce système d'équation non linéaire (Eq. **III.3**) est appelé l'équation d'état du réseau neuronal de Hopfield continu.

Les réseaux de Hopfield sont des réseaux récurrents totalement connectés. Chaque neurone i est connecté à toutes les autres neurones j par des poids synaptiques et symétriques tels que $T_{ij} = T_{ji}$, sauf à lui-même $T_{ii} = 0$. En plus il n'y a aucune différenciation entre les neurones d'entrée et ceux de sortie (Figure **III.6**).

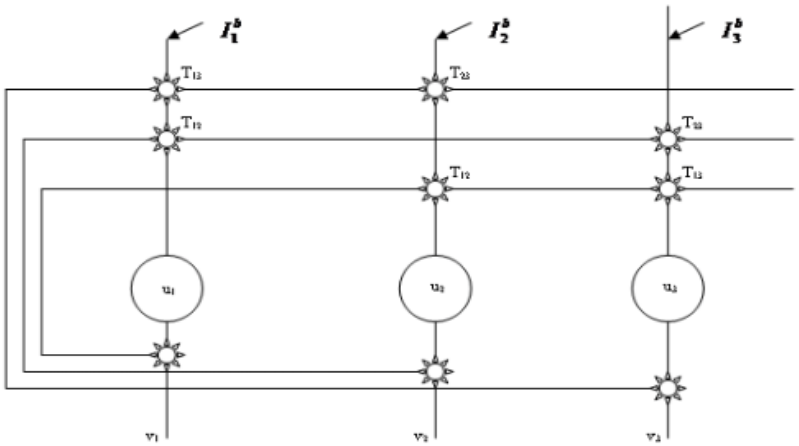


Figure III.6 **Réseau de Hopfield continu à 3 neurones**

La sortie de chaque neurone est rebouclée sur les entrées des autres neurones. Le potentiel intérieur ou l'état interne de chaque neurone u_i est calculé comme la somme des informations extérieurs x_j provenant des autres neurones pondérés par des poids T_{ij}, cette somme est perturbée par un bruit I_i^b provenant de l'extérieur.

Les états interne et externe de chaque neurone sont liés par la fonction d'activation hyperbolique (Eq.**III.2**) qui se comporte comme une fonction à seuil [Fels, 1994]. Les unités du réseau sont des automates à seuils. Celles-ci révisent leurs activations dans un ordre aléatoire jusqu'à ce que le réseau atteint un état d'équilibre stable. Cet état stable est la réponse ou la sortie statique du système à une donnée initiale en entrée [Hopfield, 1987].

4.3 Fonction d'énergie pour l'optimisation combinatoire

Hopfield a montré dans [Hopfield, 1984], via la symétrie des poids et par un argument de fonction d'énergie de Lyapunov, que la dynamique d'état de son réseau (Eq. **III.3**) est asymptotiquement stable. La méthode consiste à générer une fonction réelle de type énergétique de tout le réseau qui se décroît avec le temps, et qui garantit la convergence vers des états stables.

Cette fonction d'énergie est donnée par :

$$E(x) = -\frac{1}{2}x^{t}Tx - (I_b)^{t}x + \frac{1}{\tau}\sum_{i=1}^{n}\int_{0}^{x_i} g^{-1}(y)dy$$

Où g est la fonction d'activation des neurones (Eq. **III.2**).

En général, la pente τ (valeur du temps pour les amplificateurs) à l'origine des neurones est assez grande. Par conséquent, la fonction d'énergie associée au réseau de Hopfield continu devienne une fonction quadratique :

$$E(x) = -\frac{1}{2}x^{t}Tx - (I^{b})^{t}x \qquad (III.4)$$

Sous forme algébrique cet énergie s'écrit comme :

$$E(x) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} T_{ij}x_i x_j - \sum_{i=1}^{N} I_i{}^{b}x_i \qquad (III.4)$$

Les réseaux de Hopfield peuvent être utilisés pour résoudre approximativement des problèmes assez difficiles d'optimisation. En général, une telle fonction d'énergie (Eq. **III.4**) s'écrit sous la forme d'un coût et d'un terme qui exprime les contraintes d'un problème d'optimisation [Aiyer et al., 1990], [Dreyfus et al., 2008]. La méthode consiste à construire par pénalisation des contraintes du problème d'optimisation, un réseau dont les poids de connexions et les bruits sont les coefficients de la fonction d'énergie quadratique (Eq. **III.4**).

Le réseau de Hopfiled continu est un réseau à minimisation d'énergie, où la solution correspond à l'état stable atteint par le réseau. L'évolution de l'état global du réseau de Hopfield est un déplacement dans l'espace des états $\{0,1\}^{n}$ à la recherche d'un minimum (qui peut être local) ou d'un état stable atteint par le réseau.

Par ailleurs, un réseau de Hopfield continu est plus rapide et réel en comparaison avec un réseau de Hopfield discret (binaire) [Talavàn et Yànez, 2005]. Ce comportement de convergence ainsi que les techniques de résolution de problèmes d'optimisation quadratiques ont été exploitées pour de nombreuses applications [Ham et Kostanic, 2001], [Shih et al., 2004], [Smith, 1999].

L'algorithme suivant détermine un point d'équilibre du réseau de Hopfield continu CHN [Talavàn et Yànez, 2005] :

Fonction CHN(n, T, I^b, u_0, u^0, ε, q, R_ITER :) :

Entre : n, T, I^b, ε, u^0, u_0, q, R_ITER.

Sortie : **Point d'quilibre de CHN**

Initialisation : $x_i^0 := \frac{1}{2}(1+\tanh(\frac{u_i^0}{u_0}))$, $u^\varepsilon := \frac{u_0}{2}\ln(\frac{\varepsilon}{1-\varepsilon})$, $t := 0$, $iter := 0$

Répéter

$\Delta t = 10^{100}$

Pour $(i = 0; i \leq n; i++)$ **faire**

$\frac{du_i(t)}{dt} = \sum_{j=1}^{n} T_{ij}x_j(t) + I_i^b$

Si $(x_i(t) \in \{0,1\})$ **Alors**

Si $((x_i(t) = 0$ **ET** $\frac{du_i(t)}{dt} > 0)$ **OU** $(x_i(t) = 1$ **ET** $\frac{du_i(t)}{dt} < 0))$ **Alors**

$\Delta t = Min\{\Delta t, \frac{|u_i(t)|-u^\varepsilon}{\frac{du_i(t)}{dt}}\}$;

Fin Si

Sinon

$\frac{dx_i(t)}{dt} = \frac{2}{u_0}x_i(t)(1-x_i(t))\frac{du_i(t)}{dt}$;

Si $(\frac{dx_i(t)}{dt} < 0)$ **Alors** $\Delta t = Min\{\Delta t, \frac{-x_i(t)}{\frac{dx_i(t)}{dt}}\}$; **Fin Si**

Si $(\frac{dx_i(t)}{dt} > 0)$ **Alors** $\Delta t = Min\{\Delta t, \frac{1-x_i(t)}{\frac{dx_i(t)}{dt}}\}$; **Fin Si**

Fin Si

Fin Pour

$S_1 = \sum_{i=1}^{n} \frac{dx_i(t)}{dt}\frac{du_i(t)}{dt}$; $S_2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{dx_i(t)}{dt}T_{ij}\frac{dx_j(t)}{dt}$; $sw\_optimale = 0$;

Si $(S_2 > 0)$ **Alors**

Si $(\Delta t < \frac{S_1}{S_2})$ **Alors**

$\Delta t = \frac{S_1}{S_2}$; $sw\_optimale = 1$;

Fin Si

Fin Si

Si $(iter < R\_ITER$ **ET** $sw\_optimale = 0)$ **Alors** $\Delta t = q\Delta t$ **Fin Si**

Pour $(i = 0; i \leq n; i++)$ **faire**

Si $(x_i(t) \in \{0,1\})$ **Alors**

$u_i(t+\Delta t) = u_i(t) + \frac{du_i(t)}{dt}\Delta t$;

$x_i(t+\Delta t) := \frac{1}{2}(1+\tanh(\frac{u_i(t+\Delta t)}{u_0}))$;

Sinon

Si $(x_i(t) + \frac{dx_i(t)}{dt}\Delta t \in [\varepsilon, 1-\varepsilon])$ **Alors**

$x_i(t+\Delta t) = x_i(t) + \frac{dx_i(t)}{dt}\Delta t$;

Sinon

$u_i(t) := \frac{u_0}{2}\ln(\frac{x_i(t)}{1-x_i(t)})$; $u_i(t+\Delta t) = u_i(t) + \frac{du_i(t)}{dt}\Delta t$;

$x_i(t+\Delta t) = \begin{cases} 0, & x_i(t) + \frac{dx_i(t)}{dt}\Delta t < \varepsilon ; \\ 1, & x_i(t) + \frac{dx_i(t)}{dt}\Delta t > 1-\varepsilon. \end{cases}$

Fin Si

Fin Si

Fin Pour

$t = t + \Delta t$;

$iter = iter + 1$;

jusqu'à ce que $(\underset{i\in\{1,\ldots,n\}}{Max}\{|x_i(t) - x_i(t+\Delta t)|\} < \varepsilon^{1.5}$ **OU** $\underset{i\in\{1,\ldots,n\}}{Max}\{|x_i(t) - x_i(t+\Delta t)|\} < \varepsilon)$

Fin

Figure III.7 Point d'équilibre de l'algorithme de CHN.

4.4 L'optimisation quadratique via les réseaux de Hopfield continus

Dans cette sous-section, nous présentons l'essentiel de la démarche de résolution d'un programme quadratique à variables binaires basée sur les réseaux de Hopfield continu (CHN). Ces réseaux ont été largement utilisés pour résoudre les problèmes de programmation quadratique [Tatsumi, 2002], [Wena et al., 2009]. Le problème le plus connu est celui du voyageur de commerce qui est un problème d'optimisation de type NP-complet. Il s'agit de relier n villes en minimisant le chemin parcouru [Hopfield et Tank, 1985].

Cette méthode de résolution sera appliquée, par la suite, pour résoudre des problèmes de programmation quadratique et surtout le problème de détermination du nombre de classes ainsi que les centres initiaux de chaque classe pour le problème de la classification des documents en utilisant le problème de l'ensemble stable maximum au chapitre IV.

Un problème de programmation quadratique soumis à des contraintes d'égalité et / ou d'inégalité linéaires et à variables mixtes est un problème qui peut se mettre sous la forme suivante :

$$(PQG)\begin{cases} \text{Min } f(x) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}x_i x_j + \sum_{i=1}^{n} c_i x_i \\ \text{s.c} \\ \sum_{i=1}^{n} a_{ki}x_i \le b_k, \qquad k \in \{1, \dots, m_1\} \\ \sum_{i=1}^{n} a_{ki}x_i = b_k, \qquad k \in \{m_1 + 1, \dots, m\} \\ x_i \in \{0,1\}, \qquad i \in \{1, \dots, n\} \end{cases}$$

Dans un premier temps, la résolution de ce programme quadratique (PQG) via la fonction d'énergie du réseau de Hopfield continu (CHN) nécessite la transformation de l'ensemble de contraintes linéaires d'inégalités à un ensemble de contraintes linéaires d'égalités. Cela se fait en introduisant des variables supplémentaires qui s'appellent les variables d'écarts $x_{n+k} \in [0,1]$, avec $k \in \{1, \dots, m_1\}$. Ils représentent l'écart pour chaque inégalité pour la compléter ou la transformer en égalité. Ces variables sont incluses dans le modèle précèdent, pondérées par les coefficients $a_{k,n+k}$, pour $k \in \{1, \dots, m_1\}$:

$$(PQ)\begin{cases} \text{Min} \quad f(x) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}x_i x_j + \sum_{i=1}^{n} c_i x_i \\ \text{s.c} \\ \sum_{i=1}^{n} a_{ki}x_i + a_{k,n+k}x_{n+k} = b_k, \quad k \in \{1, \dots, m_1\} \\ \sum_{i=1}^{n} a_{ki}x_i = b_k, \quad k \in \{m_1 + 1, \dots, m\} \\ x_i \in \{0,1\}, \quad i \in \{1, \dots, n\} \\ x_{n+k} \in [0,1], \quad k \in \{1, \dots, m_1\} \end{cases}$$

Sans perte de généralité, on peut considérer le programme quadratique à variables mixtes soumis à des contraintes linéaires d'égalité suivant :

$$(PQ)\begin{cases} \text{Min} \quad f(x) = \frac{1}{2}x^t Q x + c^t x \\ \text{s.c} \\ Ax = b \\ x_i \in \{0,1\}, \quad i \in \{1, \dots, n\} \\ x_{n+k} \in [0,1], \quad k \in \{1, \dots, m_1\} \end{cases}$$

Où Q est une matrice réelle carrée d'ordre $n + m_1$, $c \in \mathbb{R}^{n+m_1}$, A est une matrice réelle rectangulaire de dimension $m \times (n + m_1)$ et $b \in \mathbb{R}^m$.

Les réseaux de Hopfield continus permettent la résolution de problèmes de programmation quadratique dans laquelle la recherche de l'extremum se ramène à la minimisation d'une fonction d'énergie quadratique E (Eq. **III.4**). L'approche retenue impose de déterminer une "bonne" fonction d'énergie qui place les solutions du problème comme des minima (locaux) de cette fonction. Ainsi, le réseau évolue vers une configuration stable, constituant une solution optimisée du problème. Pour cela on cherche à exprimer la fonction de coût et les contraintes sous la forme d'une énergie d'un réseau de Hopfield :

$$\mathbf{E = Coût + Contraintes}$$

En général, une telle fonction d'énergie prend en considération ou équivalente à la fonction objective du problème quadratique (PQ), tandis que les contraintes du problème sont incluses dans la fonction d'énergie sous le contrôle des termes de pénalité.

La fonction $E(x)$ est définie de la manière suivante :

$$\forall x \in [0,1]^n, \qquad E(x) = E_O(x) + E_c(x)$$

Où :

La fonction E_O est proportionnelle positivement à la fonction coût du problème (PQ).

$$E_O(x) = \frac{\propto}{2} f(x) \qquad \text{avec} \qquad \propto > 0$$

La fonction E_c est une fonction quadratique qui pénalise non seulement les contraintes violées du problème, mais garantit aussi la faisabilité de la solution obtenue par le réseau de Hopfield continu (CHN).

Rappelons que les paramètres de pénalité sont des coefficients qui doivent assurer un bon équilibre entre la minimisation du coût et la satisfaction des contraintes. Récemment, une fonction d'énergie a été proposée pour résoudre le Problème de Sac-à-dos Quadratique Généralisé. Pour assurer la réalisabilité du point d'équilibre des réseaux de Hopfield basés sur cette fonction, une procédure de réglage des paramètres de la fonction ainsi définie dite la procédure d'hyperplan a été proposée [Talavàn et Yànez, 2006].

A base de ce principe, différentes formulations des fonctions d'énergie sont proposées pour le programme quadratique (PQ) en fonction de E_c. Par ailleurs, une fonction d'énergie a été proposée pour résoudre des problèmes de programmation quadratiques [Talavàn et Yànez, 2006]. Cette fonction d'énergie est une généralisation des deux fonctions d'énergie proposées dans [Aiyer, 1991], elle est donnée comme suit :

$$E_c(x) = \frac{1}{2}(Ax)^t \Phi (Ax) + x^t \mathrm{diag}(\gamma)(1 - x) + \beta^t (Ax) \qquad \text{(III. 5)}$$

Avec les paramètres Φ est une matrice symétrique d'ordre m, $\mathrm{diag}(\gamma)$ est une matrice diagonale à diagonale $\gamma \in \mathbb{R}^{n+m_1}$ et $\beta \in \mathbb{R}^m$.

La présence du terme : $x^t\, \mathrm{diag}(\gamma)(1 - x) = \sum_{i=1}^{n} \gamma_i x_i\,(1 - x_i)$, Dans l'expression de la fonction E_c contraint le réseau de Hopfield continu à converger vers l'un des sommets de l'hypercube $[0, 1]^n$.

Alors, pour résoudre le programme quadratique via les réseaux de Hopfield continus, nous considérons la fonction d'énergie suivante [Talavàn et Yànez, 2006] :

$$E(x) = \alpha\left(\frac{1}{2}x^t Q x + c^t x\right) + \frac{1}{2}(Ax)^t \Phi (Ax) + x^t \mathrm{diag}(\gamma)(1-x) + \beta^t (Ax)$$

Sous la forme algébrique :

$$E(x) = \frac{\propto}{2}\sum_{i=1}^{n}\sum_{j=}^{n} q_{ij}x_i x_j + \propto \sum_{i=1}^{n} c_i x_i + \frac{1}{2}\sum_{k,l=1}^{m} \phi_{kl} \sum_{i,j=1}^{n+m_1} a_{ki}a_{lj}x_i x_j + \sum_{i=1}^{n} \gamma_i x_i\,(1-x_i)$$
$$+ \sum_{k=1}^{n} \beta_k \sum_{i=1}^{n+m_1} a_{ki}x_i \qquad \text{(III. 6)}$$

Chaque coefficient ϕ_{kl} de la matrice symétrique Φ pénalise le produit des contraintes k et l, cette expression pénalise la violation de toute contrainte du (QP), y compris celles garantissant l'admissibilité des solutions.

Revenant à la forme algébrique de la fonction d'énergie du réseau de Hopfield (Eq. **III.4**) :

$$E(x) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} T_{ij}x_i x_j - \sum_{i=1}^{N} {I_i}^b x_i \qquad \text{(III. 4)}$$

On voit que les pondérations T_{ij} sont les coefficients des termes quadratiques $x_i x_j$ de l'équation, et les seuils externes, ${I_i}^b$, sont les coefficients des termes linéaires x_i dans la fonction d'énergie généralisée (Eq.**III.6**). Par identification algébrique, on obtient :

$$\begin{cases} T_{ij} = -\left(\propto q_{ij} + \sum_{k=1}^{m}\sum_{l=1}^{m} \phi_{kl}\, a_{ki}a_{lj} - 2\delta_{ij}\gamma_i\right), & \forall (i,j) \in \{1,\dots,n+m_1\}^2 \\ {I^b}_j = -(\alpha c_j + \sum_{k=1}^{n} \beta_k a_{kj} - \gamma_j), & \forall j \in \{1,\dots,n+m_1\} \end{cases} \qquad \text{(III. 7)}$$

Avec $\gamma_i = 0,\ \forall i \in \{n,\dots,n+m_1\}$ et le symbole de Kroenecker :

$$\delta_{ij} = \begin{cases} 1, & \text{si } i = j \\ 0, & \text{sinon} \end{cases}$$

L'approche retenue impose de déterminer un "bon" réglage des paramètres ainsi introduits. Ces paramètres placent les solutions du problème QP comme des minima (locaux) de la fonction

d’énergie du réseau de Hopfield. Ainsi, le réseau évolue vers un état stable, constituant une solution optimisée du problème quadratique.

4.5 Méthode de paramétrage pour une solution réalisable

On note que les poids synaptiques et les seuils n'évoluent pas durant le processus de convergence du réseau de Hopfield, mais ils sont complètement spécifiés par l'énergie à minimiser [Hopfield et Tank, 1985]. Le problème est classifié dans les problèmes de minimisation d’une fonction sans contrainte. Toute la difficulté du problème réside dans le choix des paramètres de contrôle du coût et des contraintes α, Φ, γ et β. Ces paramètres pondérant l'importance relative des termes du problème (QP). Ils doivent assurer un bon équilibre entre la minimisation du coût et la satisfaction des contraintes. Pour résoudre le problème de réglage de paramètres, une nouvelle méthode dite d’hyperplan a été proposée [Talavàn et Yànez, 2006]. Cette méthode assure la réalisabilité du point d’équilibre associé à la stabilité du réseau de Hopfield continu. Elle consiste à diviser l’hypercube $[0,1]^n$ par un hyperplan contenant toutes les solutions réalisables. Une solution du problème d'optimisation sera un point corner de l’hypercube $[0,1]^n$ recherché sous la forme d'un vecteur d'état stable du réseau.Cet état stable $x^s \in \{0,1\}^n$ doit satisfaire les trois conditions suivantes [Talavàn et Yànez, 2006] :

$$\forall i \in \{1, \dots, n\}: \begin{cases} \dfrac{\partial E(x^s)}{\partial x_i} \geq 0 & \text{si } {x^s}_i = 0 \\ \dfrac{\partial E(x^s)}{\partial x_i} \leq 0 & \text{si } {x^s}_i = 1 \\ \dfrac{\partial E(x^s)}{\partial x_i} = 0 & \text{si } 0 < {x^s}_i < 1 \end{cases}$$

Ces conditions analytiques d’un point de stabilité du réseau de Hopfield continu imposent aux paramètres de contrôle de la fonction d’énergie un système d’inéquations à valider. Ce système porte alors sur une décomposition de l’espace des solutions non réalisables du problème étudié. Il faut signaler que cette procédure a été utilisée et détaillée pour résoudre un nombre important de problèmes réels, on cite dans ce cadre le problème de l’ensemble stable de taille maximale dans un graphe, le problème d’allocation de tâches aux processeurs et le problème de satisfaction maximale des contraintes [Ettaouil et al., 2010], [Ettaouil et al., 2012], [Ettaouil et al., 2013].

5 Conclusion

Les réseaux de neurones de Hopfield ont montré de bons résultats et largement utilisés dans différents domaines de traitement de l'information. Dans le domaine de regroupement des documents textuels, les contraintes du problème de classification sont exprimées par la connectivité du réseau (graphe), l'état initial du réseau correspond aux documents textuels, et l'état final du réseau au regroupement qu'il propose.

Dans ce chapitre, nous avons mis l'accent sur l'ensemble stable de taille maximum pour définir les modélisations et les différents attributs essentiels. Nous avons réalisé un tour d'horizon sur les méthodes proposées dans la littérature pour déterminer l'ensemble stable maximum dans un graphe. D'autre part, nous avons présenté l'architecture des réseaux de Hopfield continus et une large étude sur le concept d'apprentissage et les états stables. Ces réseaux de neurones artificiels fournissent des algorithmes d'efficacité très élevée [Talavàn et Yànez, 2002], [Talavàn et Yànez, 2005]. Dans la suite, nous allons utiliser les principes des méthodes proposées dans ce chapitre pour résoudre les problèmes de classification et de regroupement des documents textuels.

PARTIE II : APPLICATION AU CLASSIFICATION AUTOMATIQUE DE DOCUMENTS

CHAPITRE IV : AMELIORATION DU CLUSTERING VIA LES RESEAUX DE NEURONES ET LENSEMBLE STABLE MAXIMUM

1 Introduction

Aujourd'hui, les problèmes de regroupement jouent un rôle essentiel dans l'exploration des données, où nous avons de nombreuses applications dans des domaines tels que la bio-informatique, l'analyse de données du web, l'exploration de textes et l'exploration de données scientifiques. Etc.

Notre travail fait partie des techniques de classification non supervisée (clustering). Il s'agit d'un processus d'apprentissage automatique où les objets qui se ressemblent sont regroupés selon le principe de maximiser la similarité intra-classe et de minimiser la similarité interclasse sans connaître les étiquettes des données. De nombreuses stratégies de regroupement sont disponibles dans la littérature [Han et al., 2012], [Larose, 2014], parmi lesquelles deux catégories principales connues sous le nom de regroupement hiérarchique et de regroupement de partitionnement.

Dans le clustering, l'un des problèmes les plus difficiles à résoudre est la détermination du nombre de clusters dans un ensemble de données, qui est considéré comme un paramètre d'entrée de base pour la plupart des algorithmes de clustering [Dumont et al., 2018]. Les résultats de clustering dépendent fortement de nombre de classes fixé à l'avance. Par conséquent, il faut choisir le bon nombre de classes pour aspirer à une bonne qualité de classification.

La plupart des algorithmes de regroupement sélectionnent à l'avance les centres initiaux de façon aléatoire, ce qui affecte considérablement la qualité des résultats de regroupement. Dans le même contexte, K-Means est une méthode d'apprentissage non supervisée [Wu, 2012], qui est largement utilisée dans le regroupement de textes. C'est un algorithme dont le nombre de classes est donné au départ et qui est également bien construit. Mais la difficulté réside dans la mise en cluster obtenue qui dépend fortement du nombre de classes et du choix des centres initiaux, spécialement quand le jeu de données est grand et qu'on n'ait pas a priori des hypothèses sur les données [Ashour et Fyfe, 2014]. Cependant, il peut être bloqué dans un minimum local et provoquer un résultat instable. Autrement dit, si nous réinitialisons

l'algorithme avec d'autres valeurs, il peut converger vers une autre solution locale [Kuncheva et Bezdek, 1997], [Bezdek et Hathaway, 1994], [Kovesi et al., 2001], [Sarkar et al., 1997].

Pour surmonter les lacunes susmentionnées, nous proposons une méthode de détection automatique du nombre de clusters et des centres initiaux, qui sont les paramètres d'entrée de l'algorithme classique de K-Means [Karim et al., 2018].

Nous avons pensé à modéliser notre problème sous forme de graphe, et puis chercher un modèle qui a le même principe que le clustering, un modèle qui représente la structure et les relations entre les données, c'est l'ensemble stable maximale qui a le même principe que le clustering. Elle consiste à trouver le plus grand ensemble stable dans un graphe ou l'ensemble des sommets qui sont déconnectés, et dans le clustering les centres des clusters doivent être dissimilaires.

La méthode proposée est divisée en quatre étapes :

- La première consiste à construire un graphe permettant de déterminer un ensemble stable de taille maximale.
- La deuxième consiste à modéliser le problème de détermination du nombre de classes comme un problème d'ensemble stable maximum (MSSP), qui peut être modélisé sous forme d'un programme quadratique 0-1 (QP).
- La troisième étape consiste à appliquer le réseau de Hopfield continu (CHN) pour résoudre le problème de QP. Par conséquent, le modèle résolu avec Hopfield est un modèle MSSP simple avec une seule contrainte, dans ce cas, Hopfield peut fréquemment converger vers le nombre optimal de clusters (minimum global). Et même si le réseau de Hopfield est piégé dans un minimum local, il s'agit d'une borne supérieure du nombre de classes K. Ainsi le modèle MSSP fournit des centres initiaux spécifiques et non aléatoires, dans plusieurs cas les centres réels puissent être choisis parmi eux.
- La quatrième étape concerne l'implémentation de la solution obtenue comme paramètre d'entrée dans K-Means, Nous exécutons K-Means avec des valeurs inférieures ou égales à ce minimum jusqu'à la convergence, pour cela l'approche est rapide et efficace sur des grands ensembles de données. Par contre dans les méthodes les plus connues pour déterminer le nombre de classes k, la méthode K-Means est exécutée avec différentes valeurs de k (de k=2 jusqu'à la taille du corpus-1) et on choisit k à partir duquel la variance ne diminue pas, ce qui est très compliqué pour les grands corpus et prend énormément de temps.

- Finalement dans la dernière étape nous comparons K-Means amélioré (KM_MSSP) avec d'autres méthodes par des indices de mesure de qualité.

Dans ce travail nous nous sommes focalisés sur les données textuelles spécialement en langue anglaise dans le but de partitionner un ensemble S de n documents à un nombre prédéterminé de k topics S_1, S_2,.., S_k, tels que les documents assignés à chaque topic sont similaires. L'approche est rapide et efficace sur des grands ensembles de données, elle est exécutée avant l'algorithme de regroupement, ce qui signifie que notre approche est indépendante de toute méthode de clustering qui commence par k centres.

Ce chapitre est structuré comme suit : la section 2 décrit le concept de regroupement de textes et spécialement le traitement préalable des données et la représentation des mots et des documents. La section 3 aborde le problème de l'ensemble stable maximal et le réseau de Hopfield continu, qui sont les principales composantes de la méthode proposée, ainsi la reformulation mathématique. Dans la section 4, nous présentons l'implémentation et les résultats expérimentaux de la méthode proposée.

2 Préparation des données

La classification ou le regroupement de documents est une tâche primordiale dans la fouille de textes. Elle consiste à regrouper des documents similaires (actualités, tweets, etc.), et apparaît dur lorsque les topics ne sont pas connus à l'avance. Son but est de regrouper tous les documents relatifs à une thématique particulière afin d'organiser une collection de documents et faciliter l'accès à l'information, son processus se compose des étapes traitées dans les sous sections suivantes [Aliguliyev, 2009] :

2.1 Prétraitement des textes

Dans le but de représenter chaque texte dans le modèle d'espace vectoriel, qui est un modèle algébrique permettant de représenter les documents textuels comme des vecteurs de termes, plusieurs tâches de prétraitement sont indispensables à effectuer avant l'utilisation des algorithmes d'exploration de données textuelles. Dans ce travail, il s'agit notamment des tâches suivantes :

- Décomposer le texte en unités de mots en utilisant les espaces comme délimiteur, pour le convertir ensuite en un sac de mots.

- Éliminer tout symbole qui ne correspond pas à une lettre de l'alphabet (points, virgules, traits d'union, chiffres etc.).
- Supprimer des mots-vides qui n'ont pas de sens significatif et qui souvent sont trop fréquents dans des textes, par exemple (articles, prépositions, mots grammaticaux...). Pour cela, on peut créer un ensemble ou un dictionnaire de mots vides pour un domaine spécifié.
- Le "stemming" qui est le processus de réduction des mots à leur forme racine, est effectué. Par exemple, les mots « national, nationalité et nationalisation» sont remplacés par leur racine « national » et les verbes conjugués par leurs modes infinitifs. Nous avons utilisé l'algorithme de Porter pour remédier à cette étape [Porter, 1980].
- Un filtre de prétraitement est appliqué aux données pour éliminer les termes qui ont un faible poids, ce qui conduit à une réduction significative de la dimensionnalité de l'espace de travail sans perte de performance de regroupement.

2.2 Pondération des mots

La pondération de termes a pour but de déterminer de manière quantitative la représentation d'un terme, en attribuant à chaque terme un poids. Il existe différentes méthodes pour calculer ce poids. Ces méthodes sont basées sur les observations suivantes : Plus le terme t est fréquent dans un document d, plus il est en rapport avec le sujet de ce document. Plus le terme est fréquent dans une collection, moins il sera utilisé comme discriminant entre les documents.

La méthode de pondération souvent utilisée en recherche d'information et particulièrement dans la fouille de textes est la conception terme-document : TFIDF (TermFrequency / Inverse Document Frequency). C'est une mesure statistique qui permet d'évaluer l'importance d'un terme dans un document, relativement à une collection ou un corpus, elle est définie comme suit :

$$w_{ij} = TF_{ij} \times IDF_i \qquad (IV.1)$$

Avec,

$$TF_{ij} = \frac{n_{ij}}{|d_j|}$$

$$IDF_i = \log\left(\frac{|D|}{|\{d_j : t_i \in d_j\}|}\right)$$

Où :

n_{ij} : Nombre d'occurrences de terme t_i dans le document d_j.

$|d_j|$: Nombre total de termes dans le document d_j.

$|D|$: Nombre total de documents dans le corpus.

$|\{d_j : t_i \in d_j\}|$: Nombre de documents où le terme t_i apparait.

2.3 Représentation du document

L'idée de base est de représenter les documents par des vecteurs et de mesurer la proximité entre ces documents. Le principe est de coder chaque élément du sac de mots par un scalaire (nombre) appelé tfidf (Eq. **IV.1**) pour donner un aspect arithmétique quantifiable aux documents textes.

2.4 Mesure de similarité

Il existe plusieurs mesures de similarité entre les documents dans la littérature. En particulier les plus utilisées pour le regroupement des documents nous trouvons la distance euclidienne, Manhattan et la similarité cosinus. Cette dernière est fréquemment utilisée dans la comparaison de documents textuels, est une mesure qui calcule le cosinus de l'angle entre deux vecteurs d_i et d_j:

$$\cos(d_i, d_j) = \frac{< d_i, d_j >}{\|d_i\| \times \|d_j\|}$$

Où

$< d_i, d_j >$ représente le produit scalaire des vecteurs documents d_i et d_j.

$\|d_i\|$ et $\|d_j\|$ représentent respectivement les normes des vecteurs documents d_i et d_j.

Rappelons que la valeur du cosinus est de 1 lorsque les documents sont identiques et de 0 lorsqu'ils n'ont rien en commun.

Dans ce travail, nous utilisons comme métrique la dissimilarité entre deux vecteurs documents d_i et d_j basée sur la mesure cosinus présentée comme suit :

$$\mathrm{Dis}(d_i, d_j) = 1 - \cos(d_i, d_j)$$

Par conséquent, la valeur de la dissimilarité est de 0 lorsque les documents sont identiques et de 1 lorsqu'ils sont bien séparés.

3 Description de la méthode proposée

Nous présentons dans cette section la démarche adoptée pour la détermination du nombre de classes ainsi que les centres initiaux de ces classes. Cette démarche est simple et générale pour tous les algorithmes de classification qui exigent un nombre de classes connu à l'avance et qui s'initialisent par des centres initiaux.

Nous supposons que nous disposons d'un corpus de n documents et nous voulons le diviser en groupes de telle sorte que les membres de chaque groupe soient aussi semblables (similaires) que possible les uns aux autres. Le schéma global de l'algorithme est illustré dans la figure **(IV.1)**. Le processus spécifique de l'algorithme est le suivant :

Après l'étape de prétraitement (expliqué dans la section 2.1 de ce chapitre), nous calculons la matrice de poids terme-document qu'on note W. Il s'agit d'une matrice bidimensionnelle $m \times n$ dont les lignes sont les termes et les colonnes sont les documents, de sorte que chaque entrée (i, j) représente le poids w_{ij} (tf-idf) du terme t_i dans le document d_j comme le montre l'équation **(IV.1)**.

Sur la base de la distance cosinus entre les documents, nous créons une matrice carrée de dissimilarité $B = (b_{ij})$ d'ordre n. Par conséquent, les nœuds $V = \{v_1, v_2, \ldots, v_n\}$ et les arêtes E du graphe $G = (V, E)$ se construisent de sorte que chaque sommet v_i représente un document. Pour construire une arête entre le document v_i et le document v_j, nous calculons d'abord la similarité entre eux, puis si $b_{ij} > s$, il y a une connexion (arête) dans le graphe G entre le nœud v_i et le nœud v_j, avec s est un paramètre de dissimilarité.

Ensuite, nous représentons un problème de regroupement de documents textuels sous la forme d'un problème d'ensemble stable de taille maximal (MSSP). Ensuite, nous utilisons la formulation en programmation quadratique à variables bivalentes QP pour représenter le problème MSSP. Puis le réseau neuronal de Hopfield continu est mis en œuvre pour résoudre le problème QP.

La solution est désignée comme un vecteur ayant les valeurs 0 et 1. Si la valeur à la position j est 1, cela signifie que le $j^{éme}$ document est sélectionné comme centre, sinon il n'est pas sélectionné.

Par conséquent, nous obtenons la taille de l'ensemble stable maximale résultante qui représente le **nombre de clusters k**.

Ainsi nous obtenons les nœuds proposés par l'ensemble stable maximale résultante qui représentent les **centres initiaux**.

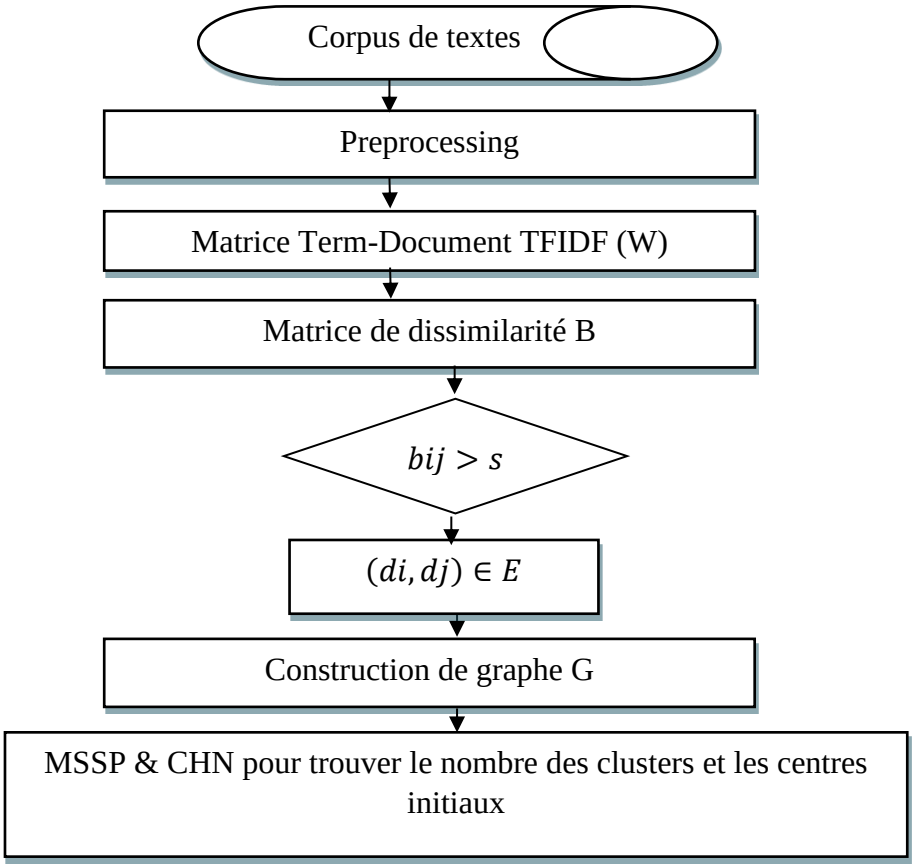


Figure IV.1. MSSP et CHN pour trouver le nombre des clusters et les centres initiaux

3.1 Construction du graphe

Pour expliquer les principales étapes de ce travail on présente un exemple comme suit :

- **Exemple :**

Soit un corpus C de sept documents tirés de l'ensemble de données « 20 newsgroup » tel que trois documents appartiennent à la classe "talk.religion.misc", deux documents appartiennent à la classe "misc.forsale", et deux documents appartiennent à la classe "comp.graphics". Après l'étape de prétraitement, nous calculons la matrice terme-document W.

Pour construire le graphe : Nous devons déterminer l'ensemble des nœuds V, de sorte que chaque nœud du graphe soit représenté par un document. Ainsi, nous avons sept nœuds :

$$V = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7\}.$$

Pour construire les arêtes E, nous calculons la dissimilarité entre les différents documents $(i, j) \in \{1, \dots, 7\}^2$ comme ci-dessous :

$$dis(d_i, d_j) = 1 - \cos(d_i, d_j)$$

On obtient la matrice de dissimilarité B, est une matrice symétrique d'ordre 7 dont les colonnes sont les documents et chaque entrée (i, j) représente le poids w_{ij} (tfidf) du terme t_i dans le document d_j.

Ensuite, nous déterminons les connexions (les arrêtes) entre les nœuds, en utilisant le paramètre de dissimilarité s de telle sorte que :

$$\begin{cases} (d_i, d_j) \in E & \text{si } b_{ij} > s \\ (d_i, d_j) \notin E & \text{sinon} \end{cases}$$

Ainsi, à partir de cette étape, nous pouvons construire notre graphe non orienté $G = (V, E)$ avec $V = \{d_1, d_2, .., d_7\}$ et E est l'ensemble des neuf arêtes obtenues via la matrice de dissimilarité comme suit :

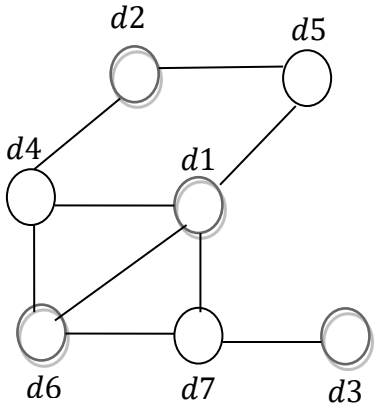


Figure IV.2 Graphe à 7 nœuds

3.2 Modélisation sous forme du problème MSSP

Étant donné un graphe non orienté $G = (V, E)$. Un ensemble stable S d'un graphe G est un ensemble de nœuds avec la propriété que tout couple de sommets de S ne soit pas adjacent. Autrement dit, chaque sommet de l'ensemble S n'est relié à aucun autre sommet de S. Le problème de l'ensemble stable maximal (MSSP) consiste à trouver un ensemble stable de taille maximale.

Nous utiliserons la formulation en programme quadratique décrite dans l'article suivant [Ettaouil et al., 2010]. La formulation mathématique de MSSP peut être définie de la manière suivante :

Soit S est l'ensemble stable. Pour tout sommet v_i, on associe une variable d'affectation binaire x_i, qui prend la valeur 1 si le sommet v_i appartient à l'ensemble stable S et 0 sinon.

$$\forall i \in \{1, \dots, n\}, \qquad x_i = \begin{cases} 1 & \text{Si } v_i \in S \\ 0 & \text{Sinon} \end{cases}$$

La contrainte imposée étant : deux sommets adjacents v_i et v_j ne peuvent pas être dans S :

$$x_i x_j = 0$$

Cette contrainte quadratique peut être agrégée dans une seule contrainte :

$$x^t C x = 0$$

Où C est une matrice symétrique d'ordre n définie comme étant la matrice d'adjacence du graphe par :

$$c_{ij} = \begin{cases} 1 & \text{si } (v_i, v_j) \in E \\ 0 & \text{sinon} \end{cases}$$

Notre objectif est de maximiser la taille de l'ensemble stable S. Ensuite, nous pouvons définir la fonction objective F (x) de la manière suivante :

$$F(x) = -\sum_{i=1}^{n} x_i$$

Enfin, nous obtenons le programme quadratique 0-1 suivant (QP) avec une fonction linéaire soumise à des contraintes quadratiques représentant le problème du MSSP avec n variables binaires. Ainsi, le problème de l'ensemble stable de taille maximal (MSSP) considéré peut-être formulé sous la forme algébrique suivante :

$$(QP) \begin{cases} \text{Min } F(x) = -\sum_{i=1}^{n} x_i \\ \text{s.c} \\ x^t C x = 0 \\ x \in \{0,1\}^n \end{cases} \qquad (IV.2)$$

Diverses approches sont proposées pour résoudre ce modèle. Par exemple, la méthode des points intérieurs, les relaxations semi-définies [Ettaouil et Loqman, 2008] et les relaxations

lagrangiennes [Thiongane et al., 2005]. Dans la suite, nous présentons ce problème sous la forme d'une fonction énergétique associée au réseau de Hopfield continu.

3.3 Réseau de Hopfield continu pour résoudre le MSSP

Les réseaux de neurones artificiels sont des approches efficaces pour résoudre différents problèmes dans différents domaines [Ettaouil et Loqman, 2008], [Ettaouil et al., 2013], [Evansi et Sulaiman, 1996]. Le réseau neuronal de Hopfield a été introduit par Hopfield et Tank [Hopfield et Tank, 1985], au début des années 1980. Il a été le point de départ du nouveau domaine des réseaux de neurones et a également démontré sa capacité à trouver des solutions à des problèmes d'optimisation difficiles [Talavàn et Yànez, 2005]. De plus, ils ont présenté l'approche de la fonction énergétique afin de résoudre plusieurs problèmes d'optimisation [Ettaouil et al., 2013]. Leurs résultats ont encouragé un certain nombre de chercheurs à appliquer ce réseau à différents problèmes.

Le CHN est un réseau neuronal entièrement connecté, ce qui signifie que chaque neurone est connecté à tous les autres neurones. Les poids de connexion entre le neurone i et le neurone j sont représentés par w_{ij} et chaque neurone i a un décalage par le biais ou un bruit externe I_i^b [Hopfield et Tank, 1985]. La dynamique du CHN est décrite par l'équation différentielle suivante :

$$\frac{du}{dt} = \frac{u}{\tau} + Wx + I^b \qquad (IV.3)$$

Où u, x et I^b seront les vecteurs des états, des sorties et des biais des neurones. La fonction de sortie $x_i = g(u_i)$ est une tangente hyperbolique, qui est limitée en dessous par 0 et au-dessus par 1.

$$\forall i \in \{1, \ldots, n\}, \qquad g(u_i) = \frac{1}{2}\left(1 + tanh\left(\frac{u_i}{u_0}\right)\right) \qquad (IV.4)$$

Où $u_0 > 0$ est un paramètre utilisé pour contrôler le gain (pente) de la fonction d'activation.

L'approche de réseau de Hopfield continu pour résoudre un problème d'optimisation combinatoire, consiste à reformuler ce dernier en fonction énergétique associée au CHN. L'expression de cette fonction énergétique est définie par l'expression suivante :

$$E(x) = -\frac{1}{2}x^t W x - (I^b)^t x \qquad (IV.5)$$

Ou sous forme analytique :

$$E(x) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_i x_j - \sum_{i=1}^{n} I^b{}_i x_i \qquad (IV.5)$$

Afin de résoudre le problème de l'ensemble stable maximum en utilisant le réseau de Hopfield continu, nous définissons la fonction d'énergie généralisée pour le problème MSSP en se basant sur le modèle. Rappelons que le MSSP est modélisé comme un programme quadratique 0-1 qui consiste à minimiser une fonction linéaire soumise à des contraintes quadratiques **(IV.2)**.

La fonction énergétique doit être définie par deux expressions. Cette fonction d'énergie inclue la fonction objective $F(x)$ du problème QP et elle pénalise les contraintes violées du problème QP avec un terme quadratique et un terme linéaire. Par conséquent, la fonction d'énergie généralisée associée au CHN est définie par [Talavàn et Yànez, 2006] :

$$E(x) = \frac{\propto}{2}F(x) + \frac{1}{2}\phi x^t C x + x^t diag(\gamma)(1-x) \qquad (IV.6)$$

Avec $\alpha > 0,\ \phi \in \mathbb{R}$ et$\gamma \in \mathbb{R}$.

Certaines considérations doivent être prises en compte afin de simplifier l'expression mathématique de la fonction énergétique. La fonction énergétique généralisée pour le problème du QP est définie par le travail [Ettaouil et al., 2013] :

$$E(x) = -\frac{\propto}{2}\sum_{i=1}^{n} x_i + \frac{1}{2}\phi\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_i x_j + \gamma\sum_{i=1}^{n} x_i\,(1-x_i) \qquad (IV.7)$$

Pour résoudre le programme quadratique 0-1 qui consiste à minimiser une fonction linéaire soumise à des contraintes quadratiques, les ensembles suivants sont nécessaires à définir :

• H est un ensemble de l'hypercube de Hamming :

$$H = \{x \in [0,1]^n\}$$

• H_C est un ensemble de coins hypercubes de Hamming :

$$H_C = \{x \in H/\ x_i \in \{0,1\},\ \forall i \in \{1,\ldots,n\}\}$$

• H_F est un ensemble de solutions réalisables :

$$H_F = \{x \in H_C\ / x^t C x = 0\}$$

Pour déterminer les poids et les seuils, nous utilisons l'assimilation entre l'équation **(IV.7)** et la forme algébrique de la fonction énergétique généralisée **(IV.5)**. Ensuite, les poids et les seuils des connexions entre les n neurones sont :

$$\begin{cases} w_{ij} = -\phi c_{ij} + 2\delta_{ij}\gamma \\ I_i^b = \alpha - \gamma \end{cases} \qquad (IV.8)$$

Où δ_{ij} est le delta du Kroenecker tel que :

$$\delta_{ij} = \begin{cases} 1 & Si \quad i \neq j \\ 0 & Sinon \end{cases}$$

De cette manière, la programmation quadratique a été présentée comme une fonction énergétique du réseau de Hopfield continu. Les paramètres α, ϕ et γ doivent être sélectionnés de telle sorte que les points d'équilibre du CHN, associés au MSSP, soient réalisables. Pour résoudre une instance du problème QP, la procédure de paramétrage est utilisée. Cette procédure attribue des valeurs particulières à tous les paramètres du réseau, de sorte que tout point d'équilibre soit associé à une affectation valide de toutes les variables lorsque toutes les contraintes sont satisfaites. La procédure de paramétrage est basée sur les dérivées partielles de la fonction énergétique généralisée :

$$\frac{\partial E(x)}{\partial x_i} = E_i(x) = -\alpha + \phi \sum_{j=1}^{n} b_{ij}x_j + \gamma(1 - 2x_i) \qquad (IV.9)$$

Pour résoudre une instance du problème QP, il faut un réglage approprié de ces paramètres. Dans cette section, notre objectif est de déterminer ces paramètres. La méthode d'hyperplan utilisée pour déterminer le paramétrage [Talavàn et Yànez, 2006]. Cette procédure consiste à diviser l'hypercube H de Hamming par un hyperplan contenant toutes les solutions possibles. La méthode de l'hyperplan est brièvement expliquée ci-dessous. Avant de présenter cette méthode, certaines conditions sont imposées pour simplifier la démarche de la détermination de ces paramètres :

$$\phi > 0, \ \gamma \geq 0.$$

Pour minimiser la fonction objective, nous imposons la contrainte suivante : $\alpha > 0$.

Afin de garantir l'instabilité des points intérieurs $x \in H - H_C$, certaines conditions initiales sont imposées sur certains paramètres :

$$\forall i \in \{1, \ldots, n\}, \qquad w_{ii} = 2\gamma \geq 0$$

Le problème QP n'a qu'une seule famille de contraintes quadratiques :

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_i x_j = 0$$

Soit $x \in H_C - H_F$, dans ce cas, deux sommets non adjacents v_i et v_j peuvent être dans S et $x_j = x_i = 1$, par conséquent la condition d'instabilité suivante est imposée (avec $\varepsilon > 0$) :

$$-\propto +\phi - \gamma \geq \varepsilon$$

Par conséquent, le paramétrage est déterminé en résolvant le système suivant [Ettaouil et al., 2010].

$$\begin{cases} \propto > 0, \phi > 0, \gamma \geq 0 \\ -\alpha + \phi - \gamma = \epsilon \end{cases} \qquad (IV.10)$$

Enfin, les poids et les seuils du CHN peuvent être calculés en utilisant ces paramètres de réglage. Ensuite, nous obtenons un point d'équilibre pour le CHN en utilisant l'algorithme décrit dans [Talavàn et Yànez, 2005], ce dernier est considéré comme le nombre de classes.

Retournons à l'exemple 1 du graphe à 7 nœuds (Figure **IV.2**). Afin de déterminer le MSSP associé à ce graphe, nous pouvons le formuler comme un problème de l'ensemble stable de taille maximale (MSSP). Ce dernier est modélisé sous forme du programme quadratique (QP) suivante :

$$(QP)\begin{cases} Min\ F(x) = -\sum_{i=1}^{7} x_i \\ s.c \\ \quad x^t C x = 0 \\ \quad x \in \{0,1\}^7 \end{cases}$$

La matrice de contraintes C est donnée par :

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

En utilisant le réseau de Hopfield continu (CHN) pour résoudre le problème de la programmation quadratique (QP). La solution obtenue est :

$$x = (0,1,1,0,0,1,0)^t$$

Par conséquent, nous obtenons un ensemble stable du cardinalité maximale 3 du graphe G (Fig.**IV.2**), qui sera égale à trois sommets $S = \{d_2, d_3, d_6\}$.

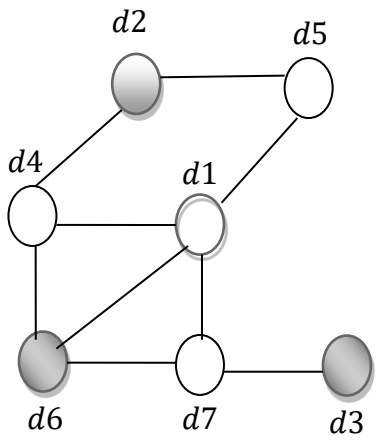


Figure IV.3. **Graphe à ensemble stable de taille maximale**

3.4 K-Means initialisé par le nombre optimal de clusters et les centres trouvés par MSSP (KM_MSSP)

L'algorithme K-Means est une méthode de regroupement couramment utilisée dans le regroupement de textes, qui permet aux centres de représenter des groupes en minimisant les erreurs. L'algorithme commence par un ensemble prédéfini de centres (qui peuvent être générer soit de manière aléatoire, soit au moyen de tout autre critère) dans cette implémentation, il est généré du MSSP. Il réalise des répétitions séquentielles du reste de l'échantillon en fonction de la similarité (en utilisant la distance cosinus) avec les centres tels que chaque document est attribué à la grappe ayant le centre le plus similaire. Ensuite, l'algorithme effectue un traitement itératif et ajuste la position du centre, jusqu'à ce qu'il n'y ait plus de réaffectation de motifs à de nouveaux centres de la grappe ou une diminution minimale de l'erreur quadratique ou que le nombre d'itérations ait dépassé un seuil. Cela produit une séparation des objets en groupes à partir desquels la métrique à minimiser peut être calculée.

Pseudo code KM_MSSP:

1. K-MOYENNES $(\{x_1, \dots, x_N\}, K)$
2. $(d_1, \dots, d_K) \leftarrow$ Selectionner les documents du MSSP $(\{x_1, \dots, x_N\}, K)$
3. Pour $k \leftarrow 1$ jusqu'à K
4. $c_k \leftarrow d_k$
5. Fin pour
6. Tandis que les critères de convergence n'ont pas été respectés

    Pour $k \leftarrow 1$ jusqu'à K

        $w_k \leftarrow \{\}$

        Pour $n \leftarrow 1$ jusqu'à N

          $j \leftarrow argmin_{j' \in \{1,..,K\}} |c_{j'} - x_n|$

            $w_k \leftarrow w_j \cup \{x_n\}$ (réaffectation des vecteurs)

        Fin pour

    Fin pour

    Pour $k \leftarrow 1$ jusqu'à K

        $c_k \leftarrow \frac{1}{|w_k|} \sum_{x \in w_k} x$ (recalcul des centres)

    Fin pour

7. Fin tandis
8. Retourne $\{c_1, \dots, c_K\}$

Algorithm IV.1 : KM_MSSP

La dernière étape concerne l'introduction de la solution obtenue par MSSP et CHN comme paramètre d'entrée dans K-Means (Fig. **IV.4**), et évalue les résultats du clustering sur la base de certains critères de validité.

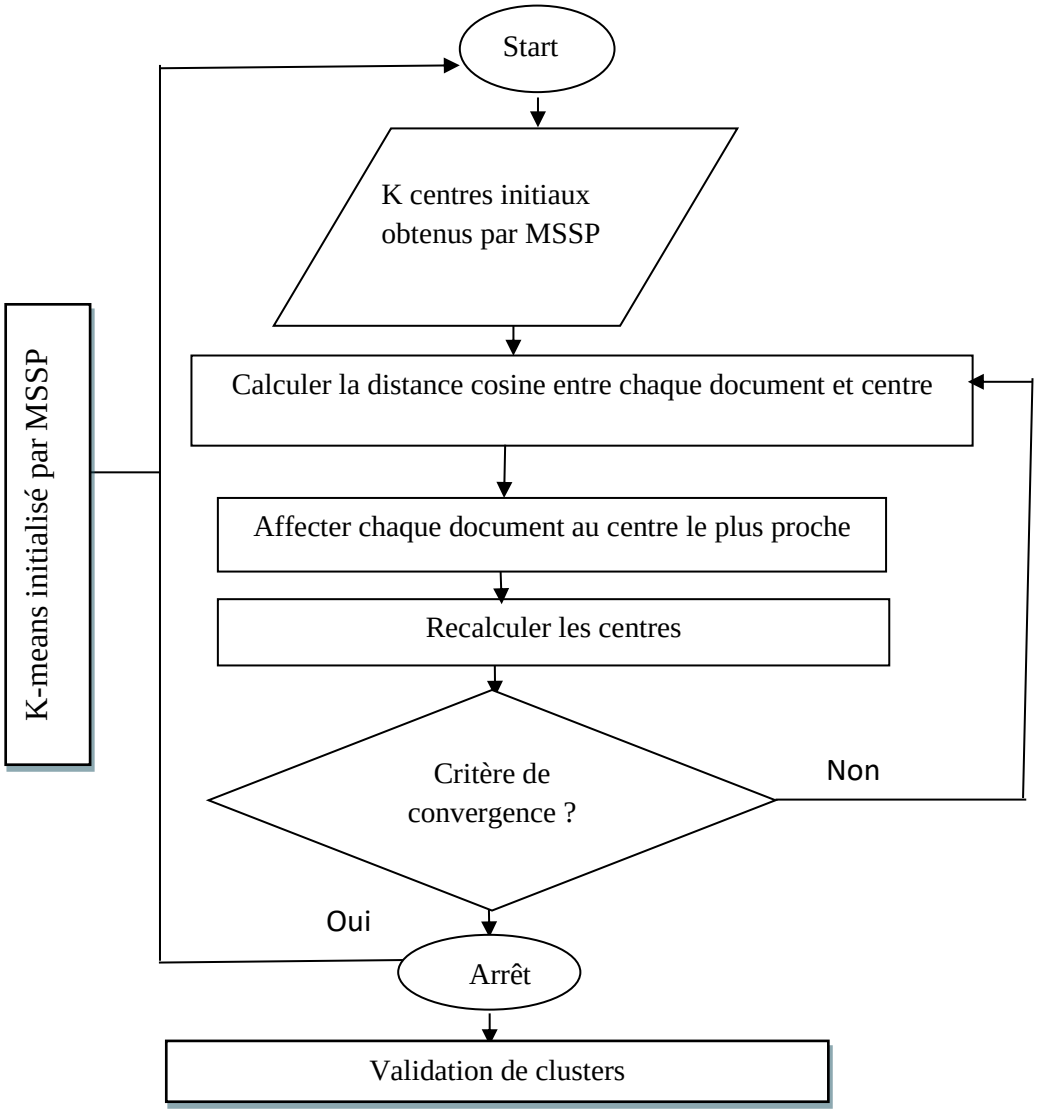


Figure IV.4. L'algorithme proposé KM_MSSP.

4 Résultats Expérimentaux

Pour évaluer et montrer l'intérêt pratique de l'approche proposée, nous avons effectué une série d'expérimentations pour déterminer un certain nombre de clusters, qui est un paramètre d'entrée de base pour plusieurs méthodes de clustering. La plupart de ces séries représentent un large spectre d'instances de dataset 20 newsgroups en faisant varier le nombre de classes. Ces instances sont : dataset1, dataset2, dataset3 et dataset4.

Aussi pour évaluer KM_MSSP nous avons effectué une série d'expériences sur des exemples tirés du jeu de données de la BBC en faisant varier le nombre de classes et le nombre de documents.

Ces expériences sont réalisées sur un ordinateur personnel équipé d'un core i5, d'un processeur à 2,50 GHz et de 6 Go de RAM.

Les points de départ sont choisis afin d'assigner les documents à des clusters de manière à maximiser le nombre d'ensembles stables. Par conséquent, le problème de l'ensemble stable maximum est déterminé. Enfin, les points de départ sont générés de manière aléatoire par l'expression suivante :

$$x_i = 0.999 + \frac{n+1-i}{n} 10^{-5} z$$

Où $i \in \{1, \ldots, n\}$ et z est une variable uniforme aléatoire dans l'intervalle $[-0{,}5; 0{,}5]$. Rappelons que n est le nombre de documents.

Le réglage des paramètres (Eq.**IV.10**) est déterminé en fixant α, ε et γ. Ensuite, le paramètre ϕ a été calculé à partir de l'équation. Les valeurs de ces paramètres sont les suivantes :

$$\begin{cases} \alpha = 1.0250, & \varepsilon = 10^{-6} \\ \gamma = 0.7, & \phi = \alpha + \gamma + \varepsilon \end{cases}$$

4.1 Description de l'ensemble des données

20NewsGroup : est une collection d'environ 20 000 articles de newsgroups, répartis dans 20 newsgroups différents, chacun correspondant à un sujet différent. Nous avons sélectionné un sous-ensemble de cet ensemble de données contenant au total 400 documents de plus de quatre catégories. Certaines sont très proches sémantiquement (par exemple "comp.sys.mac.hardware" et "comp.sys.ibm.pc.hardware") alors que d'autres n'ont aucun lien entre elles (par exemple "rec.sport.baseball" et "soc.religion.christian"). L'ensemble de données est disponible dans le dépôt d'apprentissage automatique de l'UCI.

BBC_Dataset se compose de 2225 documents du site d'information de la BBC correspondant à des articles dans cinq domaines d'actualité de 2004-2005. L'ensemble de données est classé en cinq catégories naturelles telles que les affaires, le divertissement, la politique, le sport et la technologie. Il est disponible dans [Greene et Cunningham, 2006]. Un sous-ensemble de l'ensemble de données est pris et séparé en quatre catégories, nous utilisons trois sous-ensembles des documents de l'ensemble de données de la BBC. Le premier sous-ensemble contient cinq sujets et 500 documents, le deuxième contient trois sujets et 1328 documents, le troisième contient quatre sujets et 1715 documents et chaque document a une étiquette de sujet unique, comme le montre le tableau **(IV.2)**.

BBC_Sport est constituée de 737 documents du site Internet de la BBC Sport correspondant à des articles sur le sport dans cinq domaines d'actualité de 2004-2005. L'ensemble de données est classé en cinq catégories naturelles telles que l'athlétisme, le cricket, le football, le rugby et le tennis. Il est disponible dans [Greene et Cunningham, 2006]. Nous utilisons un sous-ensemble des documents de l'ensemble de données de la BBC, qui contient trois sujets et 372 documents.

DATASET_3 : Ces données se trouvent dans un espace bidimensionnel et comportent trois classes. Le nombre total des documents est 24 et le nombre des attributs est 3178. L'ensemble de données n'est pas fourni avec des étiquettes de classe.

4.2 Détermination du nombre de clusters

Le tableau **(IV.1)** résume les résultats des exécutions de notre approche sur des instances de base de données 20NewsGroup. Pour chaque expérimentation nous avons exécuté l'algorithme 20 fois. Pour examiner la qualité de notre approche, une étude statistique a été représentée, cette étude est basée sur la performance de l'opérateur de calcul :

$$\text{Ratio} = \frac{\text{Nombre de classes obtenu par MSSP}}{\text{Nombre de classes existant dans la litérature}}$$

- Mode Ratio : le rapport entre le nombre de clusters le plus répétitif (mode) obtenu par MSSP dans un nombre d'exécutions et le nombre réel de classes existant dans la littérature.
- Ratio moyen : le rapport entre le nombre moyen de clusters obtenus par MSSP dans un nombre d'exécutions et le nombre réel de classes existant dans la littérature.
- Ratio minimum : le rapport entre le nombre réduit de clusters obtenus par la MSSP et le nombre réel de classes.
- Moyenne du temps CPU : le temps moyen consommé pour obtenir la solution en nombre d'exécutions.

Dans ce contexte, pour chaque cas, si le rapport minimum est très proche ou égal à 1, alors l'approche proposée a trouvé la meilleure solution.

Dans cette approche, nous avons calculé la valeur optimale obtenue par MSSP, nous avons obtenu les résultats présentés dans le tableau **(IV.1).**

Tableau IV.1- **Détermination du nombre de clusters en 20 exécutions**

	Description du Dataset		Nombre optimal de clusters obtenu par MSSP	Ratio			CPU time/ms
	Nombre de Documents	Nombre de classes		Mode	Min	Moy	
Dataset 1	11314	20	21	1.05	1.23	1.25	2.13
Dataset 2	7002	12	15	1.25	1.74	1.66	0.91
Dataset 3	4615	8	8	1	1.6	1.75	0.36
Dataset 4	2361	4	4	1	1.51	1.5	0.08

Comme le montre le tableau **(IV.1)**, dans l'ensemble de données Dataset 1, nous avons 11314 documents de 20newsgroup organisés en 20 classes différentes, chacune correspondant à un sujet différent, dans la quatrième colonne (valeur optimale obtenue par CHN) notre méthode donne 21 groupes, ce qui est très proche de la valeur de la troisième colonne 20 (nombre réel de classes), mais dans les Datasets 3 et 4, la solution est exactement égale au nombre réel de classes. Nous en déduisons que notre méthode donne pour chaque ensemble de données un résultat très proche ou égal au nombre réel de classes.

4.3 Détermination des centres initiaux

Pour montrer les avantages et tirer des remarques scientifiques sur l'adaptabilité de l'algorithme proposé. Ce dernier propose aussi des documents qui sont très dissimilaires qui peuvent être les centres initiaux de K-Means classique. Le paramètre de dissimilarité s est déterminé par plusieurs tests et fixé à ($s = 1$).

Comme le montre le tableau **(IV.2)**, dans le jeu de données BBC_2225, nous avons 2225 documents organisés en 5 classes différentes, dans la quatrième colonne (Nombre de classes obtenues par MSSP), notre méthode donne 6 classes pour BBC_2225 et 5 classes pour BBC_1715 qui sont très proches de la valeur de la troisième colonne (Nombre réel de classes). Cependant, dans BBC_500, BCC_1328, BBC_Sport1 et BBC_Sport2, la solution est égale au nombre réel de classes. Nous en concluons que notre méthode donne pour chaque ensemble de données un résultat très proche ou égal au nombre réel de classes. Dans la colonne 7, nous observons que les centres initiaux proposés par MSSP pour chaque instance de l'ensemble de données sont très dissemblables et appartiennent à des groupes différents. Le temps d'exécution est très limité, il varie en fonction de la taille de l'ensemble de données comme indiqué dans la colonne 8.

Tableau IV.2 - Centres initiaux obtenus par notre approche en 20 exécutions.

	Dataset	Nombre de documents	Nombre réel de classes	Nombre de clusters obtenu	Ratio			Centres initiaux obtenus		CPU/ time(sec)
					Mode	Moy	Min	Clusters	Documents	
BBC_News	BBC_2225	2225	5	6	1,2	1,19	1,2	Sport Politics Sport Entertainment Tech Business	199.txt 397.txt 324.txt 243.txt 334.txt 280.txt	4,2 s
	BBC_1328	1328	3	3	01	1,1	01	Business Politics Tech	394.txt 397.txt 284.txt	1,2 s
	BBC_500	500	5	5	01	1.1	01	Sport Business Entertainment Politics Tech	018.txt 080.txt 092.txt 074.txt 039.txt	0,18 s
	BBC_1715	1715	4	5	1,25	1,25	1,25	Sport Sport Politics Entertainment Tech	191.txt 324.txt 397.txt 154.txt 284.txt	0,06s
BBC_Sport	BBC_Sport1	737	5	5	01	01	01	Football Athletics Football Cricket Rugby	263.txt 026.txt 147.txt 077.txt 067.txt	0,01s
	BBC_Sport2	372	3	3	01	01	01	Rugby Cricket Athletics	041.txt 104.txt 019.txt	0,002s
20NewsGroup	20NG_400	400	4	4	01	01	01	Misc.forsale Comp.graphics Rec.autos Rec.motorcycles	76027.txt 38464.txt 103209.txt 104297.txt	0,16s
Dataset_3	Dataset_3	24	3	3	01	01	01	C1 C7 C4	article07.txt article07.txt article04.txt	0,001s

Le tableau **(**IV.2**)** résume les résultats des exécutions de notre approche sur certaines instances de corpus BBCNews, BBCSPORT, 20NewsGroup et DATASET3. Pour chaque expérimentation, nous avons exécuté l'algorithme 20 fois. Selon le tableau, le Ratio mode est toujours supérieur ou égal à 1, on déduit de cette observation que notre approche donne une

borne supérieure du nombre réel de classes ou elle a pu trouver le nombre exact de classes existant dans la littérature en un temps très limité.

4.4 Indices de validation de l'approche proposée

Afin de démontrer l'efficacité de notre approche, nous comparons KM qui utilise des centres aléatoires avec KM_MSSP qui s'initialise par kcentres obtenus par MSSP, en utilisant six mesures de validité : Mesure F, pureté, entropie, indice Xie-Beni, indice Fukuyama-Sugeno et le score d'Information Mutuelle Normalisée (INM).

La comparaison entre KM et KM_MSSP doit être effectuée sur le même nombre de clusters trouvé par MSSP et dans le même environnement en mesurant la distance en cosinus entre la représentation de l'élément donné et le centre du cluster. Mais la différence est que KM est basé sur le choix aléatoire des centres, alors que KM_MSSP s'initialise par les centres trouvés par MSSP.

4.3.1 Comparaison via la F-mesure

Comme on peut le voir sur la figure **(IV.5)** et le tableau **(IV.3)**, la F-mesure moyen de l'algorithme KM_MSSP est de 94% sur l'ensemble de données BBC_1328, 92% sur l'ensemble de données BBC_2225, 86% sur la dataset BBC_500, 81% sur BBC_1715, 68% sur BBC_Sport1 et 96% sur BBC_Sport2, ce qui est supérieur à la F-mesure moyen de l'algorithme KM de 81% sur BBC_1328, 75% sur l'ensemble de données BBC_2225 et 73% sur BBC_500. En effet, on observe une augmentation des valeurs de la F-mesure dans KM_MSSP par rapport à celles de KM dans tous les ensembles de données disponibles. Par conséquent, l'algorithme proposé K-Means initialisé par les centres obtenus par le stable maximum (KM_MSSP) permet une amélioration du K-Means au terme de l'indice F-mesure.

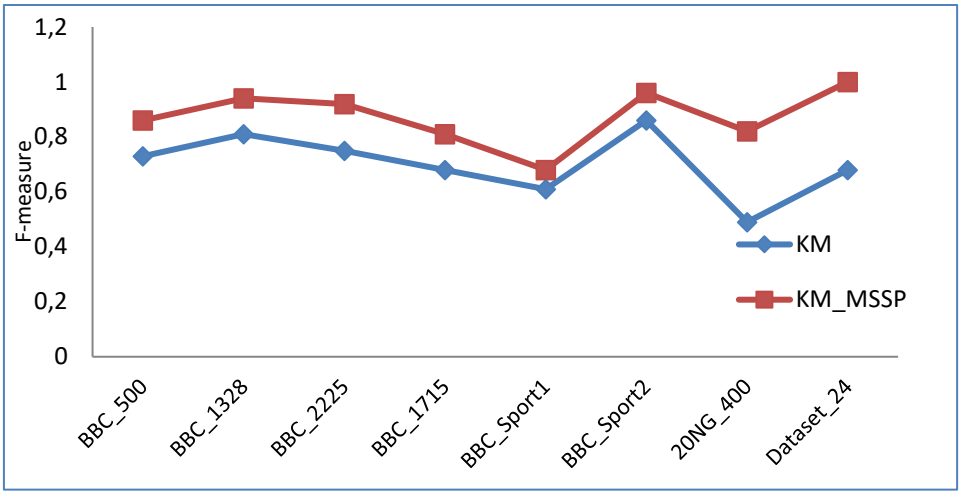


Figure IV.5 - Comparaison de F-mesure de KM et de l'algorithme KM_MSSP proposé.

Tableau IV.3 - Comparaison de la F-mesure moyenne du clustering sur un intervalle de confiance.

Dataset	KM	KM_MSSP
BBC_500	0,73±0,03	0,86±0,03
BBC_1328	0,81±0,01	0,94±0,03
BBC_2225	0,75±0,01	0,92±0,03
BBC_1715	0,68±0,02	0,81±0,03
BBC_Sport1	0,61±0,01	$0{,}68 \pm 10^{-16}$
BBC_Sport2	0,86±0,0005	$0{,}96 \pm 10^{-16}$
20NG_400	0,49±0,01	$0{,}82 \pm 10^{-16}$
Dataset_24	0,68±0,03	01 ± 10^{-16}

4.3.2 Comparaison via la pureté

Une comparaison de la pureté entre KM et KM_MSSP est présentée dans la figure **(IV.6)** et dans le tableau **(IV.4)**. Il apparaît que KM_MSSP surpasse KM en termes de pureté dans tous les ensembles de données disponibles. La pureté de KM_MSSP est de 94% sur l'ensemble de données BBC_1328, de 92% sur l'ensemble de données BBC_2225 et de 86% sur BBC_500, ce qui est supérieur à la pureté du KM de 81% sur BBC_1328, de 72% sur l'ensemble de données BBC_2225 et de 66% sur BBC_500.

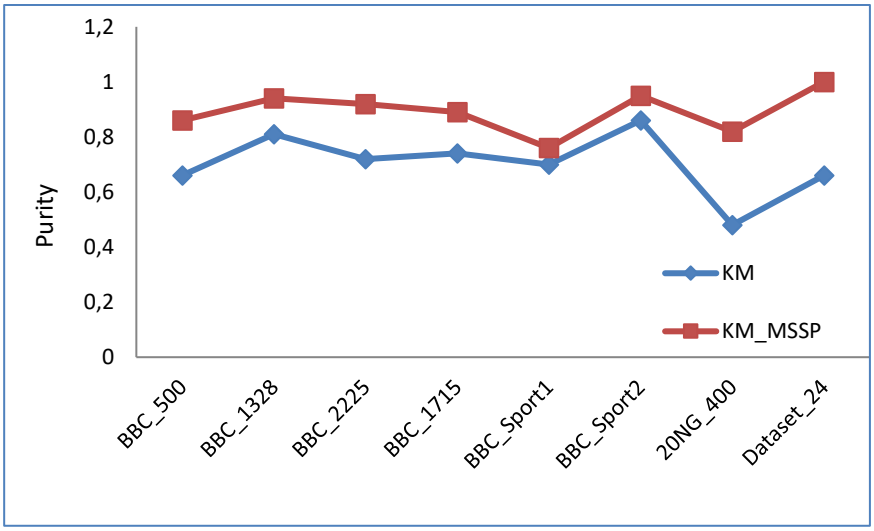


Figure IV.6 - Comparaison de la pureté de KM et de l'algorithme KM_MSSP proposé.

Tableau IV.4 -Comparaison de la pureté des classes par un intervalle de confiance.

Dataset	KM	KM_MSSP
BBC_500	0,66±0,02	0,86±0,01
BBC_1328	0,81±0,01	0,94±0,01
BBC_2225	0,72±0,005	0,92±0,01
BBC_1715	0,74±0,002	0,89±0,01
BBC_Sport1	0,7±0,005	$0{,}76 \pm 10^{-16}$
BBC_Sport2	0,86±0,01	$0{,}95 \pm 10^{-16}$
20NG_400	0,48±0,02	$0{,}82 \pm 10^{-16}$
Dataset_24	0,66±0,01	01 ± 10^{-16}

4.3.3 Comparaison via l'entropie

On observe également une baisse des valeurs de l'entropie (voir la figure **(IV.7)** et le tableau **(IV.5)**) dans l'approche proposée KM_MSSP par rapport à celles de KM. L'entropie de KM_MSSP est de 32%, ce qui est inférieur à l'entropie du cluster KM de 61% sur l'ensemble de données BBC_2225. Ainsi, la méthode KM_MSSP surpasse KM en termes d'entropie dans tous les ensembles de données disponibles.

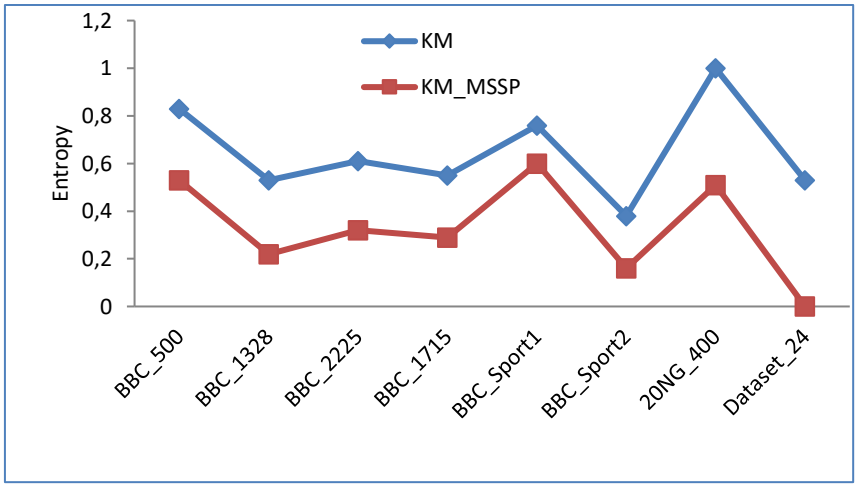


Figure IV.7 -**Comparaison de l'entropie de KM et de l'algorithme KM_MSSP**

Tableau IV.5 -Comparaison de l'entropie de clustering par un intervalle de confiance.

Dataset	KM	KM_MSSP
BBC_500	0,83±0,04	0,53±0,01
BBC_1328	0,53±0,02	0,22±0,02
BBC_2225	0,61±0,01	0,32±0,01
BBC_1715	0,55±0,01	0,29±0,01
BBC_Sport1	0,76±0,01	$0{,}6 \pm 10^{-16}$
BBC_Sport2	0,38±0,03	$0{,}16 \pm 10^{-16}$
20NG_400	01±0,01	$0{,}51 \pm 10^{-16}$
Dataset_24	0,53±0,03	00 ± 10^{-16}

4.3.4 Comparaison via le score de l'information mutuelle normalisée NMI

Une comparaison des NMI entre KM et KM_MSSP est présentée dans la figure **(IV.8)** et dans le tableau **(IV.6)**. Il apparaît que la méthode KM_MSSP surpasse la méthode KM en termes de NMI dans tous les ensembles de données disponibles. La NMI de KM_MSSP est de 78% pour les ensembles de données BBC_1328 et BBC_2225, et 63% pour BBC_500, et 85% pour BBC_Sport2, ce qui est supérieur à la NMI de la K-Means de 51% pour BBC_500 et pour l'ensemble de données BBC_1328, 77% pour BBC_2225, 64% pour BBC_Sport2.

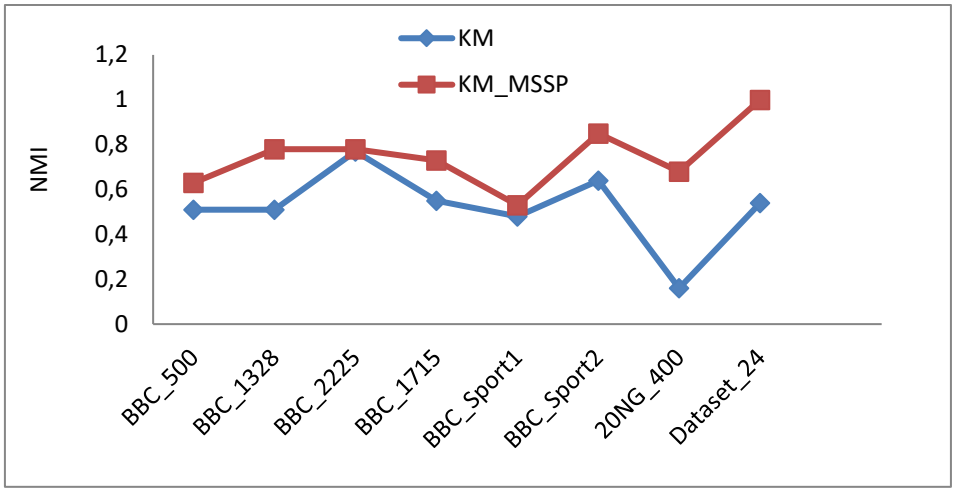


Figure IV.8 -Comparaison de la NMI de KM et de l'algorithme KM_MSSP proposé.

Tableau IV.6 -Comparaison de NMI entre KM et KM_MSSP sur un intervalle de confiance.

Dataset	KM	KM_MSSP
BBC_500	0,51±0,02	0,63±0,02
BBC_1328	0,51±0,01	0,78±0,03
BBC_2225	0,77±0,01	0,78±0,03
BBC_1715	0,55±0,03	0,73±0,03
BBC_Sport1	0,48±0,006	$0{,}53 \pm 10^{-16}$
BBC_Sport2	0,64±0,02	$0{,}85 \pm 10^{-16}$
20NG_400	0,16±0,03	$0{,}68 \pm 10^{-16}$
Dataset_24	0,54±0,01	01 ± 10^{-16}

4.3.5 Comparaison en termes de temps

Les comparaisons du temps CPU sur tous les ensembles de données sont présentées dans la figure **(IV.9)** et dans le tableau **(IV.7)**. D'après le graphique, notre approche KM_MSSP passe très peu de temps dans l'exécution sur tous les ensembles de données que KM.

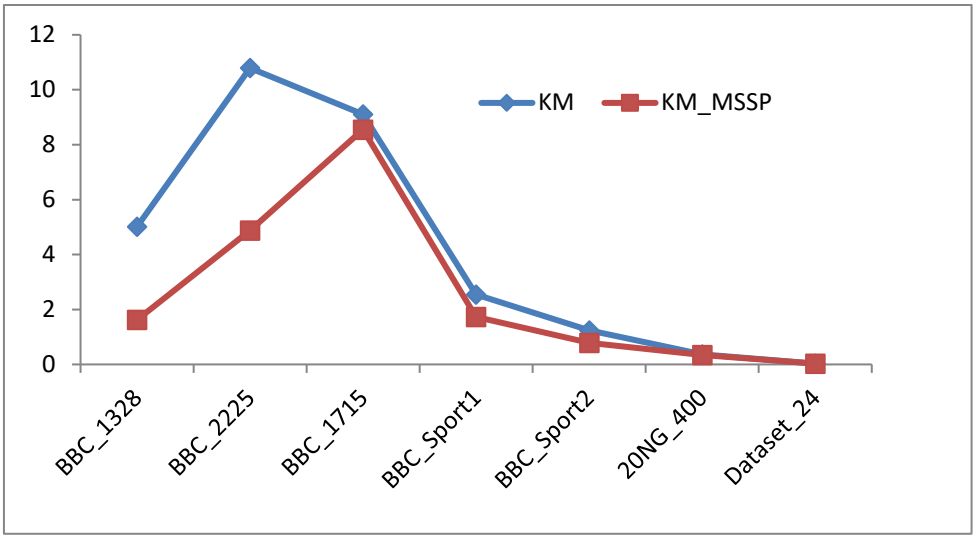


Figure IV.9 -Comparaison du temps de KM et de l'algorithme KM_MSSP proposé.

Tableau IV.7 - Comparaison au terme de temps CPU de KM et l'algorithme KM_MSSP proposé.

Dataset	KM	KM_MSSP
BBC_500	1,04	0,21
BBC_1328	5,01	1,63
BBC_2225	10,79	4,87
BBC_1715	9,11	8,55
BBC_Sport1	2,54	1,73
BBC_Sport2	1,24	0,78
20NG_400	0,36	0,34
Dataset_24	0,03	0,03

4.3.6 Comparaison via l'indice Xie-Beni

Une comparaison de l'indice Xie-Beni (XB) entre KM et KM_MSSP est présentée dans le tableau **(IV.8)**. Il apparaît que KM_MSSP surpasse KM en termes d'indice XB dans tous les ensembles de données disponibles. L'indice XB de KM_MSSP est de 3% pour l'ensemble de données BBC_1328, 7% pour l'ensemble de données BBC_2225 et 8% pour BBC_500, ce qui est inférieur à l'indice XB de KM de 45% pour BBC_1328, 47% pour l'ensemble de données BBC_2225 et 52% pour BBC_500.

Tableau IV.8 - **Comparaison de l'indice Xie-Beni de KM et de KM_MSSP**

Dataset	KM	KM_MSSP
BBC_500	0,52	0,08
BBC_1328	0,47	0,03
BBC_2225	0,45	0,07
BBC_1715	0,50	0,07
BBC_Sport1	0,53	0,09
BBC_Sport2	0,52	0,06
20NG_400	1,08	0,17
Dataset_24	0,50	0,03

4.3.7 Comparaison via l'indice de Fukuyama-Sugeno

Le tableau **(IV.9)** montre que les valeurs de l'indice de Fukuyama-Sugeno (FS) de KM_MSSP sont en baisse par rapport à celles de KM. L'indice FS du KM_MSSP est de 11 %, ce qui est inférieur à l'indice FS du cluster KM de 17 % sur l'ensemble de données_24. Ainsi, KM_MSSP surpasse KM en termes d'indice FS dans tous les ensembles de données disponibles.

Tableau IV.9 -Comparaison de l'indice de Fukuyama-Sugeno de KM et de KM_MSSP

Dataset	KM	KM_MSSP
BBC_500	0,87	0,42
BBC_1328	0,55	0,13
BBC_2225	0,75	0,4
BBC_1715	0,38	0,21
BBC_Sport1	0,20	0,16
BBC_Sport2	0,37	0,24
20NG_400	0,53	0,48
Dataset_24	0,17	0,11

Il est clairement observé que les résultats de KM_MSSP sont bien meilleurs pour tous les ensembles de données en termes d'entropie, de pureté, de F-mesure, du score NMI et de temps CPU, parce que KM_MSSP commence par un nombre de clusters (nombre de classes obtenues par MSSP) très proche ou égal au nombre réel de classes, et commence par les centres de classes initiales sélectionnées par MSSP (ce qui garantit un ensemble de documents indépendant). Mais

KM obtient les centres initiaux en utilisant une méthode aléatoire qui donne des résultats instables et trop d'itérations, ce qui affecte la qualité de clustering et coûte beaucoup de temps pendant l'exécution.

4.3.8 Comparaison entre KM_MSSP et une autre méthode déterminante

Afin de démontrer l'efficacité de notre approche, nous comparons notre approche avec celle du DSKM [Sherkat et Velcin, 2018] en termes d'information mutuelle normalisée sur le jeu de données BBC_2225.

Tableau IV.10 -Comparaison du score du NMI de clustering entre KM_MSSP et DSKM sur l'ensemble de données BBC_2225 et nombre de clusters égal au nombre trouvé par MSSP (k=6).

Méthodes	NMI
DSKM	0,681
KM_MSSP	0,78

D'après le tableau **(IV.10)**, la comparaison entre KM_MSSP et DSKM a été faite sur le même nombre de clusters trouvé par MSSP (6 pour l'ensemble de données BBC_2225 (voir tableau 2)). Le NMI de KM_MSSP est de 78%, ce qui est supérieur au NMI de DSKM qui est de 68%. Par conséquent, la performance de l'algorithme proposé est supérieure à celle de l'algorithme DSKM.

5 Conclusion

Ce chapitre a présenté ma principale contribution qui vise à développer une méthode permettant de déterminer automatiquement à la fois le nombre de clusters et les centres initiaux qui sont les paramètres de base de l'algorithme de K-Means ou d'autres méthodes de mise en cluster. Pour atteindre cet objectif et obtenir la meilleure qualité de mise en cluster, le problème de l'ensemble stable maximal (MSSP) est généralisé pour s'appliquer au clustering de documents textuels en utilisant le réseau de Hopfield continu (CHN). La méthode est indépendante de toute méthode de regroupement qui s'initialise par les k centres. Les résultats expérimentaux montrent que notre méthode peut effectivement trouver le nombre de clusters très proche ou égal au nombre réel de classes, ainsi qu'un ensemble correcte de centres, permet aussi d'obtenir de meilleurs résultats de mise en cluster en peu de temps, et qu'un grand nombre de documents peuvent être facilement traités.

CHAPITRE V : Classification des documents textuels : Synthèse

1 Introduction

La quantité de données générée par l'espèce humaine dans le monde entier augmente quotidiennement à un rythme exponentiel. C'est pourquoi la classification des données est devenue une nécessité. De nombreux chercheurs s'intéressent aujourd'hui au développement d'applications qui améliorent les méthodes de classification de textes. L'apprentissage profond a le potentiel de gérer et d'analyser cette grande quantité d'informations supervisées et non supervisées en peu de temps. En général, les systèmes de classification de textes peuvent être décomposés en plusieurs étapes :

- Extraction de caractéristiques : de nombreuses représentations se révèlent être d'un grand avantage dans diverses tâches de traitement automatique du langage naturel (sac de mots (BOW), TF-IDF en anglais "TermFrequency-InverseDocument Frequency", TermFrequency (TF), Word2Vec, et GloVe, BERT base).
- Réduction de dimension : de nombreux chercheurs préfèrent utiliser la réduction de dimension pour réduire le temps et la complexité de la mémoire dans leurs applications. L'utilisation de la réduction de la dimension pour le prétraitement pourrait être plus efficace que le développement de classificateurs peu coûteux.
- La sélection des classificateurs est une étape importante et peut influencer directement sur les résultats de classification. Récemment, les classificateurs d'apprentissage profond ont surpassé de nombreux classificateurs de l'apprentissage automatique précédents dans le domaine du langage naturel. Le succès de ces algorithmes d'apprentissage profond est basé sur leur capacité à modéliser des relations non linéaires et complexes au sein des données.
- Évaluation des résultats de la classification : Il existe plusieurs indices pour évaluer les méthodes de classification. Le calcul de la précision est la méthode d'évaluation la plus simple. Dans cette étude, nous utilisons le score F1, qui est l'une des mesures d'évaluation agrégées les plus populaires pour l'évaluation des classificateurs.

Les techniques traditionnelles du traitement du langage naturel (TLN) représentent généralement les mots comme des indices dans un vocabulaire (BOW) ne donnant aucune

notion de la relation entre les mots, et cela représente plusieurs limites pour cette représentation : certains mots sont polysémiques ("Théâtre" se réfère à l'art, le lieu, la production), d'autres peuvent être des synonymes et être traité différemment ("aimer" a le même sens que désirer, admirer, chérir), d'autres encore sont fortement liés sémantiquement sans que cela soit pris en compte dans la représentation ("chevalet" est très lié à "frette") et enfin, certains mots perdent leur sens s'ils sont extraits de leur groupe nominal (exemple : "poste de police" n'a pas le même sens que "poste" et "police" pris séparément). A ces limites s'ajoute celle du nombre trop important de mots représentant les documents qu'on appelle la malédiction de la dimension. En réponses à ces limites, et avec l'explosion des données textuelles sur le Web et le développement plus rapide des technologies de réseau neuronal profond, le plongement de mots distribué a été efficacement formé et largement développé et utilisé dans de nombreuses tâches de text mining [Bengio et al., 2003], [Mnih et Hinton, 2007], [Collobert et al., 2011], [Mikolov et al., 2013b], ces plongements sont souvent pré-entraînés sur des corpus de textes à partir de statistiques de co-occurrence, ainsi il n'est pas possible de détecter la signification du mot à partir du texte. Ce qui a motivé et a poussé la recherche vers une nouvelle technique de représentation des mots, où les vecteurs de mots dépendent du contexte du mot, appelé "Représentations de mots contextuelles profonds" ou "word embedding contextuelle", comme BERT [Devlin et al. 2019], qui a connu un succès retentissant dans une variété de tâches de TALN.

Dans ce chapitre, nous examinons l'impact de nombreuses représentations de mots (BOW, plongement de mots (GloVe, Word2Vec) et plongement de mots contextuelle) ainsi que des approches de classification (apprentissage profond par rapport aux méthodes traditionnelles d'apprentissage automatique) sur la réalisation de classification de textes [Karim et al., 2021].

Ce chapitre est organisé comme suit : nous commençons par donner un tour d'horizon sur les travaux relatifs à l'incorporation de mots et aux algorithmes de classification dans la section 2. Ensuite, la section 3 décrit le processus de classification du texte. A la fin, nous présentons les résultats de la comparaison de différentes représentations de texte et de plongement de mots avec différentes tâches de classification de texte dans la section 4.

2 État de l'art

Les techniques traditionnelles de TALN ont reposé presque exclusivement sur l'approche du sac de mots. Certaines études existantes évaluent quantitativement l'incorporation des mots dans la représentation de la sémantique des mots. L'incorporation de texte était un problème

plus difficile que l'incorporation de mots en raison de la variance des phrases, des expressions et du texte. La génération de l'incorporation des mots nécessite beaucoup de puissance de calcul, de prétraitement et de temps d'apprentissage [Rezaeinia et al., 2017].

En apprenant la prédiction basée sur le contexte, les méthodes d'incorporation de mots font correspondre chacune de ces vecteurs à des représentations textuelles, dont les éléments capturent la sémantique latente des données linguistiques. Mais les hypothèses et les simplifications que l'approche par sac de mots implique, comme la perte de la structure grammaticale ou du sens des mots, selon le contexte, ont été soulignées à maintes reprises [Grimmer et Stewart, 2013], [Lowe et Benoit, 2013].

Les méthodes d'incorporation de texte non supervisées se heurtent au problème de l'attribution d'une importance aux mots lors du calcul de l'intégration. Il existe de nombreuses méthodes différentes pour apprendre l'incorporation des mots à partir d'un corpus. De nombreuses études utilisent des applications d'apprentissage automatique [Van Atteveldt et al., 2008], [Bursher et al., 2014], [Ceron et al., 2015], [Ceron et al., 2017], [Hopkins & King, 2010].

L'importance des mots détermine dans quelle mesure l'intégration du texte doit être biaisée en faveur des mots les plus représentatifs. De Boom et al. [De Boom et al., 2016] ont introduit une méthode permettant d'attribuer une importance aux mots en fonction de leur score TF-IDF dans le texte.

2.1 Représentations d'incorporation de mots

Il s'agit d'une représentation distribuée de textes qui pourrait être l'une des percées clés pour la performance impressionnante des méthodes d'apprentissage approfondi aux problèmes difficiles de traitement du langage naturel. Dans ce contexte, [Levy et al., 2015] ont proposé l'incorporation de mots, qui a constitué une révolution majeure dans le domaine de l'exploration des données.

L'incorporation de mots la plus couramment utilisée comprend Word2Vec [Mikolov et al., 2013a], [Mikolov et al., 2013b], GloVe [Pennington et al., 2014] et fastText [Joulin et al., 2016].

2.1.1 Représentation GloVe

GloVe [Pennington et al., 2014] est un algorithme d'apprentissage non supervisé permettant d'obtenir des représentations vectorielles de mots. L'apprentissage est effectué sur des

statistiques globales agrégées de co-occurrence mot à mot à partir d'un corpus, et les représentations résultantes présentent des sous-structures linéaires intéressantes de l'espace vectoriel des mots. Nous pouvons prendre quelques notes.

Soit X la matrice de co-occurrence mot-mot, dont les entrées X_{ij} indiquent le nombre total de fois où le mot j apparaît dans le contexte du mot i. Soit $X_i = k \times X_{ik}$ le nombre de fois où un mot apparaît dans le contexte du mot i. Enfin, soit $P_{ij} = P(j/i) = X_{ij}/X_i$ la probabilité que le mot j apparaisse dans le contexte du mot i.

2.1.2 Représentation Word2Vec

Word2vec est une représentation de texte qui prend un corpus de texte en entrée et produit des vecteurs de mots en sortie. Il génère un vecteur de mots par deux algorithmes d'apprentissage : un sac de mots continu (CBOW) et un skip-gram [Mikolov et al., 2013a ; Mikolov et al., 2013b]. Dans la méthode CBOW, l'objectif est de prédire un mot en fonction des mots voisins, tandis que dans le skip-gram, un mot unique, une fenêtre ou un contexte de mots sont prédits. Les deux algorithmes apprennent la représentation d'un mot qui est utile pour prédire d'autres mots dans la phrase. En dehors du choix de l'architecture skip-gram ou CBOW, word2vec possède plusieurs paramètres, dont la taille de la fenêtre de contexte, la dimension du vecteur, qui affectent la vitesse et la qualité de l'apprentissage.

En tant que limitation, plusieurs problèmes persistent tels que la question de la conservation du sens entier des documents avec un sens cohérent pour l'apprentissage automatique. Ces incorporations de mots (word2vec, GloVe) sont souvent pré-entraînés sur des corpus de textes à partir de statistiques de co-occurrence. Par conséquent, étant donné un mot, son encastrement est toujours le même dans n'importe quelle phrase où il apparaît. Dans ce cas, l'intégration des mots pré-entraînés est statique. Dans ce but, les recherches se concentrent sur la formation de représentations contextuelles sur des corpus de textes.

2.1.3 Représentation contextuelle

Une nouvelle méthode a été introduite récemment concernant la représentation des mots. Dans cette méthode, le contexte du mot est celui sur lequel s'appuie le vecteur mot. C'est ce qu'on appelle les "représentations contextuelles". L'idée principale de cette méthode est que le contexte qui entoure le mot est capturé. Dans les incorporations classiques, un mot donné était attribué à une seule représentation. Les principales caractéristiques et limites :

Parmi les principales caractéristiques de cette technique figure sa capacité à capturer le sens du mot à partir du texte donné, y compris la gestion de la polysémie et l'incorporation du contexte.

D'un autre côté, ses limites incluent son incapacité à travailler au niveau du mot individuel car elle travaille au niveau du document et de la phrase. D'autres limitations concernent la consommation de mémoire pour le stockage, ce qui est plus coûteux si nous prenons en considération d'autres techniques.

2.2 Classifieurs basés sur l'apprentissage profond

Dans la section suivante, nous abordons des architectures neuronales plus profondes qui ont été développées pour générer ces incorporations et pour réaliser des tâches de classification de texte [Kim, 2014], certains sont basés sur les réseaux neuronaux, tels que DNN [El Harti et Boumhidi, 2018], CNN, RNN, et certaines de ces architectures incluent des informations séquentielles de texte, telles que les LSTM [Palangi et al., 2016], BERT [Devlin et al., 2018] et XLNET [Yang et al., 2019]. Nous les avons comparé à une approche statistique comme base de référence, en utilisant un classificateur Machine à vecteurs de support SVM (Vapnik, 2013).

Les modèles d'apprentissage approfondi ont permis d'obtenir des résultats de pointe dans de nombreux domaines, notamment une grande variété d'applications de TALN. L'apprentissage approfondi pour la classification des textes et des documents comprend trois architectures de base d'apprentissage approfondi en parallèle. Nous décrivons chaque modèle en détail ci-dessous.

2.2.1 Réseaux neuronaux artificiels (ANN)

Ce sont des réseaux informatiques d'inspiration biologique. Plus précisément, les modèles ANNs simulent l'activité électrique du cerveau et du système nerveux. Ils sont capables de modéliser et de traiter, en parallèle, des relations non linéaires entre les entrées et les sorties. En général, les réseaux neuronaux sont constitués de nœuds (neurones) qui fonctionnent simultanément, et s'organisant en une couche, la sortie d'une couche servant d'entrée à la couche suivante pour former le réseau.

2.2.2 Réseaux neuronaux profonds (DNN)

Il s'agit en fait d'un terme qui décrit certains types de réseaux neuronaux et d'algorithmes relatifs qui consomment souvent des données brutes. Il s'agit du réseau neuronal avec trois types de couches : les couches d'entrée, les couches cachées et les couches de sortie. La différence entre

les différents modèles de DNN réside dans la manière dont ils sont connectés. L'extraction non supervisée de caractéristiques est également un domaine où l'apprentissage approfondi excelle. Dans ce travail, nous utilisons à la fois les modèles de réseaux neuronaux à convolution (CNN) et de réseaux neuronaux récurrents (RNN) pour la tâche de classification des textes. La couche d'entrée peut être construite via le score TF-IDF, le plongement de mots ou une autre méthode d'extraction de caractéristiques. La couche de sortie est égale au nombre de classes pour la classification multi-classes ou à une seule classe pour la classification binaire.

2.2.3 Réseaux neuronaux à convolution (CNN)

Les réseaux neuronaux à convolution (CNN) sont des modèles spéciaux d'apprentissage profond pour le traitement des données. Ils sont construits pour trouver des relations entre des éléments de données en fonction de leur position relative. CNN a une excellente capacité d'analyse de données séquentielles, telles que le traitement du langage naturel [Zhang et al., 2015]. Son utilisation dans la classification des textes a donné de brillants résultats dans les données textuelles également [Kim, 2014]. CNN a généralement contenu deux opérations de base, à savoir la convolution et la mise en commun. L'opération de convolution utilisant plusieurs filtres est capable d'extraire des caractéristiques de l'ensemble de données. Ensuite, l'opération de mise en commun est utilisée pour réduire la dimensionnalité des cartes d'éléments.

2.2.4 Réseaux neuronaux récurrents (RNN)

Il s'agit d'un type de réseau neuronal puissant et robuste, qui utilise des données séquentielles ou des données qui changent au fil du temps, comme les données textuelles. La séquence elle-même contient des informations, et les réseaux récurrents les utilisent pour effectuer des tâches de classification. Les RNNs peuvent traiter les informations de manière bidirectionnelle afin de permettre l'apprentissage des informations des états précédents ainsi que des états suivants. Trouver des relations entre un élément (par exemple un mot) et ce qui le précède et ce qui le suit, c'est le principe de base du RNN. Ces algorithmes d'apprentissage profond sont couramment utilisés dans le traitement du langage naturel (NLP) car leur structure est très adaptée au traitement de textes de longueur variable comme dans [Zhang et Byron, 2015].

2.2.5 Mémoire à long et à court terme (LSTM)

La mémoire à long terme et à court terme (LSTM) est un type spécial de RNN qui préserve la dépendance à long terme de manière plus efficace que la RNN de base. Elle est particulièrement utile pour surmonter le problème du gradient de fuite. Bien que la structure de la LSTM soit

similaire à celle du RNN, les réseaux LSTM possèdent une porte d'oubli ainsi qu'une porte de mise à jour. Comme leur nom l'indique, les portes d'oubli et de mise à jour déterminent s'il faut transmettre les informations actuelles ou les rejeter.

2.2.6 Réseaux neuronaux convolutifs récurrents (RCNN)

Une autre technique de classification des textes qui proviennent de la combinaison des architectures d'apprentissage profond est appelé réseaux neuronaux convolutifs récurrents. Les RCNNs sont également utilisés pour la classification des textes [Lai et al., 2015] et [Wang et al.,2017]. L'idée principale de cette technique est de capturer des informations contextuelles avec la structure récurrente et de construire la représentation du texte en utilisant un réseau neuronal convolutif [Lai et al., 2015]. Cette architecture est une combinaison de RNN et CNN pour utiliser les avantages des deux techniques dans un même modèle.

2.2.7 Réseau DBN (Deep Belief Network)

Le réseau DBN est un modèle graphique génératif, constitué de plusieurs couches de variables latentes. Il est composé de plusieurs réseaux peu profonds tels que les machines Boltzmann restreintes, de sorte que la couche cachée de chaque sous-réseau sert de couche visible du sous-réseau suivant.

2.2.8 BERT

L'apprentissage par transfert a été présenté par Google en 2018 dans le papier “Attention is All You Need” qui s'est avéré être un progrès décisif dans le domaine de la PNL. L’apprentissage par transfert est au cœur des modèles de langage tels que les incorporations à partir de modèles de langage (ELMo) et les représentations d'encodeur bidirectionnel à partir de transformateurs (BERT). Ces modèles ne traitent pas une séquence d'entrée jeton par jeton mais prennent plutôt la séquence entière comme entrée d'un seul coup, ce qui constitue une grande amélioration par rapport aux modèles basés sur les RNN. Nous n'avons pas besoin de données étiquetées pour pré-entraîner les modèles basés sur un transformateur. Nous pouvons utiliser ce modèle pré-entraîné pour d'autres tâches de PNL.

BERT et GPT-2 sont les modèles à base de transformateurs (Figure V.1) les plus populaires. Dans ce chapitre, nous nous concentrerons sur BERT en le comparant avec d’autres techniques de classification de textes.

BERT est l'acronyme de Bidirectional Encoder Representations from Transformers, est un document récent publié par les chercheurs de Google AI Language [Devlin et al. 2019]. Il est

basé sur l'architecture des transformateurs (une explication détaillée peut être trouvée dans l'article original [Vaswani et al., 2017] ou dans [Alammar, 2019]. BERT est une nouvelle méthode de pré-entrainement des représentations linguistiques qui permet d'obtenir des résultats de pointe sur un large éventail de tâches de PNL y compris la classification de textes, la reconnaissance d'entités nommées, la génération de textes, etc**.** Ses spécificités : son incorporation de mot positionnelle basé sur le contexte et ses tâches de pré-entrainement.

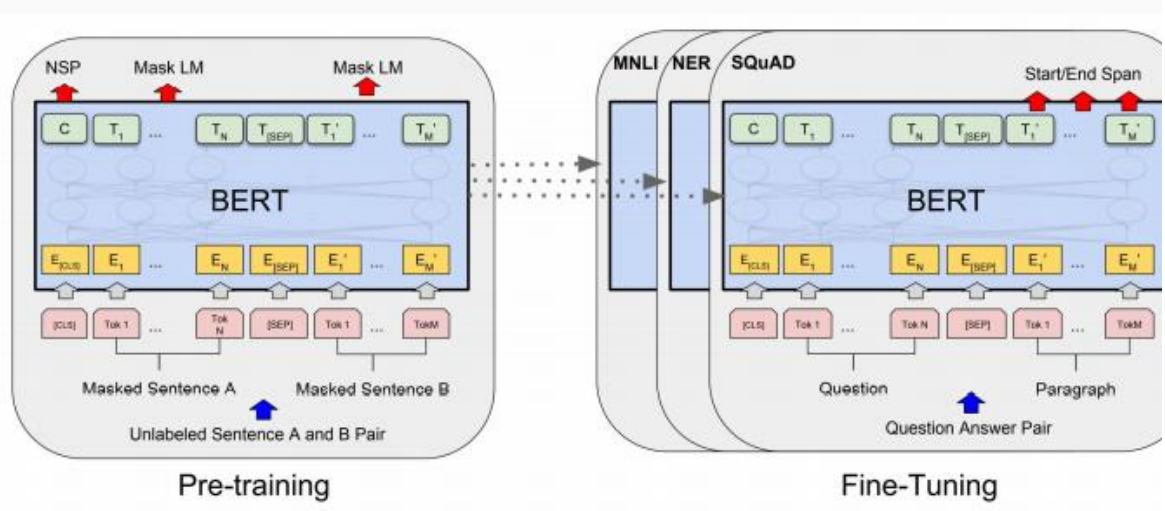


Figure V.2 -Représentation BERT

2.3 Comparaison des techniques de classification des textes

Le tableau (V.1) ci-dessous, qui est adopté de [Kowsari et al., 2019], compare les méthodes de classification de texte en tenant compte de plusieurs critères tels que la citation, l'extraction de caractéristiques et l'architecture. D'autres critères concernent également le jeu de données, les détails, la mesure de validation et les limites de chaque méthode. Il convient de mentionner que chaque méthode de classification de texte implique un modèle qui est le classificateur, nécessite également une technique d'extraction de caractéristiques qui consiste à convertir un ensemble de données d'un texte ou de documents en données numérique.

TABLE V-1. Comparaison des techniques de classification des textes

Référence	**Architecture**	**Extraction de caractéristiques**	**Détails**	**Dataset**	**Mesure de validation**	**Limitations**
[Jiang et al., 2018]	Réseau à croyance profonde	DBN	DBN complète la fonction apprenant à résoudre le problème de la dimension élevée et de la matrice creuse et la régression softmax est utilisée pour classer les textes	Reuters-21578 20-Newsgroup	Taux d'erreur	Le calcul est coûteux et l'interprétabilité du modèle reste un problème de ce type de modèle
[Kowsari et al., 2018]	Algorithme d'apprentissage profond (CNN, DNN et RNN)	TF-IDF et GloVe	Apprentissage approfondi multi modèle aléatoire (RDML)	IMDB review, Reuters-21578 20NewsGroup, et WOS	Précision	Le calcul est coûteux
[Kowsari et al., 2017]	Structure hiérarchique	TF-IDF et GloVe	Apprentissage hiérarchique approfondi pour la classification des textes (HDLTex)	Web of sciencedata set	Précision	Ne fonctionne que pour les ensembles de données hiérarchiques
[Yang et al., 2016]	Hierarchical Attention Networks	Word Embedding	Deux niveaux de mécanismes d'attention appliqués au niveau des mots et des phrases.	Yelp, IMDB revue, et Amazon revue	Précision	Ne fonctionne qu'au niveau des documents
[Lodhi et al., 2002]	SVM	Utilisation de la similarité TF-IDF	Le noyau est un produit interne dans l'espace de caractéristiques généré par toutes les sous-séquences de longueur k	Reuters-21578	F1-Macro	Le manque de transparence des résultats
[Zhang et al., 2015]	CNN	Caractères encodés	Le ConvNet au niveau des caractères contient 6 couches convolutives et 3 couches entièrement connectées	Yelp, Amazon et l'ensemble des données de Yahoo ! Answers	Erreurs relatives	Ce modèle est uniquement conçu pour découvrir les caractéristiques de leurs entrées qui ne varient pas en fonction de la position

3 Résultats et discussions

Pour améliorer la qualité de la classification des textes, il est nécessaire de procéder par l'amélioration de ses représentations. Plusieurs tâches de prétraitement doivent être effectuées avant d'appliquer les algorithmes de classification. Nous présentons ici les tâches de prétraitement qui peuvent être utilisées seules ou combinées.

Chaque texte est représenté dans le modèle d'espace vectoriel qui est un modèle algébrique permettant de représenter les documents textuels comme des vecteurs de termes. Ensuite, la liste des mots d'arrêt et les signes de ponctuation doivent être supprimés. La lemmatisation, qui est le processus de réduction des mots à leur forme de tige ou de racine, est effectuée, et un filtre de prétraitement a déjà été appliqué aux données pour éliminer les termes qui ont une faible fréquence (nombre d'occurrence < 3). La lemmatisation est un processus qui remplace le suffixe d'un mot par un autre ou supprime complètement le suffixe d'un mot pour obtenir la forme de base du mot (lemme).

Après la phase de prétraitement, il a fallu réfléchir à la manière de représenter le texte. De nombreuses représentations se révèlent très avantageuses dans les différentes tâches du traitement automatique du langage naturel (TALN). Il existe plusieurs représentations différentes, en commençant par le modèle classique du sac de mots (BOW), puis les méthodes de réduction de la dimensionnalité telles que l'analyse en composantes principales (PCA), l'analyse discriminante linéaire (LDA), et enfin la représentation compacte d'incorporation de mots comme Glove, Word2Vec et se terminant par l'intégration contextuelle de BERT.

Afin d'établir une comparaison équitable avec les méthodes de classification, nous utilisons un ensemble de données accessibles au public, 20NewsGroup qui a été assemblé par [Lang, 1995], pour entraîner l'incorporation de mots dans le cadre des expérimentations. Nous appliquons d'abord les étapes de prétraitement décrites dans la section 3, puis nous représentons mathématiquement le poids du terme dans un document par le score TF-IDF.

Pour notre analyse, nous avons considéré des modèles d'incorporation, pré-entrainés sur des données du domaine public telles que Wikipedia, et nous les avons comparé à la représentation du sac de mots (BoW). En outre, tous les modèles ont été construits en utilisant les paramètres suivants (Tableau V. 2).

TABLE V-2. Évaluation des classificateurs et des paramètres correspondants.

	Fonction d'activation	Couches	Dropout/Learning rate	Optimizer	Époques	Taille des lots
RCNN	Relu+softmax	2	0.25/0.001	Adam	15	128
DNN	Relu+softmax	4	0.5/0.001	Adam	10	128
CNN	Relu+softmax	5	0.25/0.001	Adam	15	128
LSTM	Relu+softmax	3	0.25/0.001	Adam	10	128
BERT		12	0.1/ 2e-5	Adam	4	32

La base BERT a les paramètres suivants: L = 12, H = 768, A = 12 où L est le nombre d'encodeurs empilés, H est la taille cachée et A est le nombre de têtes dans les couches d'attention multi-têtes. Nous avons fixé le nombre d'épocs à quatre.

Nous avons modifié la dimension des représentations distribuées par mots, et nous observons que l'augmentation de la dimension peut améliorer la précision de la classification lorsque la dimension est inférieure à 50, mais lorsque la dimension est supérieure à 50, l'amélioration des performances de classification n'est pas si importante, mais le temps de pré-entrainement augmentera de manière significative (Fig.V.2), nous choisissons 50 comme dimension finale dans les expériences suivantes.

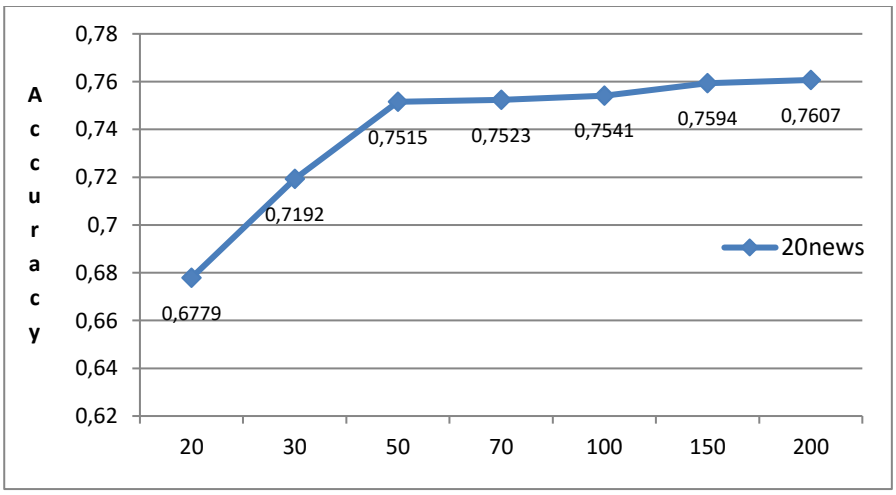


Figure V.3. Effets de la dimension de la représentation distribuée des mots sur la précision

Pour chaque étape, nous avons mesuré l'impact en évaluant la meilleure combinaison possible, sur la base du score F1 moyen (score moyen pondéré dérivé de 10 fois les résultats de validation croisée).

TABLE V-3. Les valeurs de score F1 de diverses représentations de texte dans diverses méthodes de classification sur l'ensemble de données 20NewsGroup.

Method	Extraction de caractéristiques	F1-score	Time
SVM	TFIDF	**0.85**	~ seconds
BERT fine-tuned classification	BERT base	**0.82**	~1/2 hour
DNN	TFIDF	0.81	03min
LSTM	Word2vec - CBOW	0.76	35min
LSTM	GloVe	0.75	30min
LSTM	BOW	0.67	5min
RCNN	Word2vec -CBOW	0.76	18min
RCNN	GloVe	0.75	26min
CNN	Word2vec-CBOW	0.75	31 min
CNN	GloVe	0.74	25 min

D’après les résultats du tableau **V.3**, nous pouvons conclure que:

- L'utilisation de plongement de mots pré-entraînée conduit à de meilleures performances avec tout algorithme de classification testé. Par exemple, nous observons que LSTM avec Glove ou Word2vec surpasse le LSTM avec BOW.
- GloVe a une valeur de F1 score relativement plus faible que Word2vec, ce qui indique que la qualité des représentations distribuées par GloVe est inférieure à celle des autres modèles de représentations sémantiques de mots.
- BERT produit une bonne performance en seulement 4 époques et passe peu de temps. Nous observons que le modèle BERT pré-entraîné par la base BERT surpasse toutes les autres méthodes de classification basées sur le réseau neuronal profond, et surpasse les autres incorporations de mots en termes de score F1.

Nous pouvons conclure que les incorporations de mots (word2vec, GloVe) sont souvent pré-entraînés sur des corpus de textes à partir de statistiques de co-occurrence. Par conséquent, étant donné un mot, son encastrement est toujours le même dans n'importe quelle phrase où il

apparaît. Dans ce cas, l'intégration des mots pré-entraînés est statique. Ainsi, ils ne peuvent pas détecter le sens du mot à partir du texte (ils ne peuvent pas capturer la polysémie).

Par contre les représentations contextuelles basées sur des modèles de transformation, fonctionnent sur le mécanisme d'attention (ELMO, BERT), et l'attention est une façon de regarder la relation entre un mot et ses voisins), en considérant la séquence de tous les mots dans les documents. Ainsi, étant donné un mot, les encastrements sont générés dynamiquement à partir d'un modèle pré-entraîné (ou finement ajusté). Pour cette raison, BERT obtient des résultats de pointe sur la classification de textes et d'autres tâches de traitement du langage naturel.

4 Conclusion

Récemment, les classificateurs d'apprentissage profond ont surpassé de nombreux classificateurs d'apprentissage automatique antérieurs en NLP. Le succès de ces algorithmes d'apprentissage profond repose sur leur capacité à modéliser des relations non linéaires et complexes au sein des données. Dans ce chapitre, nous avons exploré les performances de différentes approches de représentation des mots (comparant des sacs de mots à des plongements traditionnels et contextuels formés sur des corpus généraux) ainsi que des algorithmes de classification basés sur l'apprentissage profond. Les résultats ont montré que le plongement de mots (GloVe, Word2Vec avec SkipGram ou CBOW) surpasse les autres méthodes de représentation de texte comme BOW. Nous concluons également que le modèle BERT surpasse toutes les autres méthodes de classification basées sur l’apprentissage profond, et que le modèle BERT n'a pas permis d'améliorer la F1-score obtenue avec un simple SVM.

CONCLUSION GENERALE ET PERSPECTIVES

Dans ce chapitre, je conclus en présentant les différentes contributions dans la première et la deuxième section puis je propose quelques perspectives dans une troisième.

Au cours de ce travail de thèse, nous avons essayé de résoudre la problématique axée sur l'amélioration de la qualité du clustering. Nous avons proposé deux contributions pour la classification non supervisée de textes.

La première avait comme objectif la détection automatique du nombre de clusters au sein d'un grand corpus textuel. L'approche consiste à modéliser le problème de mise en cluster des documents textuels sous la forme du problème de l'ensemble stable maximum dans un graphe non orienté, et le résoudre par réseau CHN.

Ma seconde contribution est une suite logique de la première. Nous évaluons notre première contribution qui réalisait de bons résultats dans la détermination du nombre de clusters (= taille de l'ensemble stable maximum résultant), mais on n'a pas considéré les nœuds obtenus par l'ensemble stable maximum résultant, ce qui nous a motivé à les considérer comme les centres initiaux dans les algorithmes de mise en cluster qui s'initialisent souvent par des centres aléatoires (méthodes de partitionnement). L'algorithme proposé est exécuté avant l'algorithme de clustering, ce qui signifie qu'il est indépendant de toute méthode de clustering qui commence par k centres.

Pour démontrer l'efficacité de notre approche et la qualité de la classification obtenue, nous avons comparé l'algorithme de K-Means classique qui prends comme paramètres k=nombre de clusters trouvé par notre approche, et des centres aléatoires, avec K-Means initialisé par nombre de clusters et centres trouvés par notre approche (KM_MSSP), en terme de temps et de plusieurs indices comme: la F-mesure, la pureté, l'entropie, l'indice de Xie-Beni, l'indice de Fukuyama-Sugeno et le score de l'information mutuelle normalisée. Nous avons conclu que KM_MSSP surperforme K-Means standart et autre méthode déterministe et qu'il est efficace pour les grands ensembles de données et qu'il est très optimisé en termes de temps.

La dernière contribution concerne une étude comparative pour examiner l'impact de nombreuse représentation de mots (BOW, plongement de mots (GloVe, Word2Vec) et plongement de mots contextuelle) ainsi que des approches de classification (apprentissage profond par rapport aux

méthodes traditionnelles d'apprentissage automatique) sur la réalisation de classification de textes. Nous concluons d'après les résultats obtenus que le modèle BERT surpasse toutes les autres méthodes de classification basées sur l'apprentissage profond.

Plusieurs pistes de travail se dégagent des travaux présentés dans ce mémoire :

Tout d'abord, le domaine de la théorie de l'ensemble stable dans un graphe est encore ouvert à de nombreux défis qui offrent des possibilités d'amélioration future dans le problème du regroupement des documents. Les travaux futurs comprennent :

Dans la phase de prétraitement, nous envisageons d'améliorer la représentation du texte par l'incorporation de texte contextuelle.

Améliorer le réseau de Hopfield continu en le combinant avec des approches méta-heuristiques plus efficaces, par exemple : algorithmes génétiques, recherche de tabou, pour obtenir un minimum global.

D'autre perspective ambitieuse de ce travail sera l'implémentation réelle des méthodes proposées dans d'autres applications de la fouille de textes tels que la traduction automatique de texte, le résumé automatique de texte, système de Question Réponse, recherche d'information. Ils sont directement utilisables ou généralisables dans d'autres domaines d'application.

BIBLIOGRAPHIE

[Aggarwal et Zhai, 2012] C. C. Aggarwal and C. Zhai. An Introduction to Text Mining. In Mining Text Data, ed: Springer, Boston, MA, 2012, pp. 1-10.

[Aggarwal et Reddy, 2013] C. C. Aggarwal, and C. K. Reddy. DATA CLUSTERING Algorithms and Applications. Data Mining and Knowledge Discovery Series, 2013.

[Ahmad et Dey, 2007] A. Ahmad, L. Dey. A k-mean clustering algorithm for mixed numeric and categorical data. Data & Knowledge Engineering. 2007, 63 (2), 503–527.

[Aiyer et al., 1990] S. V. B. Aiyer, M. Niranjan, and F. Fall-side. A theoretical investigation into the performance of the Hopfield model. IEEE Trans. Neural Networks, 1990, vol. 1, pp. 204-215.

[Aiyer, 1991] S. V. B. Aiyer. Solving combinatorial optimization problems using neural networks. Technical report CUED/F.INFENG/TR 89, Engineering Department, Cambridge University, Cambridge, UK, 1991.

[Alammar, 2019] J. Alammar. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning). http://jalammar.github.io. 2019.

[Aliguliyev, 2009] M. R. Aliguliyev. Clustering of document collection, A weighting approach. Expert Systems with Applications, 2009, 36, (4), 7904-7916.

[Arora et Safra, 1992] S. Arora, S. Safra. Probabilistic Checking of Proofs; a new Characterization of NP. In Proceedings 33rd IEEE Symposium on Foundations of Computer Science, 1992, pp 2–13. IEEE Computer Society, Los Angeles.

[Ashour et Fyfe, 2014] W. Ashour, C. Fyfe. Improving Bregman K-Means. International Journal of Data Mining, Modelling and Management, 2014, 6 (1), 65-82.

[Bengio et al., 2003] Y. Bengio, J. Ducharme, P. Vincent and C. Janvin. A neural probabilistic language model. March 2003.

[Besançon et al., 2001] R. Besançon, A. Rozenknop, J-C. Chappelier et M. Rajman.Intégration probabiliste de sens dans la représentation de texte.TALN 2001, Tours, 2-5 juillet 2001. Laboratoire d'Intelligence Artificielle, Département Informatique - École Polytechnique Fédérale de Lausanne.

[Bezdek et Hathaway, 1994] J. C. Bezdek, and R. J. Hathaway. Optimization of fuzzy clustering criteria using genetic algorithms. In Proc. First IEEE Conf. Evolutionary Computation, Piscataway, NJ: IEEE Press, 1994, Vol. 2, pp. 589-594.

[Biernat et Lutz, 2016] E. Biernat and M. Lutz. Data Science : fondamentaux et études de cas. In chapter Book, Eyrolles, 2016.

[Bomze et al., 1999] I. M. Bomze, M. Budinich, P. M. Pardalos, M. Pelillo. The Maximum Clique Problem. Handbook of Combinatorial Optimization, 1999, Kluwer, Boston.

[Bonomo et al., 2005] F. Bonomo, G. Durán, M. C. Lin, J. L. Szwarcfiter. On Balanced Graphs. Math. Program, 2005, 105(2–3), 233–250.

[Bourjolly et al. 1999] J-M. Bourjolly, G. Laporte, H. Mercure. A combinatorial column generation algorithm for the maximum stable set problem. Oper. Res Lett, 1999, 20 (1), pp. 21–29.

[Bovo et al., 2013] A. Bovo, S. Sanchez, O. Heguy, and Y. Duthen. Clustering Moodle data as a tool for profiling students. In Proc. Second Int. Conf. E-Learning E-Technologies Educ., Sep. 2013, pp. 121-126.

[Brill, 1992] E. Brill. A simple Rule-based Part-of-speech Tagger. Proceedings of the 3rd Conference on Applied Natural Language Processing. March 1992, Trento, Italy, pp. 152-155.

[Buckley et al., 1992] C. Buckley, G. Salton and J. Allan. Automatic Retrieval with Locality Information using Smart. Proceedings of the First Text Retrieval Conference. Gaithersburg, 1992. pp 59-72.

[Budanitsky et Hirst, 2001] A. Budanitsky, G. Hirst. Semantic Distance in WordNet: An experimental Application-oriented Evaluation of Five Measures. In Workshop on WordNet and Other Lexical Ressources, In the North American Chapter of the Association for Computational Linguistics 5NAACL-2001), Pittsburgh, PA.

[Burer et al., 2002] S. Burer, R. D. C. Monteiro, Y. Zhang. Maximum stable set formulations and heuristics based on continuous optimization. Math Program, 2002, 94 (1):137–166.

[Bursher et al., 2014] B. Burscher, D. Odijk, R. Vliegenthart, M. De Rijke & C. H. De Vreese. Teaching the computer to code frames in news: Comparing two supervised machine learning approaches to frame analysis. Communication Methods and Measures, 8 (3), 2014, pp. 190-206. doi:10.1080/19312458.2014.937527.

[Butenko, 2003] S. Butenko. Maximum independent set and related problems, with applications. PhD thesis, University of Florida, 2003.

[Celebi et al., 2013] M. E. Celebi, H. A. Kingravi, and P. A. Vela. A comparative study of efficient initialization methods for the K-Means clustering algorithm.Expert Syst. Appl. 2013, 40 (1), pp. 200-210.

[Ceron et al., 2015] A. Ceron, L. Curini & S. M. Iacus. Using sentiment analysis to monitor electoral campaigns: Method matters-evidence from the United States and Italy. Social Science Computer Review, 2015, 33 (1), pp. 3-20. doi :10.1177/0894439314521983.

[Ceron et al., 2017] A. Ceron, L. Curini & S. M. Iacus. Politics and big data: Nowcasting and forecasting elections with social media.London, UK: Routledge. 2017.

[Cichocki et Unbehauen, 1993] A. Cichocki, R. Unbehauen. Neural Networks for Optimization and Signal Processing. B. G. Teubner Stuttgart, 1993.

[Cogisa et Thierry, 2005] O. Cogisa, E. Thierry. Computing maximum stable sets for distance-hereditary graphs. Discret Optim., 2005, 2(3):185–188.

[Cohen et Hunter, 2008] K. B. Cohen and L. Hunter. Getting started in text mining. PLoS Comput. Biol., vol. 4, ed. 20, 2008.

[Collobert et Weston, 2008] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In International Conference on Machine Learning, ICML, 2008.

[Celebi et al., 2013] M. E. Celebi, H. A. Kingravi, and P. A. Vela. A comparative study of efficient initialization methods for the K-Means clustering algorithm.Expert Syst. Appl. 2013, 40 (1), pp. 200-210.

[Ceron et al., 2015] A. Ceron, L. Curini & S. M. Iacus. Using sentiment analysis to monitor electoral campaigns: Method matters-evidence from the United States and Italy. Social Science Computer Review, 2015, 33 (1), pp. 3-20. doi :10.1177/0894439314521983.

[Ceron et al., 2017] A. Ceron, L. Curini & S. M. Iacus. Politics and big data: Nowcasting and forecasting elections with social media.London, UK: Routledge. 2017.

[Cichocki et Unbehauen, 1993] A. Cichocki, R. Unbehauen. Neural Networks for Optimization and Signal Processing. B. G. Teubner Stuttgart, 1993.

[Cogisa et Thierry, 2005] O. Cogisa, E. Thierry. Computing maximum stable sets for distance-hereditary graphs. Discret Optim., 2005, 2(3):185–188.

[Cohen et Hunter, 2008] K. B. Cohen and L. Hunter. Getting started in text mining. PLoS Comput. Biol., vol. 4, ed. 20, 2008.

[Collobert et Weston, 2008] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitasks learning. In International Conference on Machine Learning, ICML, 2008.

[Cuttung et al., 19992] D. Cutting, D. Karger, J. Pederson, J. Tukey. Scatter/Gatter : A Cluster Approach to Browsing Large Document Collections. ACM SIGIR 92, 1992, pp. 318-329.

[Das et al., 2006] S. Das, A. Abraham, and A. Konar. Spatial information based image segmentation using a modified particle swarm optimization algorithm.Intelligent Systems Design and Applications. Sixth International Conference on IEEE, 2006, vol. 2, pp. 438-444.

[David, 1996] David j. Ketchen; Christopher l. shook (1996). The application of cluster analysis in strategic management research: an analysis and critique. 17(6), 441–458.

[Davies et Bouldin, 2000] D. L. Davies, D. W. Bouldin. A cluster separation measure. IEEE Trans. Pattern Anal. Machine Intell., 2000, 1 (4), pp. 224-22.

[Daver, 2016] Raver. org. deeplearning4j.models. word2vec. https://deeplearning4j.org/api/latest/org/deeplearning4j/models/word2vec/Word2Vec.html.

[De Boom et al., 2016] C. De Boom, S. Van Canneyt, T. Demeester, B. Dhoedt. Representation learning for very short texts using weighted word embedding aggregation. Pattern Recogn Lett. 2016, 80: 150–6. doi: 10.1016/j.patrec.2016.06.012.

[Deerwester et al., 1990] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman. Indexing by Latent Semantic Indexing. Journal of the American Society for Information Science, 1990, 41 (6), pp. 391- 407.

[Desler et Hakimi, 1970] J. F. Desler, et S. L. Hakimi. On finding a maximum internally stable set of a graph. In Proc. of Fourth Annual Princeton Conference on Information Sciences and Systems,1970, vol. 4, pp. 459–462.

[Devlin et al., 2018] J. Devlin, M.W. Chang, K. Lee, K. Toutanova. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805 (2018). Doi: 10.18653/v1/N19-1423

[Devlin et al. 2019] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.2019. arXiv:1810.04805 [cs]. ArXiv: 1810.04805.

[Dreyfus et al., 2004] G. Dreyfus, M. Samuelides, J-M. Martinez, M. B. Gordon, F. Badran, S. Thiria, L. Hérault. Réseaux de neurones Méthodologies et applications. Editeur: Eyrolles, 2004.

[Dreyfus et al., 2008] G. Dreyfus, J. Martinez, M. Samuelides, M. B. Gordon, F. Badran, S. Thiria, L. Hrault. Réseaux de neurones Méthodologie et applications. Eyrolles 2ième édition, 2008.

[Dumont et al., 2018] M. Dumont, P. Reninger, A. Pryet, G. Martelet. Agglomerative hierarchical clustering of airborne electromagnetic data for multi-scale geological studies. Journal of Applied Geophysics, 2018, 157, pp. 1-9.

[El Harti et Boumhidi, 2018] C. El Harti, J. Boumhidi. Fuzzy deep learning based urban traffic incident detection. Cogn. Syst. Res. 2018, 50, pp. 206-213.

[Estivill et Yang, 2000] V. Estivill-Castro, J. Yang. A fast and robust general purpose clustering algorithm. In: Proceeding of 6th Pacific Rim International Conference Artificial Intelligence, Melbourne, Australia, 2000, pp. 208–218.

[Ettaouil et Loqman, 2008] M. Ettaouil, C. Loqman, Constraint Satisfaction Problems Solved by Semidefinite Relaxations. Journal of Wseas Transactions on Computers, 2008, 7 (7), pp. 951-961.

[Ettaouil et al., 2010] M. Ettaouil, C. Loqman et K. Elmoutaouakil. Improved Optimal Competitive Hopfield Network for the Maximum Stable Set Problem. International Journal on Computer Science and Engineering, 2010, 2, (6), pp 2071-2077.

[Ettaouil et al., 2012] M. Ettaouil, C. Loqman, Y. Hami et K. Haddouch. Task Assignment Problem Solved by Continuous Hopfield Networks. International Journal of Computer Science Issues (IJCSI), 2012, Vol. 9, Issue 2, No 1, pp. 206-212.

[Ettaouil et al., 2013] M. Ettaouil, C. Loqman, Y. Hami et K. Haddouch. Maximal Constraint Satisfaction Problems solved by Continuous Hopfield Networks. Wseas Transactions on Computers, 2013, 7 (7), pp. 951-961.

[Evansi et Sulaiman, 1996] D. J. Evansi and M. N. Sulaiman. Solving optimisation problems using neucomp-a neural network compiler. International Journal of Computer Mathematics, 1996, Vol. 62, No. 1, pp. 1-21.

[Fay-Varnier et al., 1991] C. Fay-Varnier, C. Fouqueré, G. Prigent et P. Zweingenbaum. Modules syntaxiques des systèmes d'analyse du français. TSI-Techniques et Science

Informatiques. Editions AFCET-Bordas, 1991, volume 10, N°6, pp. 403-425.

[Fayyad et al., 1996] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. From Data-Mining to Knowledgge Discovery. Chapitre 1, Eds Advances in Knowledge Discovery and Data Mining, AAAI, 1996.

[Feldman et Dagan, 1995] R. Feldman, I. Dagan. Knowledge Discovery in Texts. Proccedings of the First International Conference on Knowledge Discovery, 1995, pp. 112-117.

[Fels, 1994] S. S. Fels. Glove-Talk II: mapping hand gestures to speech using neural networks - An approach to building adaptive interfaces. Thèse de doctorat mention informatique, Université de Toronto, Toronto (Canada), 1994.

[Friden et al., 1989] C. Friden, A. Hertz, D. Werra, D. Stabulus. A technique for finding stable sets in large graphs with tabu search. Computing1989, 42, 1, pp. 35–44.

[Fujisawa et al., 1995] K. Fujisawa, S. Morito, M. Kubo. Experimental Analyses of the Life Span Method for the Maximum Stable Set Problem. The Institute of Statistical Mathematics Cooperative Research Report, 1995, 75, pp. 135–165.

[Fukuyama et Sugeno, 1989] Y. Fukuyama, M. Sugeno. A new method of choosing the number of clusters for the fuzzy c-means method. In: Proc. 5th Fuzzy Syst. Symp., 1989, pp. 247–250 (In Japanese).

[Gabrilovich et Markovitch, 2007] E. Gabrilovich, and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07, pages 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[Giovanelli et al., 2017] C. Giovanelli, X. Liu, S. Sierla, V. Vyatkin, R. Ichise. Towards an aggregator that exploits big data to bid on frequency containment reserve market. In Proceedings of the 43rd Annual Conference of the IEEE Industrial Electronics Society (IECON 2017), Beijing, China, 29 October–1 November 2017; pp. 7514–7519

[Goldberg & Levy, 2014] Y. Goldberg, &O. Levy, (2014). Word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv Preprint arXiv:1402.3722. En ligne : https://arxiv.org/pdf/1402.3722.pdf

[Greene et Cunningham, 2006] D. Greene et P. Cunningham. Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering. Proc. ICML. Homepage of BBCNews database 2006, http://mlg.ucd.ie/datasets/bbc.html.

[Grimmer et Stewart, 2013] J. Grimmer & B. M. Stewart. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. Political Analysis, 2013, 21 (3), pp.267-297. doi:10.1093/pan/mps028.

[Gruber et Rendl, 2003] G. Gruber and F. Rendl. Computational experience with stable set relaxations. SIAM J Opt., 2003, 13, pp. 1014-1028.

[Guha et al., 1998] S. Guha, R. Rastogi, K. Shim. CURE: An efficient clustering algorithm for large databases. In: Proceeding of ACMSIGMOD International Conference Management of Data, 1998, pp.73–84.

[Guha et al., 2000] S. Guha, R. Rastogi, K. Shim. ROCK:a robust clustering algorithm for categorical attributes. Information Systems, 2000, 25 (5), pp. 345–366.

[Guo, 2014] G. Guo. Soft biometrics from face images using support vector machines. In Support Vector Machines Applications; Springer: Berlin/Heidelberg, Germany, 2014; pp. 269-302.

[Habert et al., 1997] B. Habert, A. Nazarenko and A. Salem. Les linguistiques de corpus. Armand Colin/Masson (edts.), 1997, 240p. ISBN: 2200017758.

[Ham et Kostanic, 2001] F. M. Ham and I. Kostanic. Principles of Neurocomputing for Science and Engineering. Mc. Graw-Hill, New York. 2001.

[Han et al., 2011] J. Han, M. Kamber, and J. Pei. Major Tasks in Data Preprocessing. In Data Mining Concepts and Techniques, 3 ed: Morgan Kaufmann, 2011.

[Han et al., 2012] J. Han, J. Pei et M. Kamber. Data Mining Concepts and Techniques (3rd ed), USA: Morgan Kauffman Publishers, 2012.

[Hàstad, 1999] J. Hàstad. Clique is hard to approximate within $n^{1-\varepsilon}$. Acta Math., 1999, 182, pp. 105-142.

[Hastie et al., 2001] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning: Data mining, inference and prediction. (2d ed.) Springer-Verlag, 2001.

[Hopfield, 1982] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences of the United States of America, 1982, Vol. 79, pp. 2554-2558.

[Hopfield, 1984] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-states neurons. Proceedings of the National academy of sciences of the USA, 1984, 81, pp. 3088-3092.

[Hopfield et Tank, 1985] J. J. Hopfield and D.W. Tank. Neural computation of decisions in optimization problems. Biological Cybernetics, 1985, Vol. 52, pp. 1-25.

[Hopfield, 1987] J. J. Hopfield. Learning algorithms and probability distributions in feedforward and feedback neural networks. Proceedings of the National Academy of Sciences, 1987, 84, pp. 429-433.

[Hopkins & King, 2010] D. J. Hopkins & G. King. A method of automated non parametriccontent analysis for social science. American Journal of Political Science, 2010, 54 (1), pp. 229-247. doi: 10.1111/ajps.2010.54.issue-1

[Hoppner et al., 1999] F. Hoppner, F. Klawonn, R. Kruse. Fuzzy Cluster Analysis: Methods for Classification. Data Analysis, and Image Recognition, Wiley, NewYork, 1999.

[Hsu et al., 2007] C. C. Hsu, C. L. Chen, Y. W. Su. Hierarchical clustering of mixed data based on distance hierarchy. Information Sciences, 2007, 177 (20), pp. 4474–4492.

[Huang, 1997] Z. X. Huang. A fast-clustering algorithm to cluster very large categorical data sets in data mining. In: Proceeding of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997, pp.1–8.

[Huang, 1998] Z. X. Huang. Extensions to the K-Means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery, 1998, 2 (3), pp. 283–304.

[Huang et al., 2011] M. Huang, Z. S. He, X. L. Xing, Y. Chen. New K-Means clustering center select algorithm. Computer Engineering and Applications, 2011, 47, (35), pp. 132-134.

[Idrissi et al., 2015] A. Idrissi, H. Rehioui, A. Laghrissi and S. Retal. An improvement of DENCLUE algorithm for the data clustering. 5th Internationl Conference on Information & Communication Technology and Accessibility, 2015, pp. 1- 6.

[Jacquemin et Zweigenbaum, 2000] C. Jacquemin, P. Zweigenbaum. Traitement automatique des langues pour l'accès au contenu des documents. In Jacques Le Maître, J. Charlet et C. Garbay, éditeurs, le document en sciences du traitement de l'information, chapitre 4, pages 71-109. Cepadues, Toulouse, 2000.

[Jain et al., 1999] A. K. Jain, M. N. Murty and P. J. Flynn. Data clustering: A review. ACM Comput. Surv. Sept. 1999, 31, 3, pp. 264–323.

[Jiang et Conrath, 1997] J. J. Jiang et D. W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In International Conference Research on Computational Linguistics (ROCLING X) 1997.

[Jiang et al., 2018] M. Jiang, Y. Liang, X. Feng, X. Fan, Z. Pei, Y. Xue, R. Guan. Text classification based on deep belief network and softmax regression. Neural Comput. Appl. 2018, 29, pp. 61-70.

[Jin et Mobasher, 2003] X. Jin and B. Mobasher. Using Semantic Similarity to Enhance Item-Based Collaborative Filtering. In Proceedings of the 2nd IASTED International Conference Information and Knowledge Sharing, Scottsdale, Arizona, November 2003.

[Jin et Hao, 2015] Y. Jin, et J.-K. Hao. General swap-based multiple neighborhood tabu search for the maximum independent set problem. Engineering Applications of Artificial Intelligence, 2015, 37, pp. 20–33.

[Jixue, 2009] D. Jixue. Data mining for time series based on wave cluster. International Forum on Information Technology and Applications, 2009, pp. 697- 699.

[Jose et Gomez, 2016] A. Jose-Garcia and W. Gomez-Flores. Automatic clustering using nature-inspired metaheuristics: A survey. Applied Soft Computing, 2016, vol. 41, pp. 192-213.

[Karim et al., 2018] Karim, A., Loqman, C. and Boumhidi J. (2018). Determining the Number of Clusters using Neural Network and Max Stable Set Problem. The 1st. Int. Conf. On Intelligent Computing in Data Sciences. Procedia Computer Science, Meknes, Morocco, 127, 16-25.

[Joulin et al., 2016] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. 2016, http://arxiv.org/abs/1612.03651.

[Joya et al., 2002] G. Joya, M. Atencia, and F. Sandoval. Hopfield neural networks for optimization: study of the different dynamics.Neuro-computing, 2002, 43, (1), pp. 219-237.

[Karamizadeh et al., 2014] S. Karamizadeh, S. M. Abdullah, M. Halimi, J. Shayan, M. Javad Rajabi. Advantage and drawback of support vector machine functionality. In Proceedings of the 2014 International Conference on Computer, Communications, and Control Technology (I4CT), Langkawi, Malaysia, 2–4 September 2014; pp. 63–65.

[Karypis et al., 1999] G. Karypis, E. Han, V. Kumar. Chameleon: hierarchical clustering using dynamic modeling.IEEE Computer, 1999, 32 (8), pp. 68–75.

[Kaufman et Rousseeuw, 1990] L. Kaufman, P. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley,1990.

[Kim, 2014] Y. Kim. Convolutional neural networks for sentence classification. arXiv preprint arXiv:14085882. (2014). doi: 10.3115/v1/D14-1181.

[Kim et al., 2008] Y.-M. Kim, J.-F. Pessiot, M.-R. Amini, P. Gallinari. Apprentissage d'un espace de concepts de mots pour une nouvelle représentation des données textuelles. Laboratoire d'Informatique de Paris 6104, Proceedings of the 5th Conférence en Recherche d'Information et Applications, 12, 2008.

[Kohonen, 1995] T. Kohonen. Self-Organizing Maps. Berlin/Heidelberg, Germany: Springer, vol. 30, 1995.

[Kohonen, 2000] T. Kohonen, S. Kaski, K. Lagus, J. Salojr , J. Honkela, V. Paatero, A. Saarela. Self-organization of a massive document collection. IEEE transaction on neural networks, 11, No. 3, 2000.

[Kothari et Pitts, 1999] R. Kothari, D. Pitts. On finding the number of clusters. Pattern Recognition Letters, 1999, 20 (4), pp. 405–416.

[Kovesi et al., 2001] B. Kövesi, J.-M. Boucher, and S. Saoudi. Stochastic K-Means algorithm for vector quantization. Pattern Recognit. Lett., May-2001, vol. 22, no. 6, pp. 603–610.

[Kowsari et al., 2017] K. Kowsari, D.E. Brown, M. Heidarysafa, K. Jafari Meimandi, M.S. Gerber, L. E. Barnes. HDLTex: Hierarchical Deep Learning for Text Classification. Machine Learning and Applications (ICMLA). In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017.

[Kowsari et al., 2018] K. Kowsari, M. Heidarysafa, D. E. Brown, K. Jafari Meimandi, L. E.Barnes. RMDL: Random Multimodel Deep Learning for Classification. In Proceedings of the 2018 International Conference on Information System and Data Mining, Lakeland, FL, USA, 9–11 April 2018; doi:10.1145/3206098.3206111.

[Kowsari et al., 2019] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mondue, L.Barnes, D. Brown. Text Classification Algorithms: A Survey. Information-an International Interdisciplinary Journal, vol. 10, no. 4, 2019, p. 150.

[Kummamuru et al., 2004] K. Kummamuru, R. Lotlikar, A. Roy, K. Signal, R. Krishnapuram. Monothetic Docu- ment Clustering Algorithm for Summarization and Browsing Search Results. In ACM WWW's04, 2004.

[Kuncheva et Bezdek, 1997] L. I. Kuncheva and J. C. Bezdek. Selection of cluster prototypes from data by a genetic algorithm. EUFIT, September-1997, pp.1683-1688, Aachen, Germany.

[Kuo et al., 2012] R. Kuo, Y. Syu, Z.-Y. Chen, and F.-C. Tien. Integration of particle swarm optimization and genetic algorithm for dynamic clustering.Information Sciences, 2012, vol. 195, pp. 124-140.

[Lai et al., 2015] S. Lai, L. Xu, K. Liu, J. Zhao. Recurrent Convolutional Neural Networks for Text Classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015 ; Volume 333, pp. 2267–2273.

[Landauer et al.,1998] T. K. Landauer, P. W. Foltz and D. Laham. Discourse processes. 1998, 25 (2-3), pp. 259-284.

[Lang] K. Lang. www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html

[Larose, 2014] D. Larose. Discovering knowledge in data: an introduction to data mining. John Wiley & Sons, Inc., Hoboken, New Jersey, 2014.

[Latrache et al., 2014] A. Latrache, E. Nfaoui, J. Boumhidi.A Mobile Agent based Approach for Automating “Discover-Compose” Process of Semantic Web Services. J. Comput. Sci. 2014, 10 (9), pp. 1628-1641.

[Lee, 1995] J. H. Lee. Combining Multiple Evidence from Different Properties of Weighting Schemes. Proceedings of the 18th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR 1995). Washington, USA, 1995. PP 180-188.

[Lefèvre, 2002] S. Lefèvre. Une nouvelle approche pour la classification non supervisée en segmentation d’image. Thèse de doctorat en électronique à l’université de Strasbourg, 2002.

[Lehmann et al., 2006] K. A. Lehmann, M. Kaufmann, S. Steigele, K. Nieselt. On the maximal cliques in c-max-tolerance graphs and their application in clustering molecular sequences. Algorithm Molecular Biol, 2006, 1:9, pp. 1–17.

[Levy et al., 2015] O. Levy, Y. Goldberg, &I. Dagan.Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics, 2015, 3, pp. 211-225. En ligne: http://bit.ly/2Y06VfQ

[Lewis et Croft, 1990] D. D. Lewis and W. B. Croft. Term clustering of syntactic phrases. University of Massachusetts, Colins Technique Report, 1990, pp. 90-71.

[Li et Biswas, 2002] C. Li, G. Biswas. Unsupervised learning with mixed numeric and nominal data. IEEE Transaction son Knowledge and Data Engineering, 2002, 14 (4), pp. 673–690.

[Lodhi et al., 2002] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, C. Watkins. Text classification using string kernels. J. Mach. Learn. Res. 2002, 2, pp. 419-444.

[Lovász, 1979] L. Lovász. On the Shannon capacity of a graph. IEEE Transactions on Information Theory, 1979, 25 (1), pp. 1–7.

[Lowe et Benoit, 2013] W. Lowe & K. Benoit. Validating estimates of latent traits from textual data using human judgment as a benchmark. Political Analysis, 2013, 21 (3), pp. 298-313. doi:10.1093/pan/mpt002.

[Mac-Queen, 1967] J. B. Mac Queen. Some methods for classification and analysis of multivariate observations. In: Proceeding of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[Mannino et Sassano, 2005] C. Mannino, A. Sassano. An exact algorithm for the maximum stable set problem. Comput. Optim. Appl., 2005, 3 (3), pp. 243–258.

[Mikolov et al., 2013a] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013.

[Mikolov et al., 2013b] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean.

[Miller, 1995] G. A. Miller. WordNet: a lexical database for English. Communications of the ACM, 1995, 38 (11), pp. 39-41.

[Mnih et Hinton, 2007] A. Mnih, and G. Hinton. Three new graphical models for statistical language modelling. In ICML ’07: Proceedings of the 24th international conference on Machine learning. ACM. 2007.

[Morin, 1999] E. Morin. Extraction de liens sémantiques entre termes à partir de corpus de textes techniques. Thèse de doctorat, Université de Nantes, Faculté des sciences et des techniques. 8 décembre 1999.

[Lang, 1995] K. Lang. Newsweeder: Learning to filter netnews. In Proceedings of the Twelfth International Conference on Machine Learning, 1995, pp. 331-339. http://qwone.com/~jason/20Newsgroups/

[Ng, 2012] A. Ng. Clustering with the K-Means Algorithm, Machine Learning, 2012.

[Ng et Han, 2002] R. T. Ng, and J. Han. Clarans: a method for clustering objects for spatial data mining. IEEE Transactions on Knowledge and Data Engineering, 2002, 14, pp. 1003–1016.

[Pacual et al., 2010] D. Pacual, F. Pla, J. S. Sánchez. Cluster validation using information stability measures. Pattern Recognition Letters, 2010, 31, pp. 454-461.

[Palangi et al., 2016] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R.

[Patwary et al., 2013] Md. M. A. Patwary, D. Palsetia, A. Agrwal, W. K. Liao, F. Manne and A. Choudhary. Scalable parrallel OPTICS data clustering using graph algorithmic techniques. International Conference on High Performance Computing Networking Storage and Analysis, 2013, pp. 1- 12.

[Pennington et al., 2014] J. Pennington, R. Socher and C. Manning. GloVe: global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, Doha, Qatar, 2014.

[Porter, 1980] M. F. Porter. An algorithm for suffix stripping. Program: electronic library and information systems, 1980, Vol. 14 No. 3, pp. 130-137.

[Pouliquen, 2002] B. Pouliquen. Indexation de textes médicaux par extraction de concepts, et ses utilisations. Thèse, Université Rennes I, 7 juin 2002.

[Quinlan, 1987] J. R. Quinlan. Simplifying decision trees.Int. J. Man-Mach. Stud. 1987, 27, pp. 221–234.

[Rajman et Lebart, 1998] M. Rajman, L. Lebart. Similarités pour données textuelles. Computer Science, 1998.

[Ramadan et Tairi, 2015] H. Ramadan and H. Tairi. Collaborative Xmeans-EM clustering for automatic detection and segmentation of moving objects in video.

IEEE/ACS 12th International Conference of Computer Systems and Applications,2015, pp. 1-2.

[Rebennack, 2008] S. Rebennack. Stable Set Problem: Branch & Cut Algorithms. In Encyclopedia of Optimization (eds). Springer, Boston, MA, 2008. https://doi.org/10.1007/978-0-387-74759-0_634

[Régin, 2003] J-C. Régin. Using constraint Programming to Solve the Maximum Clique Problem.Lecture Notes in Computer Science. Springer, Berlin, 2003, pp. 634–648.

[Réhel, 2005] S. Réhel. Catégorisation automatique de textes et cooccurrence de mots provenant de documents non étiquetés. Thèses et mémoires, Université Laval, 2005.

[Rencher, 2003] A. C. Rencher. Methods of Multivariate Analysis. Wiley, 2003.

[Rezaeinia et al., 2017] S. M. Rezaeinia, A. Ghodsi and R. Rahmani. Improving the accuracy of pre-trained word embeddings for sentiment analysis, 2017, http://arxiv.org/abs/1711.08609v1.

[Rodriguez & Egenhofer, 2003] M.A. Rodriguez & M. J. Egenhofer. Determining Semantic Similarity Among Entity Classes from Different Ontologies. IEEE Tra. on Know. & Data Engi., 2003, vol. 15, n° 2, pp. 442-56.

[Rousseeuw, 1987] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 1987, 20, pp. 53-65.

[Ruch et al., 2003] P. Ruch, C. Chichester, G. Cohen, G. Coray, F. Ehrler, H. Ghorbel, H. Muller, V. Pallotta. Report on TREC 2003 Experiment: Genomic Track. Conférence TREC 2003, Gaitherburg, Maryland, November 18-21, 2003.

[Salton et McGill, 1986] G. Salton, and M. J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA, 1986.

[Salton et Buckley, 1998] G. Salton and C. Buckley. Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management, 1998, (5), pp. 513-523.

[Sarkar et al., 1997] M. Sarkar, B. Yegnanarayana, and D. Khemanial. A clustering algorithm using an evolutionary-based approach. Pattern Recognition Letters, 1997, Vol. 18, Iss. 10, pp. 975-986.

[Satyanarayana et Acquaviva, 2014] A. Satyanarayana and V. Acquaviva. Enhanced cobweb clustering for identifying analog galaxies in astrophysics. IEEE 27th Canadian Conference on Electrical and Computer Engineering, 2014, pp. 1-4.

[Shen-Yi et al., 2018] Q. Shen-Yi, L. Hui-hui and L. Dai-yi. Research and Application of Improved K-Means Algorithm in Text Clustering.DEStech Transactions on Computer Science and Engineering (pcmm), 2018. doi:10.12783/dtcse/pcmm2018/23653.

[Sherkat et Velcin, 2018] E. Sherkat, J. E.E. Velcin Milios. Fast and Simple Deterministic Seeding of K-Means for Text Document Clustering. 9th Conference and Labs of the Evaluation Forum (CLEF). Proceedings, 2018, pp. 76-88.

[Serraji et al., 2016] M. Serraji, D. O. El Amine J. Boumhidi. Multi swarm optimization based adaptive fuzzy multi agent system for microgrid multi-objective energy management. Int. J. Knowl. Based Intell. Eng. Syst. 2016, 20 (4), pp. 229-243.

[Shih et al., 2004] H. S. Shih, U-P. Wen, S. Lee, K-M. Lan et H-C. hsiao. A neural network approach to multi-objective and multilevel programming problems. Computers and Mathematics with Applications, 2004, Vol. 48, pp. 95-108.

[Siddique et al., 2018] Md. Abu. Siddique, R. Arif, M. M. R. Khan, and Z. Ashrafi. Implementation of Fuzzy C-Means and Possibilistic C-Means Clustering Algorithms. Cluster Tendency Analysis and Cluster Validation, 2018. 10.20944/preprints 201811.0581.v1.

[Singhal, 1997] A. Singhal. Term Weighting Revisited. PhD thesis, Department of Computer Science, Cornell University, 1997.

[Smith, 1999] K. A. Smith. Neural Networks for Combinatorial Optimization: A Review of More Than a Decade of Research, INFORMS Journal on Computing, 1999, Vol. 11, pp.15-34.

[Song et al., 2015] J. Song, X. Li and Y. Liu. An Optimized K-Means Algorithm for Selecting Initial Clustering Centers. International Journal of Security and Its Applications, 2015, 9, (10), pp. 177-186.

[Srivastava et al., 2005] T. Srivastava, P. Desikan, and V. Kumar. Web Mining – Concepts, Applications and Research Directions. In Foundations and Advances in Data Mining, T. Y. L. Wesley Chu, Ed., ed: Springer Berlin Heidelberg New York, 2005.

[Subbalakshmi et al., 2015] C. Subbalakshmi, G. Krishna, K. Rao, P. Rao. A method to find optimum number of clusters based on fuzzy silhouette on dynamic data set. Procedia Computer Science, 2015, 46, pp. 346-353.

[Talavàn et Yànez, 2002] P. M. Talavàn, J. Yànez. Parameter setting of the Hopfield network applied to TSP. Neural Networks, 2002, 15, pp. 363-373.

[Talavàn et Yànez, 2005] P. M. Talavàn, J. Yànez. A continuous Hpfield network equilibrium points algorithm. computers and operations research, 2005, 3 (2), pp. 2179-2196.

[Talavàn et Yànez, 2006] P. M. Talavàn, J. Yànez. The generalized quadratic knapsack problem. A neuronal network approaches. Neural Networks, 2006, 19, pp. 416-428.

[Tatsumi, 2002] K. Tatsumi, Y. Yagi, T. Tanino. Improved projection Hopfield network for the quadratic assignment problem. SICE 2002. proceedings of the 41 st SICE annual conference, 2002, 4, pp. 2295-2300.

[Theodoridis et Koutroubas, 1999] S. Theodoridis, K. Koutroubas. Pattern Recognition. Academic Press, USA, 1999.

[Thiongane et al., 2005] B. Thiongane, A. Najih and G. Plateau. An Adapted Step Size Algorithm for a 0-1 Bi-knapsack Lagrangean Dual. Annals of Operations Research, 2005, 139 (1), pp. 353-373.

[Tong et al., 2011] X. Tong, F. Meng, and Z. Wang. Optimization to K-Means initial cluster centers.Computer Engineering and Design, 2011, 32, (8), pp. 2721-2723.

[Tuinstra, 2016] T. R. Tuinstra. Range and velocity disambiguation in medium PRF radar with the DBSCAN clustering algorithm. IEEE National Aerospace and Electronics Conference and Ohio Innovation Summit, 2016, pp. 396- 400.

[Van Atteveldt et al., 2008] W. Van Atteveldt, J. Kleinnijenhuis & N. Ruigrok. Parsing, semantic networks, and political authority using syntactic analysis to extract semantic relations from dutch newspaper articles. Political Analysis, 2008, 16 (4), pp. 428-446. doi:10.1093/pan/mpn006.

[Vapnik, 2013] Vladimir Vapnik. The nature of statistical learning theory. Springer Science & Business Media, 2013.

[Vaswani et al., 2017] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. Attention Is All You Need. arXiv:1706.03762 [cs.CL]. 2017.

[Venkatkumar et Shardaben, 2016] I. A. Venkatkumar and S. J. K. Shardaben. Comparative study of data mining clustering algorithms. International Conference on Data Science and Engineering, 2016, pp. 1-7.

[Verweij et Aardal, 1999] B. Verweij, K. Aardal. An Optimization Algorithm for Maximum Independent Set with Applications. In Map Labelling, Lecture Notes in Computer Science, 1999, vol. 1643, pp 426–437.

[Wang et al., 2017] B. Wang, J. Xu, J. Li, C. Hu, J. S. Pan. Scene text recognition algorithm based on faster RCNN. In Proceedings of the 2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS), Harbin, China, 3–5 June 2017; pp. 1–4.

[Wena et al., 2009] Ue-P. Wena, Kuen-M. Lan, Hsu-S. Shih. A review of Hopfield neural networks for solving mathematical programming problems, European Journal of Operational Research, 2009, Vol. 198, pp. 675-687.

[Witten et Frank, 2005] I. H. Witten et E. Frank. Data Mining: Practical Machine Learning Tools and Techniques. Second Edition, Morgan Kaufmann Series in Data Management Systems, 2005.

[Wolf et al., 2020] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac,T. Rault, R. Louf, M. Funtowicz, and J. Brew. HuggingFace's Transformers: State-of-the-art Natural Language Processing. 2020. arXiv:1910.03771 [cs]. ArXiv: 1910.03771.

[Wu, 2012] J. Wu. Advances in K-Means Clustering: A Data Mining Thinking. Springer Theses, Springer Berlin Heidelberg, 2012, ISBN 978-3-642-29807-3.

[Xie et Beni, 1991] X. L. Xie, G. Beni. A validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell., 1991, 13, 8, pp. 841–847.

[Xiong et al., 2011] T. K. Xiong, S. R. Wang, A. Mayers, E. Monga. DHCC: Divisive hierarchical clustering of categorical data. Data Mining and Knowledge Discovery, 2011. doi:10. 1007/s10618-011-0221-2.

[Xu et Croft, 1999] J. Xu, W. Croft. Cluster-based Language Models for Distributed Retrieval. ACM SIGIR 99, 1999, pp. 254-261.

[Yang et al., 2016] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, E. H. Hovy. Hierarchical Attention Networks for Document Classification. In Proceedings of the HLT-NAACL, San Diego, CA, USA, 12–17 June 2016, pp. 1480-1489.

[Yang et al., 2019] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, QV. Le. QV. Xlnet: generalized auto regressive pretraining for language understanding. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R, editors. Advances in Neural Information Processing Systems 32. Curran Associates, Inc. 2019. pp. 5754-5764.

[Yin et al., 2012] K. Ying, M. Chang, A. F. Chiarella, and J.-S. Heh. Clustering students based on their annotations of a digital text. In Proc. 2012 IEEE Fourth Int. Conf. Technol. Educ., Jul. 2012, pp. 20-25.

[Zhang et al., 1996] T. Zhang, R. Ramakrishnan, M. Livny. BIRCH: an efficient data clustering method for very large data bases. In: Proceeding of ACMSIGMOD International Conference Management of Data, 1996, pp.103-114.

[Zhang et Byron, 2015] Y. Zhang, C. W. Byron. A sensitivity analysis of (and practitioners'

[Zhang et al., 2015] X. Zhang, J. Zhao, Y. Le. Cun. Character-level convolutional networks for text classification. Adv. Neural Inf. Process. Syst. 2015, 28, pp. 649-657.

[Zweigenbaum, 1997] P. Zweigenbaum. Construction d'une représentation sémantique en graphes conceptuels – TALN 1997.