MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ MOHAMED KHIDHER - BISKRA
FACULTÉ DES SCIENCES EXACTES, DES SCIENCES DE LA NATURE ET DE LA VIE
DÉPARTEMENT D'INFORMATIQUE

T H È S E

pour obtenir le diplôme de

Docteur en Sciences

SPÉCIALITÉ : **INFORMATIQUE**

Présentée par

MEADI MOHAMED NADJIB

Technique basée HITS/SVM pour la réduction et la pondération des caractéristiques des pages Web

Devant le jury :

Président :	Pr. BACHIR Abdelmalik	Université de Biskra
Rapporteur :	Pr. BABAHENINI Mohamed Chaouki	Université de Biskra
Co-Rapporteur :	Pr. TALEB AHMED Abdelmalik	Université de Valenciennes
Examinateur :	Pr. CHIKHI Salim	Université de Constantine 2
Examinateur :	Pr. MOUSSAOUI Abdelouahab	Université de Setif 1
Examinateur :	Dr. BITAM Salim	Université de Biskra

Dédicaces

À,

mes parents,

ma femme et mes filles,

Mon frère et mes sœurs,

Toute la famille,

Mes amis.

Remerciements

Je tiens premièrement à prosterner remerciant Allah le tout puissant de m'avoir donné le courage et la patience pour terminer ce travail.

Je remercie ensuite mon cher encadreur Dr. Babahenini Mohamed Chaouki pour m'avoir honoré par son encadrement, ses conseils précieux, sa patience et ses nobles valeurs humaines.

Je remercie également mon co-encadreur Pr. Taleb-Ahmed Abdelmalik, professeur à l'université de valenciennes, pour m'avoir accueilli dans son laboratoire LAMIH, pour ses conseils et son suivi continu.

Mes remerciements vont également aux membres de jury pour m'avoir honoré par leur évaluation de ce travail.

Je témoigne toute ma reconnaissance à mon ami Dr. TIBERMACINE Okba pour ses aides et soutiens dans la rédaction et la lecture mon article.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction générale

Sommaire

1.1 Motivations

Actuellement, le World Wide Web est classé la première source d'informations dans le monde. Le problème est que le Web est un corpus hypertexte d'une taille énorme et qui ne cesse de croître de façon exponentielle sans aucun contrôle éditorial. Dans l'autre côté, la capacité humaine à trouver, lire et comprendre le contenu reste constante, alors il peut être extrêmement difficile pour les utilisateurs de localiser les ressources qui sont de bonne qualité et adaptées à leurs besoins d'information.

La plupart des données Web, sont des pages Web qui n'ont presque pas une structure unificatrice, et possédant une variabilité dans le style de création et le contenu qui est beaucoup plus grande que dans les collections de documents textes traditionnels. A cause de ces contraintes, il est impossible d'appliquer des techniques de gestion de base de données et la recherche d'informations classiques dans l'extraction des informations et des connaissances contenues dans les documents Web.

Pour remédier à ce problème le web mining est apparu comme une solution d'analyse des pages web, dont le but est d'utiliser les méthodes et les techniques du data mining dans l'extraction des connaissances à partir des pages Web avec des techniques qui respectent la nature non structurée des pages Web, concernant surtout les phases de prétraitement et l'extraction des caractéristiques.

Le Web mining vise à découvrir des connaissances à partir des hyperliens Web, le contenu des pages Web, et les journaux d'utilisation (Web logs). Sur la base des types de données Web qui ont été utilisées dans le processus de l'exploration, le Web mining peut être catégorisé en trois axes : Exploration de la structure du Web, l'exploration du contenu du Web et l'exploration de l'usage du Web.

Avant le démarrage de n'importe quel processus d'analyse des données Web, il faut d'abord résoudre le problème de la taille volumineuse de données d'entrées. Ce processus doit traiter des millions de pages Web, des dizaines de milliers de caractéristiques et des centaines ou des milliers de catégories. Par conséquent, des mécanismes de réduction de dimensions efficaces sont extrêmement nécessaire.

Une méthode de réduction de dimension est souvent définie comme un processus de prétraitement de données qui permet d'éliminer les informations redondantes et inutiles. Les méthodes de réduction de dimension sont généralement classées en deux catégories :

— La sélection de caractéristiques qui regroupe les algorithmes permettant de sélectionner un sous-ensemble de caractéristiques parmi un ensemble de départ, en utilisant divers critères et différentes méthodes.
— L'extraction de caractéristiques qui permet de créer de nouveaux ensembles de caractéristiques, en utilisant une combinaison des caractéristiques de l'espace de départ.

Dans cette thèse, nous nous sommes intéressés à la sélection des caractéristiques qui est une technique permettant de choisir les caractéristiques, les variables ou les mesures les plus intéressantes, pertinentes et adaptées à un système de résolution d'un problème particulier.

Par rapport à l'extraction des caractéristiques, la sélection de caractéristiques présente plusieurs avantages liés à la réduction de la taille de données. D'une part, cette réduction rend la gestion des données beaucoup plus facile et d'autre part, elle aide à mieux comprendre les résultats fournis par un système basé sur ces caractéristiques. Par exemple, pour un problème de classification, ce processus de sélection ne réduit pas seulement le temps d'apprentissage mais il aide aussi à mieux comprendre les résultats fournis par le classificateur et à améliorer parfois la précision de la classification, en favorisant les caractéristiques les moins bruitées par exemple.

La littérature compte plusieurs techniques de sélection de caractéristiques qui entre dans une des trois approches de sélection des caractéristiques à savoir ; filtrages, enveloppes et intégrées. Dans cette thèse, nous allons proposer une approche de sélection de caractéristiques de type filtrage afin de l'utiliser dans la réduction de caractéristiques d'un classificateur des pages Web.

1.2 Contributions

Dans le cas de recherche sur le Web et l'analyse de liens, les deux algorithmes de recherche de lien hypertexte les plus influents sont : PageRank et HITS. Ces deux algorithmes sont originalement inspirés du domaine analyse des réseaux sociaux.

PageRank et HITS exploitent les hyperliens du Web pour classer les pages Web en fonction de leurs niveaux d'autorité.

L'utilisation de l'algorithme HITS dans cette thèse est motivée par sa capacité à classer des pages Web en fonction du thème de la requête qui peut fournir pages autorités et hubs plus pertinentes. En outre, par rapport à PageRank, HITS est un algorithme général pour le calcul des autorités et des hubs afin de classer les pages Web récupérées. De plus, les résultats obtenus par HITS peuvent également être combiné avec des classements basés sur la recherche d'informations classique. Par conséquent, le classement par HITS peut améliorer la sélection des meilleures caractéristiques et des pages Web.

Depuis son apparition, l'algorithme HITS a connu de nombreuses améliorations. Par exemple, Borodin et al. dans [Borodin 2005] ont présenté un cadre théorique basé sur l'algorithme HITS pour l'étude de classement des pages Web, en utilisant quelques algorithmes d'analyse des liens.

L'algorithme HITS a été utilisé dans plusieurs domaines en dehors de sans contexte originale. Les auteurs de [Xu 2009] ont proposé un algorithme de crédit scoring fondé sur le classement avec l'analyse des liens et les machine à vecteurs de support, leur algorithme détermine automatiquement si une banque devrait fournir un prêt à un demandeur ou non. De plus, Deguchet a utilisé cet algorithme pour enquêter sur les pôles économiques et les autorités du réseau commercial mondial entre les années 1992-2012 [Deguchi 2014].

Dans cette thèse, nous proposons une nouvelle approche de construction d'un classificateur des pages Web basée sur les SVMs, pour lequel nous proposons l'utilisation de l'algorithme HITS, pour :

1. Réduire la taille du vecteur de caractéristiques de l'ensemble d'apprentissage.
2. Pondérer les caractéristiques contenues dans ses pages Web, en utilisant le vecteur autorité.

1.3 Organisation de la thèse

Notre thèse est composée de deux parties, une partie théorique et une partie pratique, chacune est composée de deux chapitres. Dans le premier chapitre, nous allons étudier le domaine de l'exploration du Web et ses différents axes à savoir ; l'analyse de contenu du Web, l'analyse de structure du Web et l'analyse d'usage du Web. Dans le même chapitre, nous allons présenter trois domaines très intéressants dans la fouille du web, qui sont la recherche d'information dans le Web, la classification des pages Web et enfin l'analyse des liens.

Le deuxième chapitre de la partie théorique est consacré à l'aspect de réduc-

tion de caractéristiques pour lequel, nous présentons les deux grandes approches qui compose ce domaine. Ces deux approches sont l'extraction des caractéristiques et la sélections des caractéristiques. Etant donné que nous sommes intéressées par la sélection des caractéristiques, nous allons étudier en détail ce domaine en présentant ses différentes types qui sont ; sélection par filtrage, enveloppe et intégrée. Nous allons aussi exposer quelques méthodes de sélection ds caractéristiques, extraites de la littérature et qui ont des bases statistiques. Ce chapitre se termine par une présentation des quelques travaux concernant la sélection des caractéristiques contenues les pages Web.

Dans le troisième chapitre, nous allons exposer notre principale contribution qui consiste en une approche de création d'un classificateur des pages Web qui utilise un algorithme d'analyse de lien HITS pour la sélection et la pondération des caractéristiques. Nous présenterons aussi les Machines à vecteurs de support (SVM) qui ont démontré leur efficacité dans la classification des données de grande échelle et que nous avons choisi pour l'apprentissage de notre classificateur. On exposera enfin, l'algorithme HITS et deux autres variants qui sont HubAvg et AT(K).

Le quatrième chapitre, est consacré à la validation de notre approche. En premier lieu, nous allons faire une description détaillée des ensembles des documents utilisées dans les différentes expérimentations, pour lesquelles nous allons préciser le nombre des pages Web (positives ou négatives) utilisées pour la phase d'apprentissage et la phase de tests et le nombre des caractéristiques dans chaque classe. Dans ce chapitre, nous allons exposer aussi le matériel utilisé et les outils de développement et de validation comme par exemple le langage de programmation, l'environnement de développement et les bibliothèques ...etc, ensuite, nous allons exposer et discuter les résultats obtenus de différentes expérimentations que nous avons effectué. Ce chapitre se termine par une étude concernant la complexité et le temps écoulé par les étapes proposées.

Notre thèse sera clôturée par une conclusion générale qui présente une synthèse des contributions apportées ainsi que les pistes définissant des perspectives possibles pour nos futurs travaux.

Chapitre 2

Exploration du Web

Sommaire

2.1 Introduction

Le Web est une grande source de différents types de données, qui contient une grande quantité de connaissances invisibles, qui peuvent être découvertes en utilisant les paradigmes d'extraction de données et les algorithmes d'apprentissage.

L'exploration du Web (Web mining) est l'application des techniques du fouille des données structurées (Data Mining) sur les données contenue dans le Web. Tous ces types de techniques sont basées sur des approches intelligentes de calcul, ou ce que l'on appelle l'intelligence informatique, qui sont largement utilisés dans la recherche de base de données, data mining, l'apprentissage, et la recherche de l'information et ...etc. Selon le type des données analysées l'exploration du Web peut être classée en trois catégories : Exploration du contenu du Web, exploration de la structure du Web et Exploration de l'usage du Web.

Le reste de ce chapitre est organisé comme suit ; La section suivante sera consacrée à l'exploration des données d'une manière générale. Après la présentation des spécificités du Web, nous allons étudier l'exploration du Web et ses différents axes, dans la section 5. Ensuite, nous allons présenter le domaine de la recherche d'information dans le Web, puis nous allons entamer la classification des pages Web et enfin l'analyse des liens.

2.2 Exploration des données

L'Exploration de données (en anglais : data mining) est défini par l'étude de la collecte, le nettoyage, le traitement, l'analyse, et l'obtention des connaissances utiles à partir de données [Aggarwal 2015]. Le "data mining" est un terme générique large qui est utilisé pour décrire les différents aspects du traitement des données. Le but principal de l'exploration de données est la découverte des connaissances cachées ou invisibles, dans ces données, sous forme de modèles [Xu 2010].

Beaucoup d'autres termes ont un sens similaire que l'exploration de données, tels que l'Extraction de Connaissances à partir de Données ou extraction de connaissances (ECD) (en anglais : KDD : Knowledge Data Discovery), l'analyse des données/modèle, l'archéologie de données, la fouille de données,...etc [Han 2011]. Alternativement, d'autres auteurs comme dans [Cios 2007] considèrent l'exploration de données comme le cœur du processus d'extraction de connaissances (ECD).

L'extraction des connaissances est un processus itératif comprenant une liste de six processus successifs (Figure 2.1) [Fayyad 1996] :

1. **Nettoyage des données** : Cette première étape traite les données manquantes et redondantes dans le fichier source. Les données du monde réel peuvent être incomplètes, incohérentes et corrompues. Dans cette étape, les valeurs manquantes peuvent être remplies ou supprimées, des valeurs de bruit sont lissées, les valeurs aberrantes sont identifiées et chacune de ces problèmes sont traités par des techniques différentes.
2. **Intégration de données** : Le processus d'intégration de données combine les données provenant de différentes sources. Les données d'entrées généralement

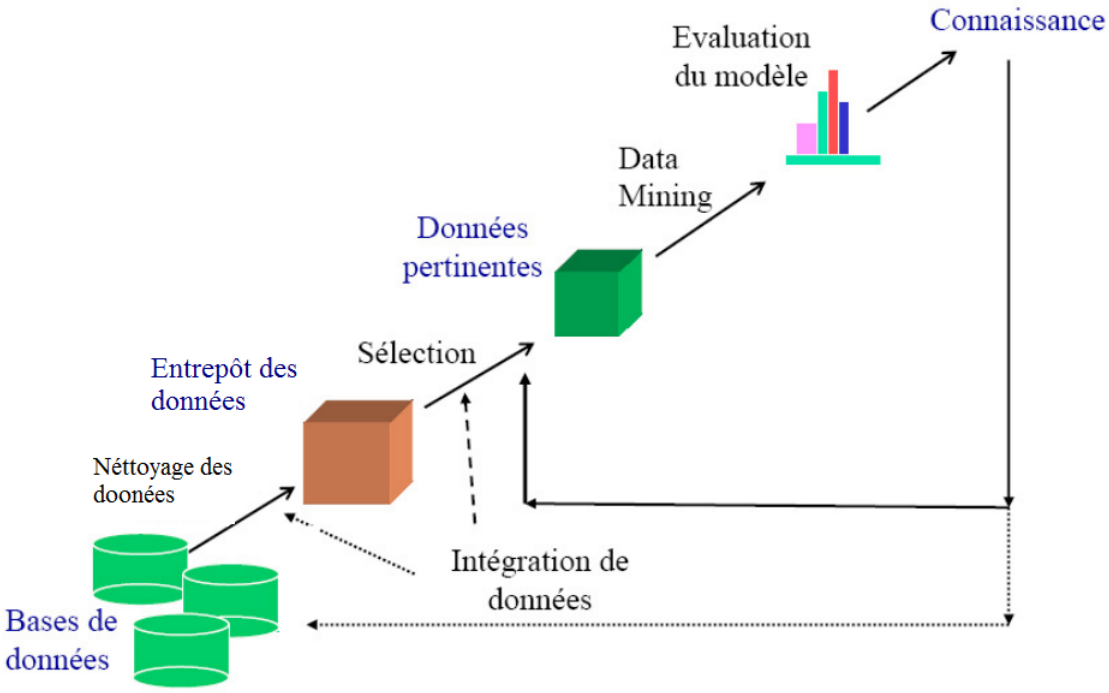

FIGURE 2.1 – Processus du data mining [Fayyad 1996]

sont collectées à partir de plusieurs bases de données ayant chacune une structure et une définition de données distincte. Dans ce cas, le processus d'intégration de données insère les données dans un seul magasin de données (entrepôt des données) cohérent à partir de ces sources de données multiples.

3. **Sélection de données** : A partir des sources de données le processus de sélection de données récupère les données jugées pertinentes pour les utilisées dans les étapes suivantes du processus de l'exploration des données.
4. **Transformation de données** : Dans la phase de transformation, les données sources vont être converties en format approprié pour l'extraction des connaissances. La transformation de données comprend des tâches de base telles que le lissage, l'agrégation, la généralisation, la normalisation et la construction des attributs.
5. **Analyse de données** : Dans le processus d'analyse de données, des méthodes intelligentes sont appliquées afin d'extraire les modèles de données.
6. **Évaluation de modèle** : L'évaluation du modèle est la tâche de découvrir des modèles intéressants parmi un ensemble des modèles et des connaissances extraites.
7. **Présentation de la connaissance** : La représentation des connaissances comprend des techniques de visualisation, qui sont utilisés pour interpréter la connaissance découverte à l'utilisateur.

Le data mining se réfère à l'extraction de connaissances à partir d'une grande bases de données, ce qui pourrait être exprimé dans différents types ; telles que les bases de données relationnelles, les bases de données transactionnelles dans les applications de e-commerce ou des expressions génétiques dans le domaine de la recherche en bioinformatique,... etc.

Le data mining est utilisé dans divers domaines d'application tels que les banques, la biologie, l'e-commerce, l'assurance, etc. D'autre part, les nouvelles applications d'exploration de données incluent le traitement des données spatiales, des données multimédia, les séries temporaires et le Web.

L'exploration de données a été introduite avec succès dans le domaine de la gestion de données Web, dans lequel plusieurs types des données Web, y compris les documents Web, les structures de liaison Web et les transactions des utilisateurs du Web, deviennent les cibles à explorer [Xu 2010]. Il est évident que les connaissances extraites des différents types de données sur le Web peuvent aider à découvrir et à comprendre les relations entre les différents objets Web et peuvent être aussi utilisées au profit de la gestion des données Web.

2.3 Le Web

Le Web a été inventé entre les années 1989-1991 par Tim Berners Lee [Berners-Lee 1994, Berners-Lee 2000], qui, à cette époque, a travaillé au CERN (Centre Européen de la Recherche Nucleaire, ou laboratoire européen pour la physique des particules) en Suisse. Le Web a été conçu pour améliorer la gestion des informations générales des expériences au CERN. Sa suggestion était d'organiser les informations utilisées au sein de cette institution dans une structure qui ressemble à un graphe où les nœuds sont des documents décrivant des objets, tels que des notes, des articles ou des personnes, et les liens sont les relations entre eux, tels que "dépend", "fait partie de", "se réfère à" ou "utilise".

Le Web (ou World Wide Web, WWW, W3, Toile) est un "système hypertexte public contenant des documents liés entre eux par des hyperliens permettant de passer automatiquement d'un document à l'autre". Selon CERN, le World Wide Web est défini comme une "initiative de recherche d'informations hypermédia à grande surface visant à donner un accès universel à un vaste ensemble de documents" [1]. En d'autres termes, c'est la plus grande source d'information qui est facilement accessible et consultable. Il se compose de milliards de documents interconnectés (appelés pages Web) qui sont rédigés par des millions de personnes, dont l'accès à ses documents est très simple en utilisant un réseau mondial appelé *Internet*.

L'implémentation du Web suit un modèle client-serveur standard 2.2. Dans ce

1. http ://info.cern.ch/hypertext/WWW/TheProject.html

modèle, un utilisateur s'appuie sur un programme (appelé client) pour se connecter à une machine distante (appelée serveur) où les données sont stockées. La navigation à travers le Web se fait au moyen d'un programme client appelé *navigateur*, par exemple, Netscape, Internet Explorer, Firefox,... etc. Les navigateurs Web envoi des requêtes à des serveurs distants en utilisant L'URL (en anglais : Uniform Resource Locator, littéralement "Localisateur Uniforme de Ressources"). Ensuite, il interprète les documents retournés qui sont écrits généralement en HTML (Hypertexte Markup Language) et enfin il affiche le contenu sur l'écran du côté client.

Le fonctionnement du Web repose sur la structure de ses documents hypertextes. L'hypertexte permet aux auteurs de pages Web de relier leurs documents à d'autres se trouvant sur des ordinateurs distants (n'importe où dans le monde). Pour visualiser ces documents, on suit simplement des liens (appelés hyperliens). L'idée d'hypertexte a été inventée par Ted Nelson en 1965 [Nelson 1965]. L'Hypertexte permet également de combiner d'autres types de médias, par exemple, image, fichiers audio et vidéo, est appelé hypermédia [Liu 2007].

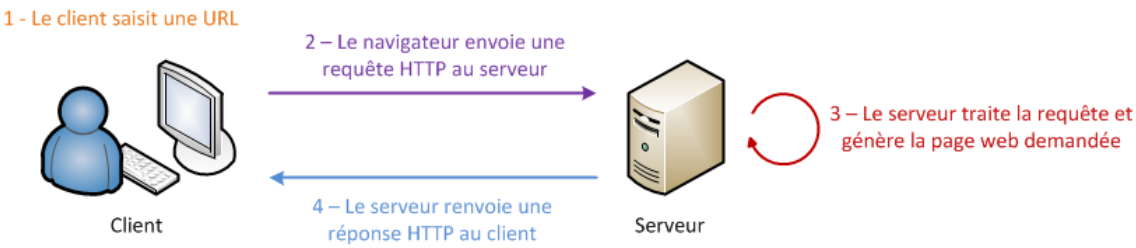

FIGURE 2.2 – Principe de fonctionnement de l'architecture Client-Serveur

La conception originale du Web a impliqué deux éléments centraux, qui sont [Easley 2010] :

1. Il a fourni un moyen de mettre facilement des documents à la disposition de tout le monde sur Internet, sous la forme de pages Web que peuvent être crées et stockées afin d'être accessibles au public.

2. Il a fourni un moyen pour les autres d'accéder facilement à ces pages Web, en utilisant un navigateur qui pourrait se connecter à l'espace public sur les ordinateurs à travers l'Internet et de récupérer les pages Web qui y sont stockées.

Aujourd'hui le défi est de rapporter la sémantique des documents hypertextes (quelque chose qui faisait partie de la proposition original de Berners-Lee), afin de faciliter l'utilisation de la grande quantité d'informations disponibles dans le Web. En d'autres termes, nous avons besoin de transformer les données dans le Web en connaissances du Web.

2.4 Les caractéristiques du Web

Le Web est considéré comme une source très riche pour l'exploration de données parce qu'il contient une collection gigantesque de pages Web (statiques et dynamiques), des informations concernant les hyperliens et l'accès aux pages Web [Han 2011]. En fait, les données sur le Web ont leurs propres caractéristiques par rapport aux bases de données classiques. Les données Web présentent généralement les caractéristiques suivantes [Han 2011, Xu 2010] :

1. La taille du Web est de l'ordre de centaines de téraoctets et est encore en croissance rapide. De nombreuses organisations et sociétés placent la plupart de leurs informations publiques accessibles sur le Web. Alors, la taille énorme de données sur le Web est difficile à gérer à l'aide des techniques de gestion des bases de données traditionnelles.
2. Les pages Web n'ont aucune structure unificatrice. Elles se caractérisent beaucoup de variations de contenus et de styles de création. Jusqu'à présent, il n'y a aucune structure de donnée rigide et uniforme ou des schémas que les pages Web doivent le suivre. En conséquence, il existe une exigence croissante d'exploiter la nature non structurée des documents sur le Web et d'extraire les relations mutuelles cachées dans les données Web pour faciliter aux utilisateurs de trouver les information ou un service Web nécessaires.
3. Le Web est une source d'information très dynamique. Non seulement le Web croître rapidement, mais aussi sa structure, implicite et explicite, est mise à jour fréquemment. Surtout, en raison de différentes applications Web qui génèrent plusieurs formes d'un document Web et les problèmes de réinstallation qui seront produits lorsque les noms de domaine ou de fichiers changent ou disparaissent.
4. Les données sur le Web sont distribuées et hétérogènes. Les données Web sont généralement distribuées à travers un grand nombre d'ordinateurs ou de serveurs, qui sont situés à différents endroits dans le monde. Les données Web ont souvent la même nature que les données multimédia c-à-d en plus de l'information textuelle de nombreux autres types de données existent sur le Web, tels que des images, des fichiers audio et des vidéo sont souvent inclus dans une page Web.
5. Une petite partie des informations dans le Web est réellement pertinente ou utile. Il est dit que 99% des informations sur le Web sont inutiles pour 99% des utilisateurs du Web. Il est vrai qu'une personne donnée est généralement intéressée par une petite partie du Web, alors que le reste du Web contient des informations qui sont sans intérêt pour l'utilisateur et peut enterrer les résultats de recherche souhaités.

Les caractéristiques ci-dessus indiquent que les données Web sont de type différent de bases de données classiques. En conséquence, il y a une demande croissante

de développer des techniques plus avancées pour résoudre le problème de la gestion des données et la recherche d'information sur le Web.

2.5 Exploration du Web

L'exploration du Web (en anglais : Web mining) est l'utilisation des techniques d'exploration de données (data mining) en vue de récupérer, d'extraire et d'évaluer des connaissances, qui peuvent être des constantes, schémas ou modèles, à partir des ressources, à partir de documents et des services Web.

Le terme "*Web mining*" a été introduit pour la première fois par Etzioni dans [Etzioni 1996], en 1996. Le Web mining vise à découvrir des informations utiles ou des connaissances à partir des hyperliens du Web, le contenu des pages Web, et les journaux d'utilisation (Web logs) des sites Web [Singh 2010a].

Le Web mining est un domaine multidisciplinaire impliquant l'extraction de données, l'apprentissage automatique, traitement du langage naturel, les statistiques, les bases de données, la recherche d'information, multimédia, ...etc.

Certains auteurs suggèrent la décomposition du Web mining en un ensemble de sous-tâches successives [Etzioni 1996, Kosala 2000, Kantardzic 2011] (voir Figure 2.3) :

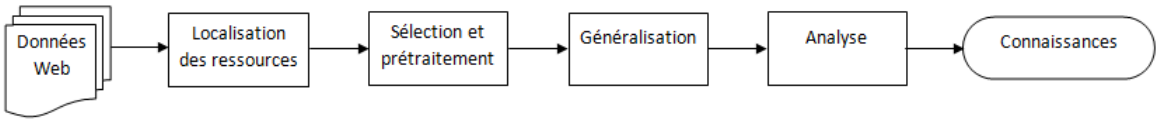

FIGURE 2.3 – Une proposition de décomposition du processus de Web mining en un ensemble des sous tâches successives [Etzioni 1996, Kosala 2000, Kantardzic 2011]

1. **Localisation des ressources** : Cette tâche comprend principalement la représentation des documents, l'indexation et la recherche et récupération des données, qui sont soit en ligne ou hors ligne, à partir des différentes ressources sur le Web. Ces ressources peuvent être des articles de presse, des forums, des blogs et le contenu textuel des documents HTML obtenus en supprimant les balises HTML, ...etc.
2. **Sélection des informations et prétraitement** : Lorsque les documents ont été récupérés le défi consiste à extraire automatiquement des connaissances et d'autres informations requises sans intervention humaine. Un système de prétraitement robuste est nécessaire pour extraire tout type de connaissances à partir d'une grande collection des données non structurées.

Lorsqu'un utilisateur demande une page Web, une variété de fichiers comme des images, sons, vidéos, fichiers exécutables et pages Web, sont accessibles. Une des techniques de prétraitement utilisé pour l'extraction et la sélection des informations est la réduction de dimension qui cherche à transformer les vecteurs de documents originaux à un espace de dimension inférieure, en analysant la corrélation entre les termes dans cette collection de documents. Parmi d'autres techniques de prétraitement il existe par exemple, la suppression des mots vides , la lémmatisation (stemming), la recherche des phrases dans le corpus, ... etc.

3. **Généralisation** : Cette tâche s'intéresse au processus de découverte automatique des modèles généraux au sein des sites Web individuels, ainsi que sur plusieurs sites. Dans cette tâche, les techniques de reconnaissance des formes, d'apprentissage automatique, d'extraction de données et les méthodes orientés Web sont généralement utilisées.
4. **Analyse** : L'analyse est un problème fondée sur les données qui suppose qu'il existe suffisamment de données disponibles, afin que des informations potentiellement utiles peuvent être extraites et analysées. Puisque le Web est un média interactif, les humains jouent un rôle important dans le processus de découverte d'informations ou de connaissances sur le Web. Ce rôle est particulièrement important pour la validation et/ou l'interprétation des modèles (connaissances) extraits qui ont lieu dans cette phase.
5. **Visualisation** : Dans cette tâche, les résultats de l'analyse obtenus sont présentés en mode visuel qui est facile à comprendre.

2.6 Les axes du Web mining

Dans le Web mining les données peuvent être collectées de différentes sources, par exemple du serveur, des clients, les serveurs proxy, ou obtenus à partir des bases de données des organisations ...etc. En fonction du type de la source, le type de données collectées diffères. Ces données disposent également d'une variation extrême du contenu (par exemple, texte, image, audio, symbolique...etc.) et les méta-informations, qui pourraient être disponibles. Cela rend les techniques à utiliser pour des tâches particulières dans le Web mining très diverses.

Selon les types des données Web utilisées dans le processus de l'exploration, le Web mining peut être classé en trois types principaux (voir Figure 2.4) : Exploration de la structure du Web, exploration du contenu du Web et exploration de l'utilisation du Web [Kosala 2000].

L'exploration de la structure du Web vise à découvrir des connaissances à partir liens hypertextes, qui représentent la structure du graphe du Web. L'exploration de contenu du Web a pour objectif l'extraction des informations utiles ou des connais-

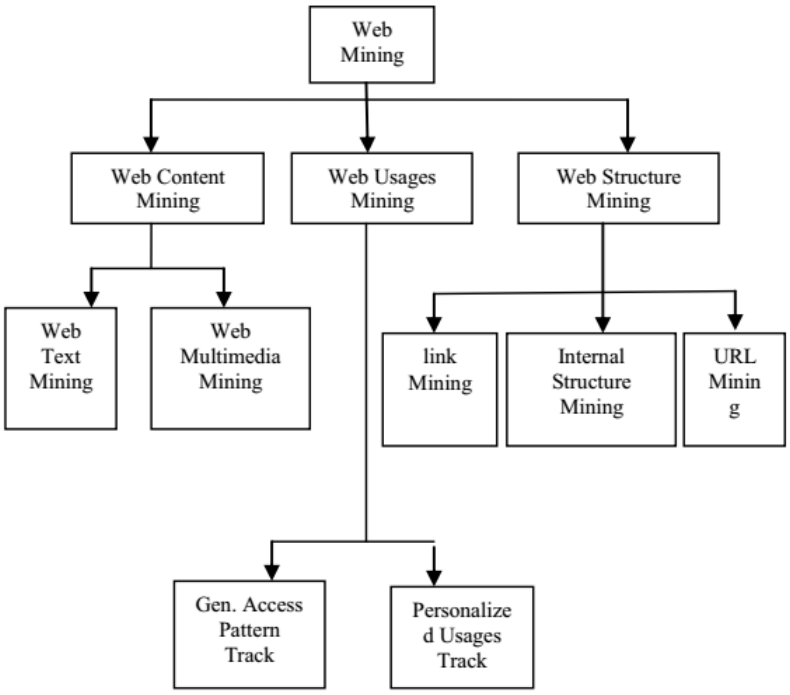

FIGURE 2.4 – Taxonomie du Web mining [Singh 2010a]

sances à partir le contenu des pages Web qui peut être des textes ou des images ...etc. L'exploration de l'utilisation du Web permet la construction des modèles d'accès des utilisateurs à partir des journaux d'utilisations, qui enregistrent les clics effectués par chaque utilisateur.

2.6.1 Analyse de contenu du Web

Le concept analyse de contenu du Web (en anglais : Web content mining) décrit l'extraction des informations utiles ou des connaissances contenues dans les pages Web. Ce domaine a été développé à peu près entre les années 1995 et 2002 [Chakrabarti 2002]. Les travaux de recherches dans ce domaine propose que l'analyse de contenu est basée sur d'autres domaines telles que la recherche d'information (RI) et le Traitement du Langage Naturel (TLN) [Desikan 2002].

Les données contenues dans les pages Web sont considérées comme une collection d'objets utilisés pour transmettre des informations aux utilisateurs. Dans la plupart des cas, ces données peuvent être des textes et d'autres types de contenu multimédia, qui comprennent des pages Web statiques en HTML, des pages XML, des images, des vidéos, des fichiers audio, des pages dynamiques générées par des scripts et des bases de données ... etc. [Johnson 2012, Xu 2010].

Le domaine analyse de contenu du Web comprend la découverte des ressources à partir du Web, la catégorisation, le résumé et le regroupement de documents et l'extraction d'informations à partir de pages Web. Par exemple, le classement automatique ou le regroupement des pages Web en fonction de leurs sujets.

Cependant, il est possible de découvrir des modèles dans les pages Web afin d'extraire, pour de nombreuses raisons, des informations utiles telles que des descriptions des produits, l'affichage des forums, etc. En outre, Il est possible de faire l'extraction des commentaires des internautes et les messages des forums, la découverte des sentiments des consommateurs... etc.

2.6.2 Analyse d'usage du Web

L'analyse d'usage du Web (en anglais : Web Usage Mining) se réfère à la découverte des modèles d'accès des utilisateurs aux sites Web à partir des journaux d'utilisation du Web (Web logs), qui enregistrent chaque clic effectué par chaque utilisateur. L'analyse d'usage du Web fournit des modèles d'utilisation aux organisations afin d'obtenir des profils de clients et, par conséquent, elle peut faciliter la navigation sur le site ou la présentation des produits ou des pages Web spécifiques. Cette dernière a un grand intérêt pour les entreprises, car elle peut par exemple augmenter les ventes si elles offrent produits attrayants les clients.

Bien que l'exploration de contenu utilise les données réelles (ou primaires) sur le Web, l'analyse d'usage du Web explore ce qu'on appelle données "secondaires" qui ont été générées par l'interaction des utilisateurs avec le Web. Les données d'utilisation du Web comprennent des données des journaux d'accès du serveur Web (Web logs), les journaux de serveur proxy, les registres du navigateur, les profils utilisateur, les fichiers d'enregistrement, les sessions utilisateur ou des transactions, les requêtes des utilisateurs, dossiers de signets, des clics de souris, et toutes les autres données générées par l'interaction de utilisateurs et le Web [Pal 2002].

La plupart des outils d'analyse d'utilisation du Web existants fournissent des mécanismes pour rendre compte sur les activités des utilisateurs dans les serveurs et les différentes formes de filtrage des données. L'utilisation de ces outils, permet de déterminer le nombre d'accès au serveur et à des fichiers individuels, les temps de visites, les noms de domaine, les URL des utilisateur, etc [Cooley 1997].

2.6.3 Analyse de la structure du Web

l'analyse de la structure du Web (en anglais : Web structure mining) permet de découvrir, en se basant sur la topologie des liens hypertextes, le modèle qui sous-tend les structures de liaison du Web. Ce modèle peut être utilisé pour classer les pages Web. Il est utile aussi pour générer d'autres informations telles que la similitude et

les relations entre les sites Web. Cet axe est considéré comme une technologie clé utilisée dans la construction des moteurs de recherche.

Les données de structure Web concerne la représentation des relations entre les pages Web, ce qui reflète le concept d'organisation d'un site à partir du point de vue du concepteur. Il est normalement capturé par la structure de liaison inter-page du site, qui est appelé les données de liaisons. En particulier, les données de structure d'un site Web sont généralement représentées par un composant Web spécifique, appelé "plan du site" qui est généré automatiquement lorsque le site est terminé. Pour les pages générées dynamiquement, le plan du site est de plus en plus compliqué à réaliser parce que plusieurs techniques sont nécessaires pour s'adapter à l'environnement dynamique [Xu 2010]. La structure de liaison du Web contient des informations implicites importantes, et peut aider le filtrage ou de classement des pages Web. En particulier, un lien de la *page A* vers la *page B* peut être considéré comme une *recommandation* de *la page B* par l'auteur de *A*.

Certains algorithmes exploitent la structure des liens non seulement pour la recherche par mot clé, mais d'autres tâches comme la construction automatique d'une hiérarchie comme celle de Yahoo, et l'identification des communautés d'utilisateurs qui partagent des intérêts communs...etc.

La performance qualitative de ces algorithmes est généralement mieux que celle des algorithmes de recherche d'informations traditionnels car ils font usage de plus d'informations que seulement le contenu des pages. Il existe deux principaux algorithmes de recherche fondées sur les liens, HITS (Hypertext Induced Topic Research) et PageRank. Ces deux algorithmes sont inspirés du domaine analyse des réseaux sociaux. L'idée de base de l'algorithme HITS [Kleinberg 1999] est d'identifier, selon la requête de l'utilisateur, un petit sous-graphe du Web et d'appliquer l'analyse des liens sur ce sous-graphe pour localiser les autorités et les Hubs pour la requête donnée. Les sélections d'un petit sous-graphe (typiquement quelques milliers de pages), se concentre non seulement sur l'analyse du lien sur la partie la plus pertinente du Web, mais aussi réduit la quantité des travail pour les prochaines phases.

PageRank [Brin 1997] est un algorithme de classement statique des pages Web dans le sens où une valeur de PageRank est calculée pour chaque page hors ligne et il ne dépend pas de requêtes de recherche. PageRank interprète un lien hypertexte de la page A à la page B comme un vote de la page A pour la page B. Sur la base de l'algorithme PageRank, Brin et Page ont construit le moteur de recherche Google.

2.7 La différence entre Web mining et text mining

Par rapport à la classification de texte, la classification des pages Web est différente dans plusieurs aspects. Tout d'abord, les styles des pages Web se diffèrent

d'une manière significative par rapport au documents textuels qui vont être utilisés dans le texte mining. Au lieu de texte brut, la plupart des pages Web utilisent HyperText Markup Language (HTML) pour afficher leur contenu [Chakrabarti 1998, Miner 2012]. Le HTML nous fournit des pages qui sont des combinaisons des textes, des informations de formatage (tels que les tables, les en-têtes, ...etc.), des éléments multimédias (tels que des images ou de la vidéo (y compris la publicité)), des hyperliens vers d'autres pages HTML.

Les hyperliens entre les pages et la structure des documents HTML sont les deux principaux points positifs du Web mining. Les documents Web existent dans un hypertexte, avec des connexions vers, et à partir, d'autres documents. Cette caractéristique est essentielle pour le Web mining et elle n'est pas présente dans la classification du texte classique.

La structure des documents HTML peut également fournir des indices riches aux algorithmes d'exploration. Souvent, les titres et les en-têtes contiennent des mots les plus importants pour décrire le texte. Puisque le HTML marque clairement les entêtes et les titres en utilisant les balises (<header> et </header>), cette information peut facilement être utilisée automatiquement. En outre, HTML définit les balises pour les tables, les listes ordonnées et non ordonnées, ...etc. Ces balises structurelles fournissent des indices pour trouver des mots significatifs sur le document Web [Miner 2012].

En plus de leur contenu principal, la plupart des pages Web comprennent de grandes quantités d'informations qui ne sont pas importantes, comme par exemple les publicités, les éléments de navigation ...etc. Chacun de ces éléments surcharge le contenu de la page par des contenues ne sont pas directement liés à son contenu, mais il est souvent difficile à éviter lors de l'extraction des documents Web.

2.8 Recherche d'informations dans le Web

La Recherche d'informations dans le Web (RIW) a sa racine dans la recherche d'information (RI) classique. La RI classique suppose que l'unité d'information de base est un document, et une grande collection de documents sont disponibles pour former la base de textes. Sur le Web, les documents sont des pages Web. Il est évident de dire que la recherche dans le Web est la application la plus importante de la RI. La recherche dans le Web est, cependant, n'est pas une application pure et simple des modèles RI traditionnels. Il utilise des résultats RI, mais elle a aussi ses techniques uniques et présente de nombreux problèmes pour la RI.

2.8.1 Recherche d'information

Les techniques de la recherche d'information (RI) (en anglais : Information retrieval (IR)) sont directement issues des sciences de l'information et plus précisément de la bibliothéconomie [Bellia 2008]. La recherche d'information est une discipline qui traite la représentation, le stockage, l'organisation et l'accès aux informations. Le but de la recherche d'information est d'obtenir des informations qui pourraient être utiles ou pertinentes pour l'utilisateur [Ceri 2013]. Dans [Christopher 2008], la tâche RI est définie comme étant la découverte des documents non structurés (généralement des textes) répondant à un besoin d'informations contenues dans une collections des documents très vaste, stockés dans les ordinateurs.

L'idée de RI est de récupérer des documents en utilisant un critère booléen simple : la présence ou l'absence des mots spécifiques (mots clés ou termes), dans les documents. Les mots clés peuvent être combinés dans disjonctions et conjonctions, offrant ainsi plus de sens aux requêtes. Une requête par mot clé ne peut pas identifier les documents correspondants, et donc il retourne généralement un grand nombre de documents. Par conséquent, dans RI, il est nécessaire de classer les documents par leur pertinence pour la requête. Le classement selon la pertinence est une différence importante par rapport à l'interrogation de données structurées où le résultat d'une requête est un ensemble (de collection non ordonnée) des données.

Actuellement, la recherche d'information s'est développée et est devenue un domaine transdisciplinaire. Par exemple, La recherche d'information multimédia dont un document multimédia combine des données de différents types (texte, image, audio, vidéo). La RI a connu aussi une extension au domaine des bases de données dédiées à un accès local, ou bien mises en réseau reliées par des liens hypertextes comme sur le Web [Bellia 2008].

2.8.2 Processus de RI

Un système de recherche d'information manipule un corpus de documents qu'il va être transformé, à l'aide d'une fonction d'indexation, en un corpus indexé. Ce corpus lui permet de résoudre des requêtes traduites à partir de besoins des utilisateurs. Un tel système repose sur la définition d'un modèle de recherche d'information qui effectue ces deux transformations et qui fait correspondre les documents aux requêtes. La transformation d'un document en un document indexé repose sur un modèle de document. De même, la transformation du besoin utilisateur en requête repose sur un modèle de requête. Enfin, la correspondance entre une requête et des documents s'établit par une relation de pertinence [Bouramoul 2011].

La figure 2.5, présente le déroulement d'un processus de recherche d'information. L'utilisateur émettre une requête (requête utilisateur) au SRI via une interface uti-

lisateur, sous la forme d'une requête textuelle (typiquement composée de certains mots clés). Cette requête est analysée et transformée par un ensemble d'opérations textuelles. Cette étape donne une requête normalisée. Le module de recherche utilise l'index du document pour récupérer les documents qui contiennent des termes de requête (ces documents sont susceptibles d'être pertinents pour la requête).

Le SRI calcule un score de pertinence pour chaque document indexé par rapport à la requête. Cette étape s'appelle l'appariement. Selon leurs scores de pertinence, les documents sont classés et présentés à l'utilisateur. Notez qu'il ne compare généralement pas la requête utilisateur à chaque document de la collection, ce qui est trop inefficace. Au lieu de cela, seulement un petit sous-ensemble des documents contenant au moins un terme de requête est d'abord trouvé dans l'index et les scores de pertinence avec la requête utilisateur sont ensuite calculés uniquement pour ce sous-ensemble de documents.

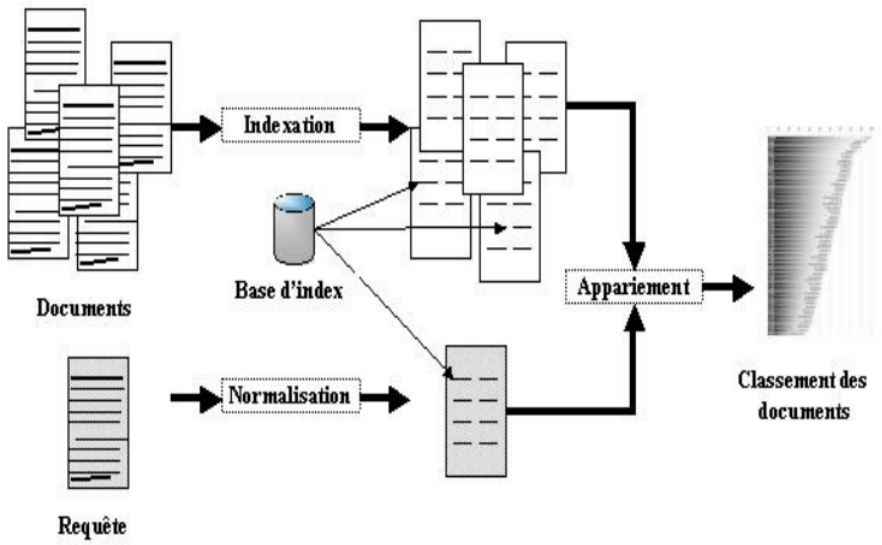

FIGURE 2.5 – Processus de recherche d'informations

2.8.3 Modèles de RI

Un modèle de RI s'intéresse à la façon de représentation d'un document et d'une requête et comment définir la pertinence d'un document à une requête de l'utilisateur. Généralement, Il existe plusieurs modèles de RI. Les modèles les plus utilisés pour représenter l'ensemble des pages Web sont : le modèle booléen, le modèle vectoriel, et le modèle probabiliste.

Il faut noter la plupart des modèles de RI représentent des documents et des requêtes comme un "sac" de mots (ou de termes), c-à-d, un ensemble de termes distinctifs. Notons que le terme cette fois-ci peut ne pas être un mot du langage

naturel dans un dictionnaire. Chaque terme est associé à un poids. soit la collection de documents D, $V = t_1, t_2, \cdots, t_{|V|}$ l'ensemble des termes distinctifs de la collection, où t_i est un terme. L'ensemble V est généralement appelé le vocabulaire de la collection, et $|V|$ est sa taille, c-à-d le nombre de termes dans V. Un poids $w_{ij} > 0$ est associé à chaque terme t_i d'un document $d_j \in D$. Pour un terme qui ne figure pas dans le document d_j, $w_{ij} = 0$. Chaque document d_j est donc représenté par un vecteur $dj = (w_{1j}, w_{2j}, ..., w_{|V|j})$. Avec cette représentation vectorielle, une collection de documents est simplement représenté comme une table relationnelle (ou une matrice). Chaque terme est un attribut, et chaque poids est une valeur d'attribut. Dans les différents modèles de RI , le poids w_{ij} est calculé différemment.

2.8.3.1 Modèle booléen

Le modèle booléen [Salton 1969] est l'un des modèles de recherche d'information les plus anciens et les plus simples. Il repose sur la théorie des ensembles et de l'algèbre de Boole, de sorte que les requêtes sont définies comme des expressions booléennes en utilisant les opérateurs booléens AND, OR et NOT [Ceri 2013]. Ce modèle utilise la notion de correspondance exacte pour faire correspondre des documents à la requête de l'utilisateur.

En utilisant la représentation vectorielle du document ci-dessus(section 2.8.3), le poids w_{ij} du terme t_i dans le document d_j est 1 *si* t_i apparaît dans le document t_i, et 0 sinon, comme suit :

$$w_{ij} = \begin{cases} 1 & si \quad t_i \in d_j \\ 0 & sinon \end{cases} \tag{2.1}$$

Une requête booléenne q peut être calculée en récupérant tous les documents contenant les termes et la construction d'une liste pour chaque terme. Une fois que ces listes sont disponibles, les opérateurs booléens doivent être traitées comme suit :

- q_1 *OR* q_2 exige la construction de l'union des listes de q_1 et q_2 ;
- q_1 *et* q_2 nécessite la construction de l'intersection des listes de q_1 et q_2 ;
- q_1 *AND NOT* q_2 nécessite la construction de la différence des listes de q_1 et q_2.

Les systèmes de recherche basés sur le modèle booléen récupère chaque document qui rend la requête logiquement vrai. Ainsi, la récupération est basée sur le critère de décision binaire, c-à-d un document est soit pertinente ou non, ce qu'on appelle **correspondance exacte**.

Par exemple ; le calcul de l'ensemble des résultats de la requête t_a *AND* t_b implique les cinq étapes suivantes :

1. localiser le terme t_a dans le dictionnaire ;
2. récupérer la liste L_a des documents contenant t_a ;

3. localiser le terme tb dans le dictionnaire ;
4. récupérer la liste L_b des documents contenant t_b ;
5. faire l'intersection de L_a et L_b

Cependant, les limites du système booléen sont bien connues [Salton 1983] :
— La taille de la sortie obtenue en réponse à une requête donnée est difficile à contrôler ; en fonction de la fréquence d'attribution des termes et les combinaisons des termes utilisés dans cette requête, peut produire un grand ensemble de sorties ou, alternativement, aucune sortie peut être récupérée.
— Aucune pondération des termes n'est possible, tous les termes ont la même importance.
— La formulation de requêtes booléennes peut produire des résultats contre-intuitifs : par exemple, en réponse à une requête *ou* ("*A ou B ou ... ou Z*"), un document contenant un seul terme de requête est considéré aussi important que un document contenant tous les termes de la requête ; de même, étant donné une requête *et* ("*A et B et ... et Z*"), un document contenant tous les termes sauf un alors le document est supposé inutile comme un document qui ne contient aucun des termes de la requête.
— Les résultats de la recherche dépendent du degré de maîtrise des opérateurs booléens alors qu'il n'est pas toujours évident de traduire un besoin exprimé en langue naturelle à l'aide des opérateurs logiques.

2.8.3.2 Modèle vectoriel

Le modèle vectoriel introduit par Salton [Salton 1975], repose sur les bases mathématiques des espaces vectoriels. Le modèle d'espace vectoriel définit les documents et les requêtes utilisateurs en tant que vecteurs (ou points) dans un espace euclidien multidimensionnel où les axes (dimensions) sont représentés par des termes d'indexation $t_1, t_2, \cdots, t_N$, Où N est le nombre total de termes issus de l'indexation de la collection des documents. Les termes de poids nul représentent les termes absents dans le document alors que les poids positifs représentent les termes existants dans ce document.
Le document j est représenté par le vecteur : d_j $d_j = (w_{1j}, w_{2j}, \cdots, w_{Nj})$
La requête q est représentée par un vecteur : $q = (w_{1q}, w_{2q}, \cdots, w_{Nq})$
Où w_{iq} et w_{ij} sont des poids attribués à des termes t_i pour la requête q et pour chaque document d_j, selon un modèle de pondération (les poids) choisie.

Schéma de pondération Un document dans le modèle vectoriel est représenté comme un vecteur de poids, dans lequel chaque poids de composant est calculé en fonction de certaines variations de TF (Term Frequency) ou le modèle TF-IDF(Term Frequency-Inversed Document Frequency).

Term Frequency(TF) Dans l'approche fréquence du terme (en anglais : Term Frequency (TF)), les coordonnées de vecteur du document d_j sont représentés par le nombre des occurrences d'un terme t_i, généralement normalisé avec le nombre des termes contenus dans ce document. Pour chaque terme t_i et chaque document d_j, la $TF(t_i, d_j)$ est calculée de différentes manières, par exemple :

— En utilisant le nombre total de termes dans le document [Markov 2007] :

$$TF(t_i, d_j) = \begin{cases} \frac{n_{ij}}{\sum_{k=1}^{k=m} n_{kj}} & si \quad n_{ij} > 0 \\ 0 \quad si \qquad n_{ij} = 0 \end{cases} \tag{2.2}$$

— En utilisant le maximum des nombres d'occurrences des termes dans le document :

$$TF(t_i, d_j) = \begin{cases} \frac{n_{ij}}{max(n_{kj})} & si \quad n_{ij} > 0 \\ 0 \quad si \qquad n_{ij} = 0 \end{cases} \tag{2.3}$$

— En utilisant l'échelle logarithmique pour conditionner le nombre des termes (cette approche est utilisée dans le système Cornell SMART) :

$$TF(t_i, d_j) = \begin{cases} \frac{n_{ij}}{1 + log\ (1 + log\ n_{ij})} & si \quad n_{ij} > 0 \\ 0 \quad si \qquad n_{ij} = 0 \end{cases} \tag{2.4}$$

L'inconvénient du système TF est que tous les termes sont considérés comme tout aussi important dans l'évaluation de la pertinence d'une requête. En fait, certains termes ont peu ou pas de pouvoir de discrimination dans la détermination de la pertinence. Par exemple, une collection de documents sur l'industrie automobile est susceptible d'avoir le terme auto dans presque tous les documents [Xu 2010] .

Term Frequency-Inversed Document Frequency (TF-IDF) : L'idée de base de l'approche Fréquence inverse de document (IDF) consiste à réduire les coordonnées de certains axes, correspondant à des termes qui existent dans beaucoup documents. Pour chaque terme t_i la mesure IDF est calculée en proportion des documents où t_i est apparu par rapport au nombre total de documents du corpus. La fréquence inverse de document (notée idf_i) du terme t_i est donnée par :

$$idf_i = \quad log \frac{N}{df_i} \tag{2.5}$$

Où N est le nombre de tous les documents dans le corpus et df_i est le nombre de documents dans lesquels le terme t_i apparaît au moins une fois. Le poids TF-IDF d'un terme est donné par :

$$w_{ij} = TF_{ij} \times IDF_i \tag{2.6}$$

Mesure de similarité Contrairement au modèle booléen, le modèle vectoriel ne prend pas une décision si un document est pertinent pour une requête donnée ou non. Au lieu de cela, les documents sont classés en fonction de leur degré de pertinence pour la requête. Une façon de calculer le degré de pertinence est de calculer la similarité de la requête q pour chaque document d_j dans la collection de documents D. Il existe de nombreuses mesures de similarité.

La mesure de JACCARD compare le nombre l'attributs communs avec le nombre l'attributs uniques pour une paire de mots. Elle a été généralisée par Grefenstette [Grefenstette 1994] en remplaçant l'intersection avec le poids minimum et l'union avec le poids maximum :

$$sim(d_j, q) = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sum_{i=1}^{|V|} w_{ij}^2 + \sum_{i=1}^{|V|} w_{iq}^2 - 2\sum_{i=1}^{|V|} w_{ij} \times w_{iq}} \tag{2.7}$$

Le produit scalaire des deux vecteurs peut être utilisé pour mesurer la similarité entre le document et la requête :

$$sim(d_j, q) =< d_j \bullet q > \tag{2.8}$$

La Mesure de similarité la plus connue est la similitude **cosinus**, qui est le cosinus de l'angle entre le vecteur de la requête q et le vecteur de document d_j :

$$\cos(d_j, q) = \frac{< d_j \bullet q >}{||d_j|| \times ||q||} = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}^2}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}}} \tag{2.9}$$

Il faut noter que un document peut avoir un score élevé de similarité à une requête, même si elle ne contient pas tous les mots clés. Les documents de haut classement sont ensuite présentés à l'utilisateur. Le processus peut alors être répété si l'utilisateur souhaite raffiner la requête.

Parmi les avantages de modèle vectoriel, nous trouvons par exemple : l'amélioration des performances de la recherche par rapport au modèle booléen grâce au système de pondération. La correspondance partielle est autorisée, ce qui donne un classement naturel des documents retrouvés. L'inconvénient majeur de ce modèle réside dans la considération des termes comme étant mutuellement indépendants mais malheureusement c'est pas le cas en réalité.

2.8.3.3 Modèle probabiliste

Le modèle probabiliste a été proposé par Robertson et Sparck Jones [Robertson 1976]. Ce modèle mathématique est fondé sur la théorie de la probabilité conditionnelle

(appelé aussi modèle de recherche de l'indépendance binaire) [Maxwell 2014]. Dans le modèle probabiliste, au lieu d'être basées sur une arithmétique vectorielle relativement abstraite, les approches probabilistes nécessitent un ensemble de documents obtenus en demandant aux utilisateurs de fournir des jugements de pertinence par rapport aux résultats de requête [Gudivada 1997, Nettey 2006]. La base de l'apprentissage est utilisée pour calculer le poids du terme par l'estimation des probabilités conditionnelles qu'un terme se produit dans un document pertinent (ou non pertinent).

Les méthodes probabilistes génèrent des indexes complexes basés sur des informations sur la dépendance des termes. Puisque, cela nécessite la prise en considération d'un nombre exponentiel de combinaisons des termes, et pour chaque combinaison, il exige l'estimation, des probabilités de coïncidences dans les documents pertinents et non pertinents, seules certaines paires de termes dépendants sont pris en compte dans la pratique. En théorie, ces dépendances peuvent être pour des utilisateurs spécifiques [Gudivada 1997].

Le modèle probabiliste considère que les termes d'indexations indépendants c'est-à-dire que leur probabilité d'apparition est la même avec ou sans la présence des autres termes. Sous cette hypothèse, on cherche à estimer la probabilité qu'un document soit pertinent par rapport à une requête.

Soit la requête q composée de séquence de termes, $q = q_1 q_2 \cdots q_m$, et un ensemble de documents $D = d_1, d_2, \cdots, d_N$. $P(PERT)$ est la probabilité de pertinence, c'est-à-dire, la chance de tomber sur un document pertinent, si on choisit un document au hasard dans le corpus. Et inversement, $P(NPERT)$ est la probabilité d'avoir un document non pertinent si on réalise un tirage au hasard. $P(D)$ est la probabilité d'extraire le document D du corpus.

Le modèle probabiliste tente d'estimer la probabilité $P(PERT|D)$ (resp. $P(NPERT|D)$) qu'un document d appartienne à la classe des documents pertinents (resp. non pertinents). Autrement dit, on observe la pertinence ou la non pertinence sachant le document D. Seules la présence et l'absence de termes dans les documents et dans les requêtes sont considérées comme des caractéristiques observables. Autrement dit, les termes ne sont pas pondérés, mais prennent seulement les valeurs 0 (absent) ou 1 (présent).

Lorsque l'indépendance des mots est supposée, le modèle probabiliste sera simplement une classification binaire de type Bayes Naïf, qui vise à classer les documents comme pertinents ou non pertinents.

En utilisant la règle de Bayes, un document est considéré comme pertinent si :

$$P(D|PERT)P(PERT) > P(D|NPERT)P(NPERT) \tag{2.10}$$

Cela équivaut à :

$$\frac{P(D|PERT)}{P(D|NPERT)} > \frac{P(NPERT)}{P(PERT)} \tag{2.11}$$

Donc, les documents sont ordonnés en fonction de la partie gauche de l'équation 2.11. C'est le rapport de vraisemblance, donc les modèles probabilistes sont également connus sous le nom de modèles de vraisemblance de documents. En d'autres termes :

$$P(PERT|R, D) = \frac{P(D|PERT)}{P(D|NPERT)} \tag{2.12}$$

Les documents sont ensuite classés selon :

$$P(PERT|R, D) = \frac{P(D|PERT)}{P(D|NPERT)} \tag{2.13}$$

Un modèle plus sophistiqué que le précédent calcule $P(D|PERT)$ à partir des termes apparaissant dans D_i.

$$P(D|PERT) = \prod_{t_j \in D_i} P(t_j|PERT) \times \prod_{t_j \in D_i} (1 - P(t_j|PERT)) \tag{2.14}$$

$$P(D|NPERT) = \prod_{t_j \in D_i} P(t_j|NPERT) \times \prod_{t_j \in D_i} (1 - P(t_j|NPERT)) \tag{2.15}$$

La valeur de $P(D_i|PERT)$ est estimée comme étant le produit des probabilités associées à chaque terme dans le document, multipliées par le produit des probabilités que les termes absents n'apparaissent pas dans un document pertinent.

$$\begin{aligned} log\frac{P(D|PERT)}{P(D|NPERT)} = &\sum_{t_j \in D_i} log(P(t_j|PERT)) + \sum_{t_j \in D_i} log(1 - P(t_j|PERT)) \\ &- \sum_{t_j \in D_i} log(P(t_j|NPERT)) - \sum_{t_j \in D_i} log(1 - P(t_j|NPERT)) \end{aligned} \tag{2.16}$$

La valeur $log\frac{P(D|PERT)}{P(D|NPERT)}$ est appelée *valeur de statut de recherche* et le but est de trouver les documents qui la maximisent. Le problème est maintenant d'estimer $P(t_j|PERT)$ et $P(t_j|NPERT)$.

Il existe deux manières d'estimer $P(t_j|PERT)$ et $P(t_j|NPERT)$. L'estimation a priori de ces facteurs est une technique qui suggère de donner une valeur fixe

à $P(t_j|PERT)$ et de calculer $P(t_j|NPERT)$ en fonction de sa distribution dans l'ensemble des documents. Plus un terme t_j est rare plus $P(t_j|NPERT)$ est basse et vice versa.

Le deuxième mode d'estimation consiste à établir un échantillonnage des documents selon leur pertinence. Supposons qu'il existe NDP documents pertinents et que NDP_{t_j} soit le nombre de documents pertinents contenant t_j. On a alors :

$$P(t_j|PERT) = \frac{NDP_{t_j}}{NDP} \tag{2.17}$$

Par ailleurs, ND_{t_j} est le nombre des documents contenant t_j et ND est le nombre de documents total. $P(t_j|NPERT)$ est calculée par :

$$P(t_j|NPERT) = \frac{ND_{t_j} - NDP_{t_j}}{ND - NDP} \tag{2.18}$$

Le problème de ce modèle réside dans le passage de l'estimation de la pertinence des documents à l'estimation de la pertinence des termes qui repose sur l'hypothèse de l'indépendance des termes [Bellia 2008].

2.8.4 Évaluation de la recherche d'information

Il existe plusieurs mesures pour évaluer la performance des systèmes de recherche d'information. Ces mesures nécessitent une collection de documents et d'une requête. Ces mesures supposent une notion de pertinence c-à-d chaque document soit pertinent ou non pertinent pour une requête particulière.

Rappel est la proportion des documents qui sont pertinents à la requête qui sont récupérées avec succès.

$$Rappel = \frac{Nombre\ de\ documents\ pertinents\ restitués}{Nombre\ de\ documents\ pertinents} \tag{2.19}$$

$$Rappel = \frac{VraiPositifs}{VraiPositifs + FauxNégatifs} \tag{2.20}$$

Précision est la proportion des documents récupérés qui sont pertinents pour les besoins de l'utilisateur.

$$Précision = \frac{Nombre\ de\ documents\ pertinents\ restitués}{Nombre\ de\ documents\ restitus} \tag{2.21}$$

$$Précision = \frac{VraiPositifs}{VraiPositifs + FauxNégatifs} \tag{2.22}$$

Où :

— *Positifs* : Ensemble des documents pertinents
— *Négatifs* : Ensemble des documents qui ne sont pas pertinents.
— *Vraipositifs* et *Vrainégatifs* : Désignent les documents qui ont été correctement classés par le classificateur.
— *Fauxpositifs* et *Fauxnégatifs* : Désignent les documents qui ont été mal classés par le classificateur.

F-mesure examine la succession des précisions à n ou des rappels à n, et permet d'évaluer un ordonnancement dans son ensemble et ainsi de déterminer telle ou telle propriété de l'algorithme étudié. Il peut être intéressant d'avoir une valeur synthétisant ces deux mesures. On voudrait pouvoir maximiser la précision et le rappel, mais comme on l'a vu ces deux mesures évoluent souvent de façon opposée. La précision est globalement décroissante au fur et à mesure que le SRI restitue des documents, alors que le rappel est globalement croissant. On peut choisir la F-mesure comme valeur synthétique exploitant la précision et le rappel. Elle est calculée comme suit :

$$F = \frac{(2 \times Rappel \times Précision)}{(Rappel + Précision)} \tag{2.23}$$

2.8.5 Moteurs de recherche

Les moteurs de recherche sont largement utilisés pour l'accès à l'information Web et ils rendent les informations accessibles plus facilement . Les dernières décennies, comptent beaucoup de projets des moteurs de recherche dans le Web. Le système Excite a été introduit en 1993 par des étudiants de l'Université de Stanford. EINet Galaxy a été créé en 1994 dans le cadre du Consortium de recherche MCC à l'Université du Texas. Jerry Yang et David Filo ont créé Yahoo ! en 1994, qui a commencé comme une liste de leurs sites Web favoris. Dans les années suivantes, de nombreux systèmes de recherche ont émergé, par exemple, Lycos, Inforseek, AltaVista, Inktomi, Ask Jeeves, Northernlight, etc. Google a été lancé en 1998 par Sergey Brin et Larry Page en se basant sur leur projet de recherche à l'Université de Stanford. Microsoft a commencé la recherche de ce domaine en 2003, et a lancé le moteur de recherche MSN au printemps 2005 [Liu 2007]. Yahoo ! a fourni une capacité de recherche générale en 2004 après avoir acheté Inktomi en 2003. Un moteur de recherche commence par l'exploration de pages sur le Web. Les pages explorées sont ensuite analysés, indexés et stockés. Le processus d'un moteur de recherche peut être résumé dans les étapes suivantes :

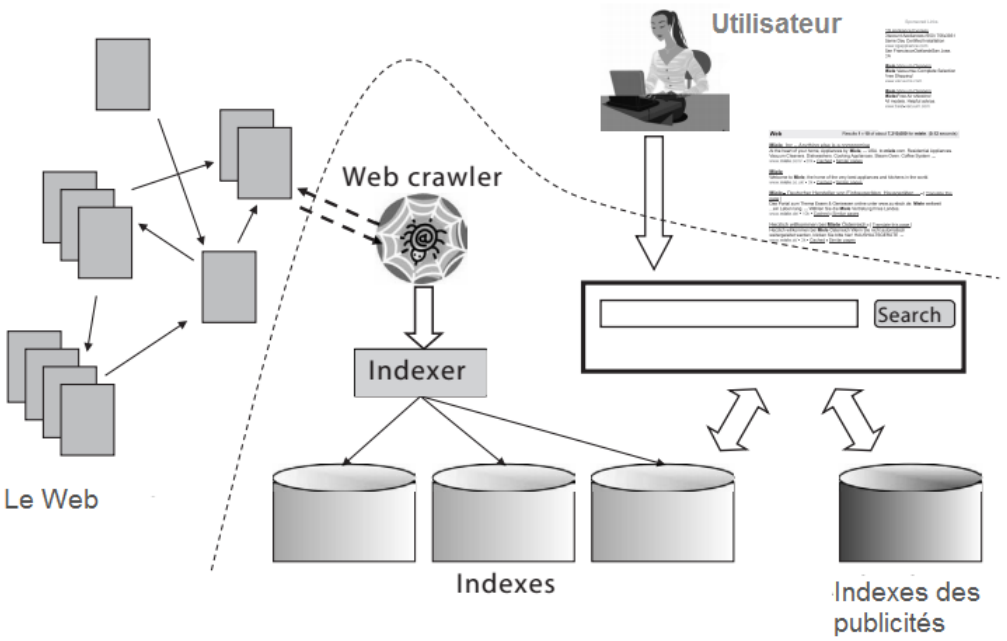

FIGURE 2.6 – Les différents éléments d'un moteur de recherche [Christopher 2008]

1. Certaines tâches de pré-traitement peuvent être effectuées avant ou après l'analyse.

2. Un analyseur lexical et syntaxique est utilisé pour analyser la page HTML, ce qui produit un ensemble des termes à indexer.

3. Indexation : Cette étape produit un index inversé. Pour plus d'efficacité, le moteur de recherche peut construire plusieurs indexes inversés. Par exemple, étant donné que les titres et les ancres sont souvent des descriptions très précises des pages, un petit index inversé peut être construit sur la base des termes est apparu en eux seuls. Un index complet est alors construite sur la base de tout le texte dans chaque page, y compris les ancres. Dans la recherche, l'algorithme peut rechercher dans le petit index d'abord, puis l'index complet.

4. Recherche et Classement : L'algorithme de classement est le cœur d'un moteur de recherche. Cependant, Il n'y a pas beaucoup de détails sur les algorithmes utilisés dans les moteurs de recherche commerciaux. La littérature se base généralement sur l'algorithme du système ancien de Google [Brin 1997].

2.9 Classification des pages Web

La classification des pages Web, également connue sous la catégorisation des pages Web, peut être définie comme la tâche consistant à déterminer si une page Web appartenant à une ou plusieurs catégories [Choi 2005]. Formellement, soit $C = c_1, \cdots, c_K$ un ensemble de catégories prédéfinies, $D = d_1, \cdots, d_N$ un ensemble de pages Web à classer, et $A = D \times C$ une matrice de décision (Voir Figure 2.7) :

Web Pages	Categories				
	c_1	...	c_j	...	c_K
d_1	a_{11}	...	a_{1j}	...	a_{1K}
...	...	...	...	...	...
d_i	a_{i1}	...	a_{ij}	...	a_{iK}
...	...	...	...	...	...
d_N	a_{N1}	...	a_{Nj}	...	a_{NK}

FIGURE 2.7 – La forme matricielle de la catégorisation des pages Web [Choi 2005]

chaque entrée a_{ij} indique si la page Web d_i appartient à la catégorie c_j ou non, c-à-d a_{ij} égale à 1 si la page Web d_i appartient à la catégorie c_j, et 0 sinon. Une page Web peut appartenir à plus d'une catégorie.

Qi et Davidson dans [Qi 2009] ont définit la classification des pages Web "s'est le processus d'affectation d'un ou plusieurs étiquettes de catégorie prédéfinie à une page Web". La classification des pages Web est souvent posée comme un problème d'apprentissage supervisé dans lequel un ensemble de pages Web étiquetées est utilisé pour créer un classificateur permettant ensuite de classer des nouvelles pages Web .

2.9.1 Types des classifications

La classification des pages Web peut être divisée en plusieurs sous-problèmes : la classification des sujets, la classification fonctionnelle, la classification des sentiments, ...etc [Qi 2009]. La classification des sujets s'intéresse au sujet d'une page Web. Par exemple, juger si une page est sur "les arts", "politiques" ou "sport" etc. La classification fonctionnelle concerne le rôle d'une la page Web. Par exemple, décider si une page est une "page personnelle", "page du cours", "page d'accueil ...etc. La classification des sentiments se concentre sur l'avis qui est présenté dans une page Web, c-à-d, les orientations de l'auteur.

La classification selon [Qi 2009] peut être divisée en classification binaire et

multi-classes. La classification binaire catégorise les instances en une des deux classes (voir Figure 2.8), par contre la classification de multiclasses manipule plus de deux classes (voir Figure 2.9).

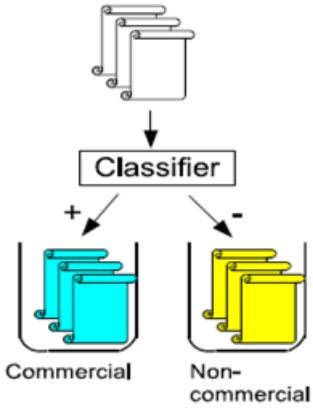

FIGURE 2.8 – Classification binaire [Qi 2009]

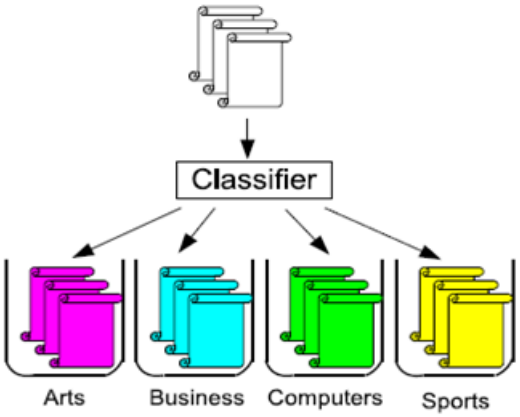

FIGURE 2.9 – Classification multi-classes [Qi 2009]

En se basant sur le nombre de classes qui peuvent être assignées à une page Web, la classification peut être divisée en classification en une seule étiquette et multi-étiquette. En classification en une seule étiquette, une et une seule étiquette d'une classe doit être affectée à chaque instance, mais dans la classification en multi-étiquettes, plus d'une étiquette peuvent être affectées à une instance.

Basée sur le type d'attribution de la classe, la classification peut être divisée en classification dure et douce. Dans la classification dure, une instance peut être ou ne pas être dans une classe particulière, tandis que dans la classification douce, une instance peut être prédite d'être dans une certaine classe avec une certaine probabilité (souvent une distribution de la probabilité dans toutes les classes).

La classification des pages Web peut être également divisée en classification plate et de classification hiérarchique. Dans la classification plate, les catégories sont considérées en parallèle, c'est à dire, une catégorie ne remplace pas l'autre (voir Figure 2.10). Alors que dans la classification hiérarchique, les catégories sont organisées en une structure hiérarchique arborescente, dans lequel chaque catégorie peut avoir un certain nombre de Sous-catégorie (voir Figure 2.11).

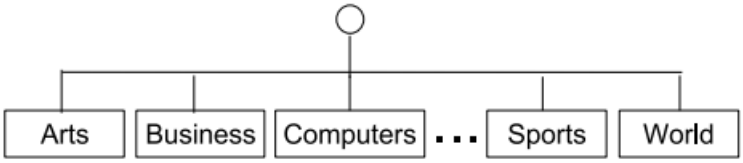

FIGURE 2.10 – Classification plate [Qi 2009]

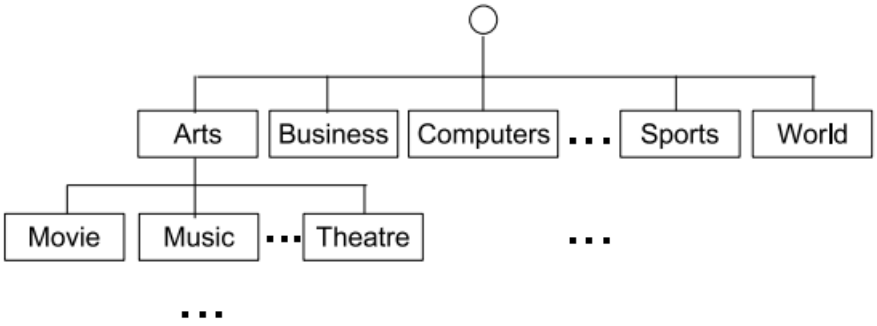

FIGURE 2.11 – Classification hiérarchique [Qi 2009]

2.9.2 Domaines d'application de la classifications des pages Web

La classification des pages Web est très utilisée dans plusieurs domaines comme la construction et la maintenance des annuaires des pages Web, l'amélioration des réponses des moteurs de recherches, l'amélioration des performances des systèmes des questions/réponses et même peut être utilisé pour améliorer l'efficacité et la concentration des robots aspirateurs (ou les Web crawlers) et le filtrage des emails [Qi 2009].

2.9.2.1 Construction des annuaires Web

Un annuaire Web, répertoire Web, annuaire Internet ou répertoire Internet est un site Web proposant une liste classée de sites Web. Le classement se fait selon une arborescence de catégories, afin de couvrir tous ou partie des domaines d'intérêt

des visiteurs. Les annuaires peuvent être généralistes, spécialisés (thématiques) ou géographiques.

— Les généralistes n'excluent, à priori, aucun centre d'intérêt ;
— Les annuaires spécialisés et thématiques se concentrent exclusivement sur les sites ou les pages Web traitant d'un certain sujet, ou destinés à un certain public ;
— Les annuaires géographiques, peuvent à la fois se révéler généralistes ou spécialisés ; dans les deux cas, ils sont relatifs à un pays, une région, ..etc.

Parmi les annuaires du Web, nous trouvons ce fourni par Yahoo ! (2007) et le Projet de dmoz Open Directory (ODP) (2007), Actuellement, ces répertoires sont principalement construits et entretenus par les éditeurs, ce qui nécessite beaucoup des efforts humains.

2.9.2.2 Amélioration de la recherche dans le Web

La recherche d'information est la science qui étudie la manière de répondre pertinemment à une requête en retrouvant toutes les informations demandées dans un corpus. L'ambiguïté de requête est l'un des problèmes qui peut perturber la qualité des résultats de recherche. Plusieurs approches ont été proposées pour améliorer la qualité de la recherche par désambiguïser les termes de la requête. La littérature comptent plusieurs travaux qui ont proposé l'utilisation de la classification automatique des pages Web dans l'amélioration de la précision des résultats de la recherche dans le Web. Nous citons par exemple [Chekuri 1997, Chen 2000, Haveliwala 2002, Käki 2005, Kohlschütter 2007]

2.9.2.3 Amélioration du système de question-réponse

Un système de question-réponse peut utiliser des techniques de classification afin d'améliorer sa qualité de réponses. les auteurs de [Yang 2004] ont suggéré de trouver des réponses aux questions en utilisant la classification fonctionnelle des pages Web.

Soit une question, un certain nombre de requêtes sont formulées et envoyé à des moteurs de recherche. Les pages Web dans les résultats sont récupérés et ensuite classés par des classificateurs par exemple arbre de décision. Afin d'accroître la couverture, pages thématiques supplémentaires sont inclus en suivant les liens sortants des pages collectées. Après cela, les pages thématiques sont regroupés, à partir de lesquelles les réponses sont extraites. (voir aussi [Kwok 2001, Katz 2003])

2.9.2.4 Amélioration de l'efficacité des Web crawlers

Un Web Crawler est un programme ou un script automatisé conçu pour parcourir les pages sur l'Internet d'une manière méthodique et automatisés afin de créer un index des données qu'il cherche. En tant que tel l'efficacité d'un moteur de recherche est directement liée à la performance de son Web crawler. Ceci explique pourquoi un Web crawler est très important pour l'optimisation des moteurs de recherche sur Internet.

Chakrabarti et al [Chakrabarti 1999] ont proposé une approche appelée exploration automatisée ciblée (focused crawling), dans lequel uniquement les documents pertinents à un ensemble prédéfini de sujets seront récupérés. Dans cette approche, un classificateur est utilisé pour évaluer la pertinence d'une page Web par rapport un sujet donnée.

2.9.2.5 Filtrages des emails

Les Spams sont des courriers commerciaux ou des emails non désirables qui sont envoyés à plusieurs utilisateurs et qui occupent leurs boîtes de réception et provoquent une mauvaise utilisation du temps et des ressources de calcul, ce qui conduit à des pertes financières. La solution la plus applicable à ce problème est le filtrage anti-spam. La plupart des filtres anti-spam sont basés sur les techniques d'apprentissage [Androutsopoulos 2000, Yue 2007, Carreras 2001, Drucker 1999].

2.10 Analyse des liens

Les pages Web sont reliés par des hyperliens, qui portent des informations importantes. Ces liens fournissent souvent un moyen implicite de transport de l'autorité des pages Web vers des autres. Par conséquent, les pages qui sont pointées par de nombreuses autres pages sont susceptibles de contenir des informations fiables. Ces liens doivent évidemment être utilisés pour évaluer le classement de la page dans les moteurs de recherche.

Au cours de la période 1997-1998, deux algorithmes de recherche basés sur l'analyse de lien hypertexte, ont été introduits qui sont les algorithmes PageRank [Brin 1997] et HITS [Kleinberg 1999]. Ces algorithmes exploitent la structure des liens hypertextes du Web pour classer les pages Web en fonction de leurs niveaux d'autorité.

Getoor et Diehl, dans [Getoor 2005], ont regroupé les tâches de l'analyse des liens en Trois classes, comme suit :

1. Tâches liées aux objets :

   (a) Classement des objets basé sur les liens.
   (b) Classification des objets basé sur les liens.
   (c) Clustering des objets (Détection des Groupes et des communautés).
   (d) Identification des entité (Entité résolution).
2. Tâches liées aux liens
   (a) Prévision d'un Lien.
3. Tâches liées aux graphes
   (a) Détection des sous-graphes.
   (b) Classification des graphes.
   (c) Les modèles génératifs pour les graphiques.

2.10.1 Analyse des réseaux sociaux

Les réseaux sociaux ont été largement étudiés bien avant l'avènement du Web. Entre 1950 et 1980, les sciences sociales ont fait des grands pas dans l'analyse des réseaux sociaux [Chakrabarti 2008]. L'analyse des réseaux sociaux est l'étude des entités sociales (appelées acteurs), et leurs interactions et les relations entre eux. Les interactions et les relations peuvent être décrites comme un réseau ou d'un graphe, où chaque sommet (ou nœud) représente un acteur et chaque lien représente une relation [Xu 2009]. A partir du réseau, nous pouvons étudier les propriétés de sa structure, et le rôle, la position et le prestige de chaque acteur social [Thelwall 2006]. Il est aussi possible de trouver les différents types de sous-graphes, par exemple, les communautés formées par des groupes d'acteurs.

L'analyse des réseaux sociaux est utile pour le Web parce que le Web est essentiellement une société virtuelle, et donc un réseau social virtuel peut être représenté formellement par un graphe orienté avec des poids attribués à ses arrêtes. Les nœuds représentent les documents et les arêtes représentent des citations d'un document à d'autres documents. La plupart des résultats des réseaux sociaux peut être adapté et étendu pour une éventuelle utilisation dans le Web.

Dans le contexte du Web, la notion de *prestige* peut être associée avec le nombre des nœuds entrants (in-degree). Une hypothèse évidente dans le réseau social est que le prestige dépend de l'autorité de citations. En d'autres termes, le prestige a un caractère récursif. Ainsi, le score de prestige d'un nœud est pas simplement égale à son le nombre des nœuds entrants, mais doit être défini de manière récursive en utilisant les scores de prestige des nœuds qui les citent. Ceci est faite en utilisant les notions de l'algèbre linéaire [Markov 2007].

Généralement, la littérature compte (voir [Wasserman 1994]) deux types d'analyses des réseaux sociaux ; Centralité et Prestige, qui sont étroitement liées à l'analyse des liens hypertextes et de la recherche sur le Web. La Centralité et le Prestige sont des mesures de degré d'importance d'un acteur dans un réseau social.

2.10.1.1 Centralité

La Centralité (en anglais : Centrality) donne une indication approximative de la puissance sociale d'un nœud dans le réseau basé sur la façon dont elle influe sur la réseau. des acteurs importants sont ceux qui sont liés ou impliqué avec beaucoup d'autres acteurs [Wasserman 1994, Xu 2010]. . Plusieurs types de centralité sont définies sur les graphes orientés et non orientés.

La figue 2.12 montre un exemple simple utilisant un graphe non orienté. Chaque nœud du réseau social est un acteur et chaque lien indique que les acteurs sur les deux extrémités du lien communiquent entre eux. Intuitivement, nous voyons que l'acteur i est l'acteur le plus central parce qu'il peut communiquer avec la plupart des autres acteurs.

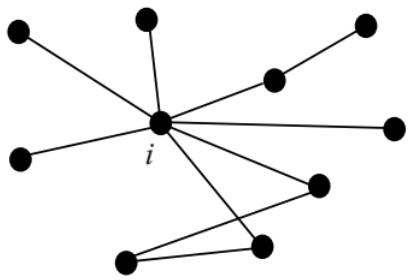

FIGURE 2.12 – Exemple d'un réseau sociale

Les trois types essentiels de centralité comprennent le degré de centralité (degree centrality), centralité de proximité (closeness centrality) et centralité d'intermédiarité (betweenness centrality) [Wasserman 1994, Xu 2010]. .

degré de Centralité Le degré de Centralité est basé sur l'idée que l'importance d'un acteur au sein d'un groupe dépend du nombre total des acteurs qu'il connaît ou avec lesquels il interagit. Selon cette mesure, l'importance d'un acteur dans un réseau social revient donc à calculer le nombre de ses sommets voisins, ou d'une manière équivalente, à calculer le nombre de liens qui lui sont incidents [Wasserman 1994, Xu 2010].

Le degré de centralité d'un acteur i (désigné par $C_D(i)$) est tout simplement le degré du nœud (le nombre d'arêtes) de l'acteur, notée $d(i)$, normalisée avec le degré maximum, $n-1$, comme suit :

$$C_D(i) = \frac{d(i)}{(n-1)} \tag{2.24}$$

Centralité de proximité Cette mesure correspond à l'idée qu'un acteur est important s'il est capable de contacter facilement un grand nombre d'acteurs avec un minimum d'effort (l'effort ici est relatif à la taille des chemins). En pratique, la centralité de proximité d'un acteur est obtenue en calculant sa proximité moyenne par rapport à des autres acteurs du réseau [Wasserman 1994, Xu 2010].

Soit $d(i, j)$ la plus courte distance de l'acteur i à l'acteur j (mesurée par le nombre de liens dans le plus court chemin). La centralité de proximité $C_C(i)$ de l'acteur i est défini :

$$C_C(i) = \frac{(n-1)}{\sum_{j=1}^{n} d(i,j)} \tag{2.25}$$

Centralité d'intermédiarité La centralité d'intermédiarité est une autre mesure de centralité globale. L'intuition de cette mesure est que, dans un graphe, un nœud est d'autant plus important qu'il est nécessaire de le traverser pour aller d'un nœud quelconque à un autre. Plus précisément, un sommet ayant une forte centralité d'intermédiarité est un sommet par lequel passe un grand nombre de chemins plus courts dans le graphe [Chikhi 2010]. Dans un réseau social, un acteur ayant une forte centralité d'intermédiarité est un sommet dont un grand nombre d'interactions entre des sommets non adjacents dépend de lui [Wasserman 1994, Xu 2010].

Dans un réseau de communication, la centralité d'intermédiarité d'un nœud peut être considérée comme la probabilité qu'une information transmise entre deux nœuds passe par ce nœud intermédiaire.

Soit P_{jk} le nombre de chemins les plus courts entre les acteurs j et k. L'intermédiarité d'un acteur i est défini par le nombre des plus courts chemins qui passent i (désigné par $P_{jk}(i)$ $j \neq i$ *et* $k \neq i$) normalisée par le nombre total des chemins les plus courts de toutes les paires d'acteurs non compris $i()$:

$$C_B(i) = \sum_{j<k} \frac{P_{jk}(i)}{P_{jk}} \tag{2.26}$$

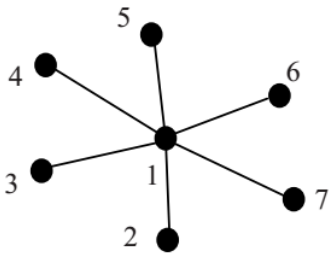

FIGURE 2.13 – Exemple d'un réseau sociale

Dans le réseau de la Figure 2.13, l'acteur 1 est l'acteur le plus central. Il se trouve sur les 15 chemins les plus courts reliant les 6 autres acteurs. $C_B(1)$ a la valeur maximale 1 et $C_B(2) = C_B(3) = C_B(4) = C_B(5) = C_B(6) = C_B(7) = 0$

2.10.1.2 Prestige

Prestige est une mesure d'importance appliquer uniquement sur les graphes orientés, en tenant en compte les différences entre les liens entrants et les liens sortants. Prestige est une mesure plus précise de l'importance. Un acteur prestigieux est celui qui reçoit beaucoup des liens importants. En d'autres termes, pour calculer le prestige d'un acteur, on considère seulement les liens dirigés vers cet acteur (liens entrants). La différence principale entre les concepts de centralité et de prestige est que la centralité se concentre sur les liens sortant alors que prestige se concentre sur les liens entrants [Liu 2007]. Il existe trois mesures de prestige [Wasserman 1994, Xu 2010].

Prestige de degré Un acteur est prestigieux lorsqu'il reçoit beaucoup des liens entrants(ou des nominations). Ainsi, la mesure la plus simple de prestige d'un acteur i (désigné par $P_D(i)$) est son degré entrant (in-degree).

$$P_D(i) = \frac{d_I(i)}{(n-1)} \tag{2.27}$$

où $d_I(i)$ est le in-degré i (le nombre des liens entrants de i) et n est le nombre total d'acteurs du réseau. Comme dans la centralité de degré, en divisant par $n-1$, on normalise la valeur de prestige dans l'intervalle $[0, 1]$. La valeur maximale de prestige est égale à 1 lorsque tous les autres liens d'acteur pointent vers (ou choisissent) l'acteur i.

Prestige de proximité Prestige de proximité considère la proximité de l'acteur i aux d'autres acteurs dans son domaine d'influence, c-à-d l'ensemble de tous les acteurs du réseau qui peuvent atteindre l'acteur i, directement et indirectement.

Soit I_i l'ensemble des acteurs qui peuvent atteindre l'acteur i, qui est appelé aussi le domaine d'influence de l'acteur i. La proximité est définie comme la distance des autres acteurs à i. Soit $d(j, i)$ indique la distance du plus court chemin de l'acteur j à l'acteur i. Pour calculer le prestige de proximité, on utilise la distance moyenne, ce qui est :

$$P_P(i) = \frac{\sum_{j \in I_i} d(j, i)}{|I_i|} \tag{2.28}$$

Rang Prestige Le Rang Prestige constitue la base de la plupart des pages Web des algorithmes d'analyse des liens, y compris PageRank et HITS. L'idée principale du prestige de rang est que le prestige d'un acteur est influencé par les rangs ou les statuts des acteurs concernés. Sur la base de cette intuition, le prestige de rang d'un acteur i ($PR(i)$) est défini comme suit :

$$PR(i) = A_{1i}PR(1) + A_{2i}PR(2) + \cdot + A_{ni}PR(n) \tag{2.29}$$

où, $A_{ji} = 1$ si l'arc j pointe sur i, et 0 sinon. Cette équation confirme que le rang de prestige d'un acteur est une fonction des rangs des acteurs qui votent ou choisissent cet acteur [Wasserman 1994, Xu 2010].

2.10.2 PAGERANK

L'algorithme PageRank a été présenté par Sergey Brin et Larry Page [Brin 1997] à la 7^{ime} Conférence internationale du World Wide Web (WWW'7), en Avril 1998 [Chakrabarti 2008]. Cet algorithme est basé sur le principe de rang Prestige dans l'analyse des réseaux sociaux. Le PageRank est une mesure de la qualité des pages Web utilisées dans le moteur de recherche Google. PageRank utilise les liens hypertextes comme un indicateur de qualité d'une page Web.

Essentiellement, le PageRank interprète un lien hypertexte de la page x à la page y comme un moyen de transport de prestige, de la page x vers la page y. Cependant, le PageRank ne considère pas uniquement le grand nombre de liens qu'une page reçoit mais prend en compte la qualité de la page qui transmet le prestige. Alors, les Hyperliens de pages qui sont elles-mêmes importantes aident à faire d'autres pages plus importantes.

PageRank propose un classement statique des pages Web dans le sens où une valeur de PageRank est calculée pour chaque page hors ligne et il ne dépend pas de requêtes de recherche. Puisque, PageRank est basé sur la mesure de l'autorité dans les réseaux sociaux, la valeur PageRank de chaque page peut être considérée comme son autorité [Liu 2007].

Les principaux concepts de l'algorithme PageRank peuvent être décrits comme suit :

1. Un lien hypertexte d'une page pointant vers une autre page est un moyen de transport implicite de l'autorité à la page cible. Ainsi, plus une page i reçoit des incidents, elle a plus de Prestige.
2. Un lien hypertexte d'une page de haut Prestige est plus important qu'un lien hypertexte d'une page à faible Prestige. En d'autres termes, une page est importante si elle est pointée par d'autres pages importantes.

Selon le rang Prestige dans les réseaux sociaux, l'importance d'une page donnée est déterminée en additionnant les scores de PageRank de toutes les pages qui

pointent vers cette page. Puisque une page peut pointer vers d'autres pages, son score de prestige devrait être partagé entre toutes les pages qu'il pointe.

Pour formuler les idées ci-dessus, le Web peut être traité comme un graphe orienté $G = (V, E)$, où V est l'ensemble des nœuds (les pages), et E est l'ensemble des arêtes dans le graphe (des hyperliens).

Soit le nombre total de pages sur le Web soit n (d'où $n = |V|$). Le score de PageRank de la page i (notée $P(i)$) est défini par :

$$P(i) = (1 - d) + d \sum_{(i,j \in E)} \frac{P(j)}{O_j} \tag{2.30}$$

Où O_j est le nombre de liens sortants de la page j, et d est un facteur d'atténuation.

Le problème avec l'équation ci-dessus (2.30) est que le calcul de $P(i)$ fait intervenir les autres $P(j)$ qui sont a priori eux aussi inconnus.

l'algorithme PageRank est interprété par l'idée du navigateur aléatoire [Langville 2005]. Un navigateur aléatoire représente un utilisateur virtuel qui navigue à travers le graphe des liens en suivant à chaque fois un des hyperliens de la page courante [Borodin 2005]. Dans la version simplifiée du PageRank, le navigateur aléatoire choisit à chaque fois de suivre de manière équiprobable un lien parmi les différents liens sortants de la page sur laquelle il se trouve [Chikhi 2010].

L'ensemble des valeurs PageRank des pages peut être représenté par un vecteur P de dimension n tel que :

$$P = (P(1), P(2), \cdots, P(n)) \tag{2.31}$$

Soit A la matrice d'adjacence du graphe G avec :

$$\begin{cases} \frac{1}{O_j} si \{i, j\} \in E \\ 0 sinon \end{cases} \tag{2.32}$$

En utilisant les équations (2.30 et 2.31), le système de n équations peut être écrit comme suit :

$$P = (1 - d)e + dA^T P \tag{2.33}$$

Où la solution P est un vecteur propre correspondant à la valeur propre de 1.

Puisque la définition de P est récursive (Voir l'équations 2.33) alors il est préférable d'utiliser un algorithme itératif (Algorithme 1) pour résoudre le problème.

Algorithme 1 Solution itérative de l'algorithme PageRank

$PR \leftarrow (r_1, r_2, \cdots, r_N 1)$
$D \leftarrow (\frac{1}{N}, \frac{1}{N}, \cdots, \frac{1}{N})$
$d \leftarrow 0.15$
répéter
  $PR_{i+1} \leftarrow A^T * PR_i$
  $PR_{i+1} \leftarrow (1-d) * PR_{i+1} + d * D$
jusqu'à $||PR_{i+1} - PR_i|| < \varepsilon$
retour PR

L'algorithme peut commencer avec toutes les affectations initiales des valeurs de PageRank. L'itération se termine lorsque les valeurs de PageRank convergent (c-à-d ne changent pas beaucoup). Cet algorithme retourne à la fin le vecteur PR qui représente le classement globale de toutes les pages dans le graphe du Web.

2.10.2.1 Les avantages de PageRank

Le classement des pages Web en utilisant l'algorithme PageRank est motivé par les avantages suivants [Xu 2010] :

— Sa capacité à lutter contre le spam. Comme il n'est pas facile pour le propriétaire des pages d'ajouter liens entrants dans leurs pages à partir d'autres pages importantes, il n'est donc pas facile d'augmenter le score PageRank des pages.

— Le score PageRank est une mesure globale du rang d'une page Web et indépendant de la requête de l'utilisateur. Dans Google, le graphe du Web, qui a été créé à partir des pages analysées, est d'abord utilisé pour calculer les scores PageRank de toutes les pages, et les scores de PageRank calculés seront conservés pour les processus plus tard. Lorsqu'une requête de recherche est soumis, un index de texte est d'abord consulté pour sélectionner les pages de réponses possibles. Ensuite, un système de classement combinant PageRank avec les contenus textuels correspondants est utilisé pour produire une liste de classement final des URL de réponse. Cela rend Google beaucoup plus rapide que les autres moteurs de recherche qui se basent sur le texte conventionnel.

2.10.2.2 Les inconvénients de PageRank

Malgré que l'algorithme PageRank constitue le cœur de moteur de recherche Google, néanmoins cet algorithme présente certains inconvénients, qui se résument comme suit [Xu 2010] :

— Le PageRank ne peut pas distinguer entre les pages qui font autorité en

général, et les pages qui font autorité sur le sujet de la requête, c'est-à-dire le PageRank est indépendant de la requête
— PageRank favorise les pages plus anciennes parce une nouvelle page même si elle a un très bon contenu, elle ne peut pas avoir plusieurs liens entrant (sauf si elle fait partie d'un site ancien).

2.10.3 HITS

J.Kleinberg [Kleinberg 1999] a proposé l'algorithme HITS (Hypertext Induced Topic Search) en 1998 , dans le cadre du projet de recherche CLEVER de IBM Almaden Research Center. HITS est fondé sur l'idée d'exploiter la structure du Web dans l'amélioration de la qualité de la recherche d'information. Cependant, à la différence de PageRank qui assigne à chaque page un seul degré d'importance, l'algorithme HITS caractérise chaque page par deux degrés d'importance. Ces deux degrés, que Kleinberg appelle degrés d'autorité (Authority) et d'hubité (Hub), sont respectivement des mesures de centralité par rapport aux liens entrants et aux liens sortants.

HITS est un algorithme de classement en fonction de requête de recherche. Lorsque l'utilisateur émet une requête de recherche, HITS d'abord élargit la liste des pages pertinentes retournée par un moteur de recherche, puis produit deux classements de cet ensemble, le classement de l'autorité et le classement d'hubité.

Une autorité est une page avec plusieurs liens entrants. L'idée est que la page peut avoir une bonne autorité (importance ou popularité) sur un sujet donné et donc beaucoup de gens faire confiance et envoient des liens vers elle. Un Hub (concentrateur) est une page avec des nombreux liens sortants. La page Hub joue le rôle d'un organisateur de l'information sur un sujet particulier et pointe vers de nombreuses bonnes pages d'autorité sur le sujet. Quand un utilisateur vient vers cette page hub, il trouvera plusieurs liens utiles qui le prennent vers des bonnes pages de ce sujet .

L'idée de base de HITS est que un bon Hub pointe vers beaucoup bonnes autorités et une bonne autorité est pointé par beaucoup des bons Hubs. Ainsi, les autorités et les hubs forment une relation de renforcement mutuel. Cette relation se manifeste habituellement par un ensemble des autorités et hubs densément liés (voir Figure 2.14).

L'algorithme HITS peut être divisé en deux phases :

1. La construction d'un sous-graphe d'extrait du Web,
2. Le calcul des hubs et des autorités.

HITS utilise d'abord les moteurs de recherche pour obtenir l'ensemble des pages racines W c-à-d les pages Web pertinentes avec score élevé de rang. Ensuite, Il construit un graphe réduit du Web qui contient la plupart des autorités et des

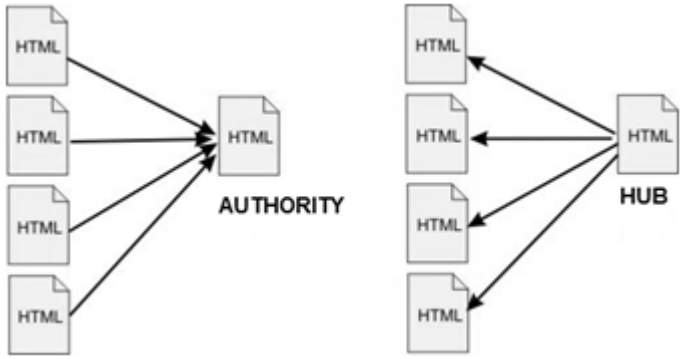

FIGURE 2.14 – Hubs et Autorités

Hubs.

Cet ensemble racine W est étendu en incluant toutes les page pointées par une page en W et toutes les pages qui pointent vers une page en W. Cela donne un ensemble plus large appelé l'ensemble de base S. Ensuite, HITS démarre la deuxième phase ; le calcul du score autorité et score Hub de chaque page dans S.

Soit n le nombre de pages en S. Nous utilisons $G = (V, E)$ pour désigner le graphe des liens hypertextes induit par S. V est l'ensemble des pages (ou nœuds) et E est l'ensemble des arêtes (ou liens). Ensuite, A est utilisé pour désigner la matrice d'adjacence du graphe.

Les poids Hub des pages Web peuvent être décrits comme un vecteur h, où h_i est le score Hub de la page i. Les score autorités peuvent être décrits aussi comme un vecteur a où a_i indique le score autorité de la page i. HITS successivement raffine ces scores en calculant :

$$\begin{aligned} a_i &= \sum_{i \to j} h_j \\ h_j &= \sum_{i \to j} a_i \end{aligned} \tag{2.34}$$

Ces équations (2.34) peuvent être écrites sous forme matricielle à l'aide de la matrice d'adjacence A du graphe du Web réalisé.

$$\begin{aligned} a &= A^t \cdot h \\ h &= A \cdot a \end{aligned} \tag{2.35}$$

Les scores autorité et Hub peuvent être calculés en utilisant la méthode itérative. A partir de $a_0 = h_0 = (1, 1, \cdots, 1)$, si nous utilisons a_k et h_k pour désigner les scores autorité et Hub à la k^{ime} itération , les solutions finales pour le processus d'itération obtenus par des substitutions sont :

$$\begin{aligned} a &= A^T A a_{k-1} \\ h &= A A^T h_{k-1} \end{aligned} \tag{2.36}$$

Cela conduit à un algorithme itératif (Algoithme 2) :

Algorithme 2 Solution itérative de l'algorithme HITS

$h_0 \leftarrow (1, 1, \cdots, 1)$
$a_0 \leftarrow (1, 1, \cdots, 1)$
$k \leftarrow 0$
répéter
  $k \leftarrow k + 1$
  $a_k \leftarrow A^t h_{k-1}$
  $h_k \leftarrow A * a_k$
  Normaliser les deux vecteurs (a_k *et* h_k)
jusqu'à $|a_k - a_{k-1}| < \xi_a$ *et* $|h_k - h_{k-1}| < \xi_h$
retour a_k *et* h_k

2.10.3.1 Les avantages de HITS

Le classement des pages Web en utilisant l'algorithme HITS présente quelques avantages [Langville 2011] :

— HITS produit deux listes classement à l'utilisateur (double classement) : autorité et hub. En tant qu'utilisateur, il est agréable d'avoir cette option. Parfois, vous voulez des pages faisant autorité parce que vous êtes à la recherche en profondeur sur une requête de recherche donnée. D'autres fois, vous voulez des pages hubs parce que vous faites une recherche étendue.

— HITS considère le problème du classement des pages Web comme un petit problème, en trouvant les vecteurs propres dominants de petites matrices. La taille de ces matrices est très faible par rapport au nombre total des pages contenues dans le Web.

2.10.3.2 Les inconvénients de HITS

Malheureusement, les résultats des classements obtenus en se basant sur l'algorithme HITS peut être influencés négativement à cause de quelques inconvénients, qui sont [Langville 2011] :

— L'algorithme HITS n'a pas la capacité anti-spam de PageRank. Il est assez facile d'influencer HITS en ajoutant des liens sortants de propre page pour pointer vers de bonnes autorités. Cela augmente le score Hub de la page. Parce que les scores Hub et autorité sont interdépendants, à son tour augmente également l'autorité score de la page.

— HITS est dépendant du sujet. En élargissant l'ensemble des pages résultant de la requête , il peut facilement recueillir de nombreuses pages (y compris les pages autorité et pages hub) qui n'a rien à voir au sujet de recherche,

car liens sortants d'une page ne peut pas pointer vers les pages qui sont pertinentes pour le sujet et liens entrants aux pages dans l'ensemble peut être sans importance aussi bien parce que les gens mettent des hyperliens pour n'importe quelle raison, y compris le spamming.

— L'évaluation du temps de recherche est également un inconvénient majeur. Au moment de la requête, un graphe de voisinage doit être construit et le problème de vecteur propre au moins d'une matrice doit être résolu. Et, cela doit être effectuée pour chaque requête.

2.11 Conclusion

Dans ce chapitre, nous avons abordé le domaine de l'exploration du Web y compris sa taxonomie. Nous avons aussi exposé les deux domaines les plus importants dans l'analyse de Web qui sont la classification des pages Web et la recherche des informations dans le Web. Nous avons clôturé ce chapitre par une étude concernant l'analyse liens puisqu' elle est un module très intéressant dans la recherche d'information et nous l'avons considéré comme la base de notre approche qui va être détaillée dans les chapitres suivants.

La caractéristique intrinsèque des données Web est le volume important. Donc, la plupart des tâches de l'analyse du Web commence par une phase de prétraitement qui parmi ses objectif la réduction de dimension de ses données. Pour cela, nous allons consacré le chapitre suivant à l'exploration du domaine de la réduction des dimensions des données, en précisant les approches et les techniques qui leur sont liées.

Chapitre 3

Réduction de dimension

Sommaire

3.1 Introduction

Au cours des dernières années, la dimension des données, utilisées comme des entrées de techniques d'apprentissage automatique et l'analyse de données, a augmenté d'une manière explosive. L'analyse de ce type données (de grand échelle) présente

des défis sérieux aux techniques d'apprentissage existantes [Tang 2014]. Dans la réalité, les caractéristiques pertinentes sont souvent inconnues à priori. Par conséquent, de nombreuses caractéristiques candidates sont introduites pour mieux représenter un domaine quelconque. Malheureusement, une grande partie de ces caractéristiques sont partiellement ou complètement inutiles ou redondantes. Une caractéristique non pertinente n'a aucun effet sur la description du concept cible, et une caractéristique redondante n'ajoute rien de nouveau à la notion de cible [John 1994].

En plus, la taille énorme des données peut ralentir le processus d'analyse, surtout la phase d'apprentissage, alors il faut supprimer les caractéristiques indésirables. La réduction du nombre de caractéristiques non pertinentes et redondantes réduit considérablement le temps de l'apprentissage. Un autre problème qui oblige les analyseurs à s'orienter vers la réduction de dimensions est que plusieurs caractéristiques sont redondantes et bruitées c-à-d peuvent influencer négativement sur la précision des modèles obtenus.

La réduction de dimension consiste à projeter l'ensemble des données de l'espace original sur un nouvel espace de dimensions réduites [Mladenić 2006]. La réduction est habituellement réalisée soit en sélectionnant un sous ensemble des dimensions d'origines ou en construisant des nouvelles dimensions. La résolution du problème de la dimension des données est une branche importante dans le domaine data mining. A cet effet, plusieurs techniques de réduction de dimension ont été proposé et étudié.

Le reste de ce chapitre est organisé comme suit : la section 2 donne un bref aperçu du domaine réduction de dimension. Les approches de sélection des sous-ensembles des caractéristiques couramment utilisés dans l'apprentissage automatique, à savoir ; filtrage, enveloppes et intégrées, sont décrites dans la section 3. La section 4 illustre l'extraction des caractéristiques et ses techniques les plus connus. Ensuite, nous allons discuter quelques travaux, extraits de la littérature, concernant le domaine de la réduction de dimension et la sélection des caractéristiques contenues dans les documents textuels et dans les pages Web. Ce chapitre se termine par une conclusion.

3.2 Réduction de la dimension

La réduction de dimension consiste à projeter l'ensemble des données de l'espace original sur un nouvel espace de dimensions réduites [Mladenić 2006]. La réduction de dimension est une étape couramment utilisée dans l'apprentissage automatique, surtout face à un espace de caractéristiques de dimension très élevé.

Les principales raisons motivant l'utilisation la réduction de dimension dans l'apprentissage sont les suivantes [Mladenić 2006] :

— Améliorer la performance de la prédiction, afin d'améliorer l'efficacité d'ap-

prentissage.
— Fournir des prédicteurs plus rapides éventuellement utilisant moins d'informations sur les données d'origine.
— Réduire la complexité des résultats produits et permettre une meilleure compréhension du processus de classification.

.

La réduction de dimension peut être faite de deux façons différentes : en ne gardant que les caractéristiques (ou les variables) les plus pertinentes de l'ensemble des données d'origine, cette technique est appelée sélection de caractéristiques. Une deuxième approche exploite la redondance des données d'entrée et produit un plus petit ensemble de nouvelles variables, chacune étant une combinaison de variables d'entrées, qui contient essentiellement les mêmes informations que les variables d'entrée, cette technique est appelée extraction des caractéristiques.

3.3 Sélection de caractéristiques

La sélection des caractéristiques est une technique largement utilisée pour la réduction de la dimension. Elle se réfère au processus de la sélection d'un sous-ensemble des caractéristiques pertinentes, afin de construire des modèles d'apprentissage robustes. Cette sélection se fait en fonction de certains critères d'évaluation de la pertinence, ce qui conduit généralement à une meilleure performance d'apprentissage.

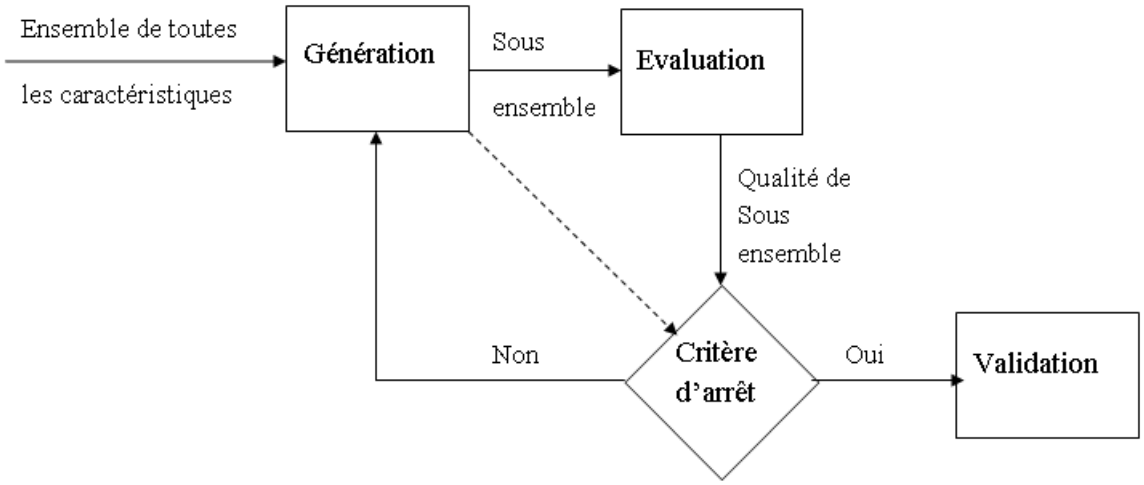

FIGURE 3.1 – Principe de déroulement du processus de sélection des caractéristiques [Nadri 2016]

Selon l'organisation du processus de la sélection des caractéristiques suivi, les algorithmes de sélection des caractéristiques sont généralement classés en trois catégories principales : des méthodes filtres, enveloppes (wrapper) et intégrées [Molina 2002]. Il est possible de combiner les algorithmes des différentes approches [Stańczyk 2015].

Les différents algorithmes de sélection des caractéristiques peuvent être regroupés en deux approches [Aha 1996] :

1. **Sélection ascendante**(Forward Selection) : Dans cette catégorie le processus de sélection commence sans variables et les ajoute une par une, à chaque étape d'ajout l'erreur globale est diminuée. Le processus s'arrête lorsque l'ajout d'une nouvelle variable ne diminue pas significativement l'erreur.
2. **Sélection descendante** : (Backward Selection) La sélection commence en utilisant toutes les variables et la suppression s'effectue une par une, donc à chaque étape on élimine la variable qui diminue l'erreur au maximum, jusqu'à ce qu'une élimination supplémentaire augmente de manière significative l'erreur.

3.3.1 Méthodes de Filtrage

En se basant sur les caractéristiques des données, les méthodes de filtrage évaluent les caractéristiques indépendamment des algorithmes de classification [John 1994]. Elles peuvent être considérées comme une sorte de procédures de prétraitement. Le principe de la méthode filtrage est illustré par la Figure 3.2.

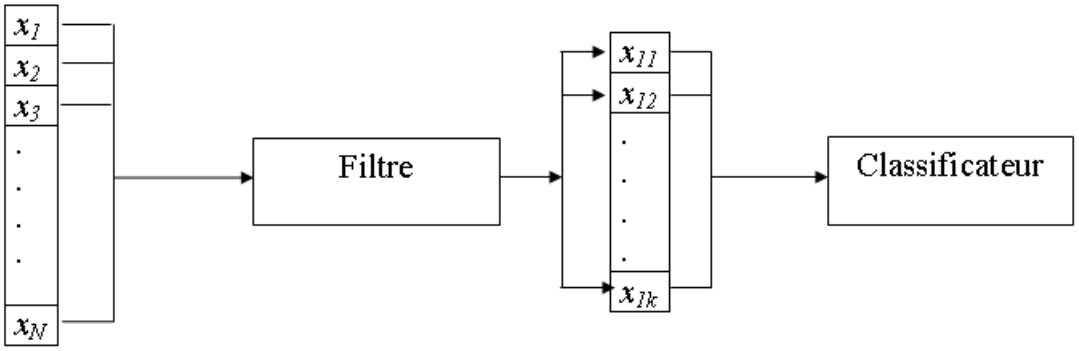

FIGURE 3.2 – Principe de la sélection des caractéristiques de type filtrage [Kohavi 1997]

Un algorithme de filtrage se compose essentiellement de deux étapes consécutives. Dans la première étape, il classe les caractéristiques en se basant sur certains critères. L'évaluation des caractéristiques pourrait être soit univariable ou multivariables. Dans le modèle univariable, chaque caractéristique est classée indépendam-

ment de l'espace des caractéristiques, alors que le système multivariable évalue les caractéristiques d'une manière par lots. Par conséquent, le système multivariable est naturellement capable de gérer des caractéristiques redondantes [Tang 2014]. Dans la deuxième étape, les caractéristiques de classement plus élevé sont choisies pour être utilisées dans les modèles de classification.

Durant la dernière décennie, un certain nombre de critères de performance ont été proposés pour la sélection des caractéristiques basées sur des filtres tels que les scores de Fisher, les méthodes basées sur l'information mutuelle et ReliefF et ses variantes.

3.3.2 Méthodes Enveloppes

La nature générale des filtres les rend applicables dans tous les cas, mais le fait de l'ignorance totale des performances d'un système de classification, en employant seulement l'ensemble des caractéristiques sélectionnées, provoque des résultats généralement pires que d'autres approches [Stańczyk 2015] et elle est considérée comme un désavantage. Pour résoudre cet inconvénient, Kohavi et John [Kohavi 1997] ont introduit le concept "enveloppe" (wrapper) pour la sélection de caractéristiques. Les méthodes enveloppes, évaluent un sous-ensemble de caractéristiques par sa performance de classification en utilisant un algorithme d'apprentissage.

Dans l'approche enveloppe, la sélection du sous-ensemble de caractéristiques est effectuée en utilisant l'algorithme d'induction comme une boîte noire, c-à-d aucune connaissance de l'algorithme est nécessaire, il suffit de l'interface. Le sous-ensemble optimal des caractéristiques doit dépendre à des biais et des heuristiques spécifiques de l'algorithme de classification. Sur la base de cette hypothèse, les méthodes wrapper utilisent un classificateur spécifique pour évaluer la qualité des caractéristiques sélectionnées, et d'offrir un moyen simple et puissant pour résoudre le problème de la sélection des caractéristiques, quel que soit l'algorithme d'apprentissage choisi. Un modèle wrapper typique (voir la Figure 3.3) effectuera les étapes suivantes [Kohavi 1997] :

- Etape 1 : La recherche d'un sous-ensemble de caractéristiques.
- Etape 2 : L'évaluation du sous-ensemble sélectionné de caractéristiques par les performances du système de classification.
- Etape 3 : Répéter les étapes 1 et 2 jusqu'à ce que la qualité voulue soit atteinte.

Les sous-ensembles de caractéristiques sélectionnées par cette méthode sont bien adaptées à l'algorithme de classification utilisé, mais ils ne sont pas forcément valides si l'on change le classificateur [Chouaib 2011].

Les méthodes Enveloppes sont généralement considérées comme étant meilleures que celles de type filtrage. Elles sont capables de sélectionner des petits sous-

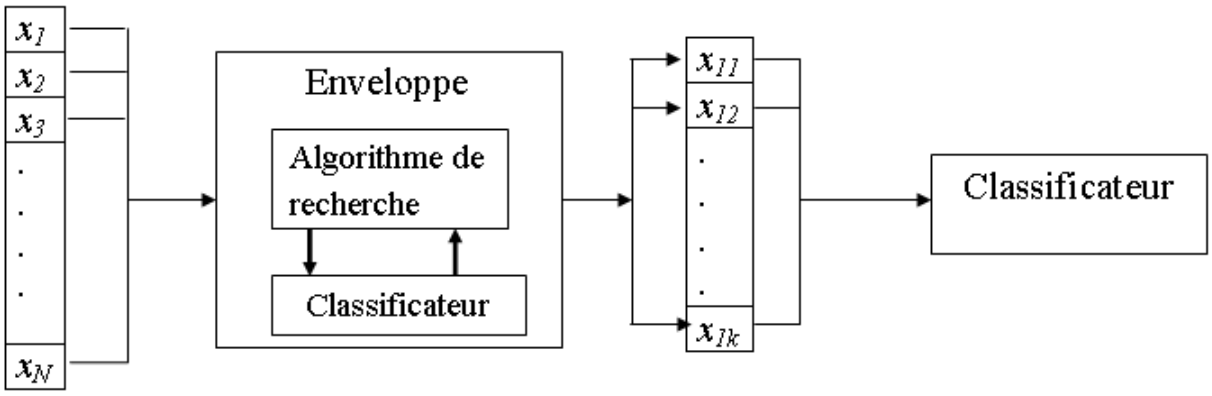

FIGURE 3.3 – Principe de déroulement de la méthode Enveloppe pour la sélection des caractéristiques [Kohavi 1997]

.

ensembles de caractéristiques performantes pour le classificateur utilisé, mais il existe deux inconvénients principaux qui limitent ces méthodes [Chouaib 2011] :

1. Le temps nécessaire pour la sélection des caractéristiques est plus long que celui des approches de filtrage et d'autres approches de sélection de caractéristiques. L'utilisation d'un classificateur pour évaluer les sous-ensembles ainsi que les techniques d'évaluation (comme par exemple la validation croisée) rendent les méthodes "wrapper" très coûteuses en terme de temps de calcul.
2. L'évaluation des caractéristiques se fait par un seul classificateur lors de la sélection. Vu que chaque classificateur a ses spécificités, le sous-ensemble sélectionné dépend toujours du classificateur utilisé.

3.3.3 Méthodes intégrées

Les méthodes intégrées (en anglais :"embedded") intègrent la sélection des caractéristiques dans le processus d'apprentissage et sont généralement spécifiques à des algorithmes d'apprentissage automatique [Guyon 2003]. Dans les méthodes de sélection de type "wrapper", la base d'apprentissage est divisée en deux parties : une base d'apprentissage et une base de validation pour valider le sous-ensemble de caractéristiques sélectionné. En revanche, les méthodes intégrées peuvent se servir de tous les exemples d'apprentissage pour établir le système. Cela constitue un avantage qui peut améliorer les résultats. Un autre avantage de ces méthodes est leur grande rapidité par rapport aux approches "Wrapper" parce qu'elles évitent que le classificateur recommence de zéro pour chaque sous-ensemble de caractéris-

tiques [Chouaib 2011]. Les modèles intégrés ont les avantages de modèles wrapper parce qu'ils comprennent l'interaction avec le modèle de classification et ont aussi les avantages de modèles de filtrages dont ils réalisent beaucoup moins de calculs que les méthodes wrapper [Tang 2014].

Il existe trois types de méthodes intégrées. Les premiers sont les méthodes d'élagage qui utilisent d'abord toutes les caractéristiques pour former un modèle, puis tenter d'éliminer certaines caractéristiques en définissant les coefficients correspondants à 0. Les seconds sont des modèles avec un mécanisme intégré (build-in) pour la sélection des caractéristiques tels que les arbres de décision (C4.5). Les troisièmes sont des modèles de régularisation avec des fonctions objectives qui minimisent les erreurs de montage et en même temps forcent les coefficients d'être petite ou égale à zéro pour les éliminer par la suite [Tang 2014].

3.4 Techniques statistiques de sélection des caractéristiques

Dans cette section, nous allons exposer quelques techniques de sélection des caractéristiques qui ont des fondations statistiques et qui sont utilisées avec succès dans la sélection des caractéristiques contenues dans les textes et les pages Web :

3.4.1 Sélection à base de Fréquence du document (FD)

La fréquence de document (FD) d'une caractéristiques est le nombre de documents (ou des pages Web) dans lequel une caractéristique donnée se produit. Les caractéristiques dont les fréquences de document (FD) sont inférieur à un certain seuil prédéfini sont supprimées. L'hypothèse de base est que les termes rares sont soient non informatifs pour la prédiction de la catégorie, ou n'influent pas sur la performance globale du classificateur. Dans les deux cas une suppression des termes rares réduit la dimension de l'espace de caractéristiques. L'amélioration de la précision de la catégorisation est également possible si les termes rares se trouvent dans l'ensemble des termes de bruit [Yang 1997]. Une caractéristique importante de la méthode fréquence de document est qu'elle ne nécessite pas des étiquettes de classe pour les exemples [Choi 2005].

La FD est la technique la plus simple pour la réduction du la taille de vecteur des caractéristiques. Elle s'adapte facilement avec les corpus volumineux, avec une complexité de calcul approximativement linéaire du nombre de documents de l'apprentissage [Yang 1997].

Cependant, la FD n'est généralement pas utilisée pour l'élimination agressive des termes en raison d'une hypothèse largement reçue dans le domaine de recherche

d'information. Cette dernière propose que les termes de faible FD soient supposés être relativement informatifs et ne doivent pas être retiré de manière agressive [Yang 1997].

3.4.2 Sélection en utilisant le Gain d'Information

En raison de son efficacité de calcul et d'interprétation simple, le Gain d'information (en anglais : Information Gain) (ou de l'entropie) est l'une des méthodes les plus populaires de sélection de caractéristiques des données textuelles. Il est utilisé pour mesurer la dépendance entre une caractéristique et une classe, en calculant le Gain d'informations entre la caractéristiques t_i et la classe C comme suit :

$$\begin{aligned} IG(t_i, C) = -\sum_{i=1}^{k} P_i.log(P_i) + F(t).\sum_{i=1}^{k} p_i(t).log(p_i(t)) \\ +(1 - F(t)).(1 - p_i(t)).(1 - log(p_i(t))) \end{aligned} \tag{3.1}$$

Dans le Gain d'information, une caractéristique est pertinente si elle a un gain de l'information élevé. Pour chaque terme unique, on doit calculer le Gain d'information et supprimer les termes ayant le Gain d'information inférieure à un seuil prédéterminé.

3.4.3 Sélection en utilisant l'Information Mutuelle

La mesure d'information mutuelle(IM) (en anglais : Mutuel Information) est dérivée de la théorie d'information et fournit une manière formelle de modéliser l'information mutuelle entre les caractéristiques et les classes. L'information mutuelle ponctuelle $M_i(t)$ entre le mot t et la classe i est définie sur la base du niveau de co-occurrence entre la classe i et le mot t. Notons que la co-occurrence attendue de la classe i et du mot t sur la base de l'indépendance réciproque est donnée par $P_i \Delta F(w)$. La co-occurrence vraie est bien sûr donnée par $p_i(t).F(t)$. En pratique, la valeur de $F(t)\Delta p_i(t)$ peut être beaucoup plus grande ou plus petite que $P_i.F(t)$, selon le niveau de corrélation entre la classe i et le mot t. L'information mutuelle est définie comme le rapport entre ces deux valeurs, comme suit :

$$\begin{aligned} M_i(t) &= log(\frac{p_i(t).F(t)}{P_i.F(t)}) \\ &= log(\frac{p_i(t)}{P_i}) \end{aligned} \tag{3.2}$$

Il est clair que le mot t est positivement corrélé à la classe i, lorsque $M_i(t) > 0$ et le mot t est corrélée négativement à la classe i, lorsque $M_i(t) < 0$.

On note que $M_i(t)$ est spécifique à une classe donnée i. Donc, il faut calculer l'information mutuelle globale en fonction de l'information mutuelle du mot t avec les différentes classes. Elles sont définies en utilisant les valeurs moyennes et maximales de $M_i(t)$ sur les différentes classes.

$$M_{avg} = \sum_{i=1}^{k} P_i.M_i(t) \tag{3.3}$$

$$M_{avg} = max_{i=1}\{M_i\} \tag{3.4}$$

3.4.4 Sélection par la méthode Relief

Relief est une méthode statistique de sélection des caractéristiques, proposée en 1992 par Kira et Rendell dans [Kira 1992]. Il s'agit d'un algorithme basé sur le poids des caractéristiques inspiré des algorithmes d'apprentissage basés sur les instances. À partir d'un ensemble d'apprentissage, il choisit d'abord un échantillon des instances. L'utilisateur doit fournir le nombre d'instances dans cet ensemble. Relief choisit aléatoirement un sous ensemble des exemples, et pour chaque exemple, il trouve les occurrences H et M basées sur une mesure de distance euclidienne. Relief cherche deux voisins les plus proches : l'un de la même classe, appelé H, et l'autre d'une classe différente, appelé M. Il met à jour les poids des caractéristiques qui sont initialisées à zéro au début sur la base d'une idée intuitive qu'une caractéristique est plus pertinente si elle distingue entre une instance et son M, et moins pertinente si elle distingue entre une instance et son H. Cette étape doit être répétée sur toutes les instances de l'échantillon. A la fin, il choisit toutes les caractéristiques ayant un poids supérieur ou égal à un seuil donné [Dash 1997]. La méthode Relief est résumé dans l'algorithme 3 .

Algorithme 3 Algorithme de la méthode Relief

$T \leftarrow \phi$
Initialiser les poids w_i *à* 0
pour $i \leftarrow 1$ *à* N **faire**
  Choisir alatoirement un point x *de la population*
  pour $i \leftarrow 1$ *à* N **faire**
    $W_j = W_j - dif(x_j, M_j)^2 + dif(x_j, H_j)^2$
pour $i \leftarrow 1$ *à* N **faire**
  si $W_j \geq Th$ **alors**
    Insrer la caractristique t_i *dans la liste* T
retour T

L'algorithme Relief originale peut traiter des caractéristiques ayant des valeurs discrètes et continues, mais il est limité à des problèmes de deux classes. Une extension, ReliefF [Kononenko 1994] traite non seulement des problèmes multiclasses,

mais elle est aussi plus robuste et capable de traiter des données incomplètes et bruyantes. ReliefF a ensuite été adapté pour le problème de la régression. L'ensemble de méthodes Relief sont spécialement intéressantes, car elles peuvent être appliquées dans toutes les situations, ont un faible biais, y compris l'interaction entre les caractéristiques et peut capturer des dépendances locales que d'autres méthodes le manquent.

3.4.5 Sélection par la statistique χ^2

La statistique Chi-square (χ^2) est utilisée pour mesurer l'absence de l'indépendance entre une caractéristique donnée et une catégorie (ou une classe) par rapport à la distribution avec un degré de liberté. χ^2 est le test statistique commun qui mesure la divergence par rapport à la distribution attendue si l'on suppose que l'occurrence d'une caractéristique est indépendant de la classe [Ladha 2011].

Soit n le nombre total de documents de la collection, $p_i(t)$ la probabilité conditionnelle de classe i pour les documents qui contiennent t, P_i soit la fraction globale des documents contenant la classe i, et $F(t)$ la fraction globale des documents contenant le mot t. La valeur χ^2 statistique entre le mot t et la classe i est défini comme suit :

$$\chi_i^2(t) = \frac{n.F(t)^2.(p_i(t) - Pi)^2}{F(t).(1 - F(t)).P_i(1 - P^i)} \tag{3.5}$$

Il est possible de calculer la statistique globale χ^2 à partir des valeurs spécifiques de la classe. Les valeurs moyennes ou maximales peut être utilisées pour créer la valeur composite [Aggarwal 2014] :

$$\chi_{avg}^2(t) = \sum_{i=1}^{k} Pi.\chi_i^2(t) \tag{3.6}$$

$$\chi_{max}^2(t) = max_i \chi_i^2(t) \tag{3.7}$$

Une différence majeure entre χ^2 et l'Information Mutuelle (§ 3.4.3) est que χ^2 est une valeur normalisée, et donc les valeurs de χ^2 sont comparables entre les termes pour la même catégorie. La statistique χ^2 n'est pas fiable pour les termes de petite fréquence.

3.4.6 Sélection en utilisant de l'Indice de Gini

L'Indice de Gini (en anglais : Gini Index) est une méthode d'impureté de séparation, qui a été proposé par Breiman en 1984. Il a été largement utilisé dans de nombreux types des arbres décisionnels pour sélectionner l'attribut de fractionnement,

et il a obtenu de très bonne précision de catégorisation [Shang 2007, Zhu 2015]. Il est couramment utilisé pour mesurer la puissance de discrimination d'une caractéristique particulière [Aggarwal 2015]. Typiquement, il est utilisé pour les variables catégorielles, mais il peut être généralisé aux attributs numériques.

L'idée principale de la théorie indice de Gini est la suivante : Supposons que S est un ensemble de s échantillons, et que ces échantillons sont de k différentes classes $C_i(i = 1, \cdots, k)$. Selon les différences de classes, on peut diviser S en k sous-ensembles $S_i(i = 1, \cdots, k)$. Supposons que S_i soit un ensemble d'échantillons appartenant à la classe C_i, et que s_i soit le nombre des échantillons dans l'ensemble S_i ; Alors, l'indice de Gini de l'ensemble S est :

$$G(S) = 1 - \sum_{j=1}^{k} p_j^2 \tag{3.8}$$

Où p_j est la probabilité que tout échantillon appartienne à C_i.

Le minimum de $Gini(S)$ est 0, c'est-à-dire que tous les membres de l'ensemble appartiennent à la même classe ; Cela signifie qu'il peut obtenir le maximum d'informations utiles. Si tous les échantillons dans l'ensemble sont distribués équitablement sur différentes classes, alors le $Gini(S)$ est maximale ; Cela signifie qu'il peut obtenir le minimum des informations utiles.

L'idée principale de l'indice de Gini est : pour chaque attribut, après qu'il traverse tous les chemin de segmentation possibles, si elle peut fournir l'indice de Gini *minimal* alors il est choisi comme un critère de division.

L'inconvénient de cette approche est que la distribution globale des classes peut être biaisée et, par conséquent, la mesure ci-dessus peut parfois ne pas refléter avec précision le pouvoir de discrimination des attributs sous-jacents. Par conséquent, il est possible de construire un indice de Gini normalisé afin de refléter plus précisément le pouvoir de discrimination des attributs.

Soit $P_1 \cdots P_k$ les distributions globales des documents dans les différentes classes. La valeur de probabilité normalisée $p'_i(w)$ est déterminée comme suit :

$$p'_i(w) = \frac{p'_i(w)/P_i}{\sum_{j=1}^{k} p'_j(w)/P_j} \tag{3.9}$$

Ensuite, l'indice de Gini est calculé en fonction de ces valeurs de probabilité normalisées.

$$G(S) = 1 - \sum_{j=1}^{k} p'^2_j \tag{3.10}$$

L'utilisation des probabilités globales P_i assure que l'indice de Gini reflète plus précisément la discrimination de classe dans le cas de la distribution des classes biaisés dans toute la collection de documents.

Wenqian Shang et al [Shang 2007] ont introduit un **nouvel** algorithme de indice Gini. Cette fonction a été utilisée pour sélectionner des caractéristiques de l'espace d'origine. La forme originale de l'algorithme de l'indice de Gini a été utilisée pour mesurer l'impureté des attributs par rapport à la classification. Plus l'impureté est faible, l'attribut est meilleur [Singh 2010b, Zhu 2015] :

$$G(W) = P(W)(1 - \sum_{i=1}^{m} P(C_i|W)^2) + P(\overline{W})(1 - \sum_{i=1}^{m} P(C_i|\overline{W})^2) \tag{3.11}$$

Un amélioration de la formule (Eq. 3.9) a été proposée par [Shang 2007], comme suit :

$$G(W) = (\sum_{i=1}^{m} P(C_i|W)^2)(\sum_{i=1}^{m} P(W|C_i)^2) \tag{3.12}$$

3.4.7 Sélection à base de score de Fisher

Le score de Fisher (F-score) est l'une des techniques de sélection de caractéristiques supervisée les plus utilisées. Les caractéristiques seront évaluées indépendamment selon son F-score. Les caractéristiques de haute qualité devraient attribuer des valeurs similaires aux instances de la même classe et des valeurs différentes aux instances de classes différentes. Avec cette intuition, le F-score S_i de la i^{me} caractéristique sera calculé comme suit :

$$S_i = \frac{\sum_{k=1}^{K} n_j(\mu_{ij} - \mu_i)^2}{\sum_{k=1}^{K} n_j \rho_{ij}^2} \tag{3.13}$$

Où μ_{ij} et ρ_{ij} sont la moyenne et la variance de la i_{me} caractéristiques dans la j_{me} classe respectivement n_j est le nombre d'instances dans la j_{me} classe et μ_i est la moyenne de la i_{me} caractéristique.

Cependant, la méthode F-score sélectionne chaque caractéristique indépendamment selon leur résultats dans le cadre du critère de Fisher, ce qui conduit à un sous-ensemble de caractéristiques non optimale. Les auteurs de [Gu 2012] ont présenté un modèle F-score généralisé pour sélectionner des caractéristiques conjointement. leur but était trouver un sous-ensemble de caractéristiques qui maximisent la limite inférieure de F-score classique.

3.4.8 Sélection des caractéristiques basée sur la Corrélation

La sélection des caractéristiques basée sur la Corrélation (Correlation-Based Feature Selection,(CFS)) [Hall 1997, Hall 1999] est un algorithme de filtrage simple qui classe sous-ensembles des caractéristiques selon une fonction d'évaluation heuristique basée sur la corrélation. Le biais de la fonction d'évaluation est tend vers des sous-ensembles qui contiennent des caractéristiques qui sont fortement corrélées avec la classe et décorrélés avec les autres classes. Les caractéristiques non pertinentes doivent être ignorées, car ils ont une faible corrélation avec la classe. Les caractéristiques redondantes doivent être éliminées car elles seront fortement corrélés avec une ou plusieurs des caractéristiques restantes. L'acceptation d'une caractéristique dépendra de la mesure dans laquelle il prévoit des classes dans les zones de l'espace d'exemples qui ne sont pas déjà prédit par d'autres caractéristiques.

La fonction d'évaluation de sous-ensemble de caractéristiques de CFS est :

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{3.14}$$

Où M_S est le "mérite" heuristique d'un sous-ensemble S contenant k caractéristiques, r_{cf} est la corrélation moyenne caractéristique-classe (f_S) et r_{ff} est l'inter-corrélation moyenne caractéristique-caractéristique.

3.4.9 Sélection par la méthode Lasso

La méthode Lasso (Least Absolute Shrinkage and Selection Operator) est une méthode pour la sélection de caractéristiques ou la réduction de dimension développée par Robert Tibshirani dans [Tibshirani 1996]. La régression Lasso est largement utilisée dans les domaines avec des ensembles de données volumineuses, tels que la génomique, où le nombre p de variables (des caractéristiques) peut être du même ordre ou largement supérieur au nombre des exemples n ($p > n$) [Djoukoué 2014]. Elle assure une meilleure précision de la prédiction par le réduction comme la régression par arête, mais en même temps, on obtient une solution clairsemée (dispersées), c-à-d plusieurs coefficients sont exactement 0. Par conséquent, Lasso peut être utilisée pour le retrait et la sélection de variables simultanément [Kim 2004].

Cette méthode d'estimation est définie comme étant un minimiseur du critère des moindres carrés pénalisés par la norme $\ell 1$ du vecteur β. L'estimateur de lasso $\hat{\beta_{Lasso}}$ est la solution du problème d'optimisation suivant :

$$\hat{\beta}_{Lasso} = arg\ min_{\beta \in R^p} = \delta_n(\beta) + \lambda \sum_{j=1}^{p} |\beta_j| \tag{3.15}$$

Où $\delta_n(\beta)$ est la fonction de vraisemblance de log négative et $\lambda > 0$ le paramètre de régularisation. Le second terme est appelé «pénalité $\ell 1$ car il repose sur la norme $\ell 1$ usuelle de β qui offre des propriétés de sélection ainsi que la "sparsité".

Le lasso présente quelques avantages [Djoukoué 2014] :

— Lasso sélectionne automatiquement les variables : certains coefficients sont estimés égale exactement à zéro pour des valeurs suffisamment élevées de λ. Ceux-ci représentent des variables qui n'ont aucun pouvoir discriminatoire.

— Le problème d'optimisation à résoudre est convexe, à savoir relativement facile à résoudre, même s'il n'y a pas de forme analytique de la solution dans le cas général. Il existe plusieurs algorithmes qui résolvent l'équation 3.15.

— Lasso est important pour sa stabilité.

3.4.10 La sélection des caractéristiques en utilisant de l'écart de Poisson

Dans les études de recherche des informations (RI), la distribution de Poisson a été utilisée avec succès dans la sélection des mots d'interrogation efficaces, ce qui motive les auteurs de [Ogura 2009] à adopter la distribution pour la sélection des caractéristiques dans les tâches de classification de texte.

La valeur du modèle de Poisson dans les études de RI est que le degré d'écart par rapport à Poisson peut être utilisé comme une mesure pour déterminer l'ensemble des mots clés efficaces. Cela signifie qu'un bon mot clé, qui sélectionne un ensemble de documents très spécifique, est loin de Poisson, alors qu'un mauvais mot clé se comporte comme ce qui serait attendu par Poisson.

L'écart significatif des bons mots clés de Poisson s'est expliqué par des variables cachées telles que le sujet, l'auteur, le genre, le style, etc. Plus un mot clé s'écarte de Poisson, plus la dépendance à des variables cachées compromet l'hypothèse d'indépendance derrière Poisson. Dans les études IR, le degré d'écart par rapport à Poisson pour un mot donné est souvent mesuré par la fréquence résiduelle des documents inverses (RIDF).

Si un bon mot-clé, sélectionné avec une mesure d'écart par rapport à Poisson, est presque toujours contenu dans les documents appartenant à une certaine catégorie C_j, le mot clé sélectionné doit être utile pour déterminer si un document donné appartient à C_j ou non. Les termes (caractéristiques) idéaux caractérisant une certaine catégorie C_j ont les propriétés suivantes :

— Les distributions réelles des termes de caractéristiques idéales sont largement déviées de Poisson pour un ensemble de documents appartenant à C_j parce que la dépendance des variables cachées est forte dans ce cas.

— D'autre part, les écarts de Poisson ne sont pas significatifs pour les documents n'appartenant pas à C_j. C'est parce que l'utilisation des termes caractéristiques caractérisant C_j est presque purement régie par la «chance» pour les documents n'appartenant pas à C_j.

3.5 Extraction de caractéristiques

3.5.1 Principe d'extraction des caractéristiques

L'extraction de caractéristiques projette l'espace original dans un nouvel espace de caractéristiques en le combinant pour obtenir des dimensions inférieures. Il est difficile de relier les caractéristiques de l'espace d'origine avec des nouvelles caractéristiques. Le processus d'extraction des caractéristiques est basé sur une transformation de l'ensemble original de caractéristiques réelles par une combinaison linéaire de ceux-ci, par lequel la discrimination entre les classes est concentrée sur un nombre réduit de caractéristiques extraites [Diamantini 2015].

Le problème est qu'il n'y a pas de signification physique pour les caractéristiques obtenues par des techniques d'extraction de caractéristiques donc l'utilisation de cette famille de méthodes n'est applicable que dans le cas où la sémantique n'intervient plus dans les étapes qui suivent la réduction [Chouaib 2011].

3.5.2 Travaux fondamentaux d'extraction des caractéristiques

La littérature compte plusieurs techniques d'extraction de caractéristiques. Dans cette section, nous exposons les plus connues comme : Analyse en Composantes Principales (ACP), Analyse linéaire discriminante, Analyse Canonique des Corrélations, Positionnement Multi-Dimensionnel, et enfin la méthode Isomap/

3.5.2.1 Analyse en Composantes Principales (ACP)

L'Analyse en Composantes Principales (ACP) consiste à rechercher la direction suivant laquelle le nuage de points des observations s'étire au maximum. A cette direction correspond la première composante principale. La seconde composante principale est déterminée de telle sorte qu'elle soit la plus indépendante possible de la première ; elle est donc perpendiculaire à celle-ci. Ces deux composantes forment le premier plan principal. Cette opération est réitérée jusqu'à trouver toutes les composantes principales expliquant le maximum de variance [Fodor 2002].

L'ACP peut être décrit comme un algorithme "non supervisé", car elle "ignore" les étiquettes de classe, son objectif est de trouver les directions (les composantes principales) qui maximisent la variance dans un ensemble de données. Les composantes principales sont de nouvelles variables indépendantes, combinaisons linéaires des variables initiales, possédant une variance maximale.

Supposons que nous avons un ensemble de données $X = x_1, x_2, \cdots, x_M$ composé de M observations où chaque observation $x_i = (x_{i1}, x_{i2}, \cdots, x_{iN})$ est composée de

N caractéristiques. X est associé à une matrice de données A de taille $N \times M$ où chaque colonne représente une caractéristique. En pratique, l'ACP peut être résumée par l'algorithme suivant [Chouaib 2011] :

1. Calculer le vecteur $\mu = (\mu_1, \mu_2, \cdots, \mu_m)^T$ qui représente le vecteur moyen où μ_i est la moyenne de la i^{me} composante des données.
2. Calculer la matrice X en soustrayant le vecteur moyen de toutes les colonnes de A afin d'obtenir des données centrées.
3. Calculer la matrice S (de taille $N \times N$) de covariance de X avec ($S = X.X^T$).
4. Décomposer cette matrice en vecteurs propres p_i, valeurs propres λ_i.
5. Calculer la matrice U (de taille $N \times N$) qui est composée des coordonnés des vecteurs propres $\overrightarrow{u_j}$ de S triés dans l'ordre décroissant des modules des valeurs propres λ_j (la première colonne de U est le vecteur propre qui correspond à la plus grande valeur propre)
6. Garder les R premières colonnes de U pour former la matrice $N \times R$ qui représente les R premières composantes principales.

La principale limitation de l'ACP est qu'il ne considère pas la séparabilité des classes, car il ne tient pas en compte l'étiquette de classe du vecteur de caractéristiques [Jolliffe 2002] :

1. ACP effectue simplement une rotation de coordonnées qui aligne les axes transformés avec les directions de la variance maximale.
2. Il n'y a aucune garantie que les directions de la variance maximale contiendra de bonnes caractéristiques de discrimination.

Un autre inconvénient concerne la linéarité de l'ACP puisqu'elle ne considère que les dépendances linéaires entre les variables et ne peut pas fournir une projection pertinente pour une distribution non-linéaire de la population des points.

3.5.2.2 Analyse linéaire discriminante

L'analyse linéaire discriminante, appelée aussi analyse discriminante linéaire de Fisher, est une méthode de réduction de dimensions proposée par Fisher en 1936 [Fisher 1936]. Cette méthode part de la connaissance de la partition d'une population des individus en classes et cherche les combinaisons linéaires des variables décrivant les individus qui conduisent à la meilleure discrimination entre les classes, c-à-d celles qui maximisent l'homogénéité de chaque classe. En d'autres termes, cette méthode consiste à chercher un espace vectoriel de faible dimension qui maximise la variance interclasse [Gu 2011].

Une analyse linéaire discriminante peut être résumée en cinq étapes générales suivantes :

1. Calculer les vecteurs moyens de d dimensions pour les différentes classes de la population.

2. Calculer les matrices de dispersion (inter-deux-classes et intra-classe).

3. Calculer les vecteurs propres $(\mu_1, \mu_2, \cdots, \mu_d)$ et valeurs propres correspondantes $(\lambda_1, \lambda_2, \cdots, \lambda_d)$ pour les matrices de dispersion.

4. Trier les vecteurs propres selon l'ordre décroissant de valeurs propres et choisir ensuite les k plus grands vecteurs propres pour former une matrice W de dimension $k \times d$ (où chaque colonne représente un vecteur propre).

5. Utilisez cette matrice (des vecteurs propres) pour transformer les échantillons dans un nouveau sous-espace. Cela peut se résumer par la multiplication de la matrice : $Y = X \times W$ (où X est une matrice de dimensions $n \times d$ représentant les n échantillons, et Y sont les échantillons transformés dans le nouvel sous-espace).

3.5.2.3 Analyse Canonique des Corrélations

L'Analyse Canonique des Corrélations (Canonical Correlations Analysis (CCA)) a été développé par [Hotelling 1936] pour découvrir les associations linéaires entre deux ensembles de données multidimensionnelles. La CCA est analogue à l'Analyse en Composantes Principales (ACP), mais au lieu d'analyser un seul ensemble de données (sous forme d'une matrice). L'objectif du CCA est d'analyser la relation entre une paire d'ensembles de données c-à-d la relation entre deux matrices de données.

D'un point de vue statistique, l'ACP extrait les directions de covariance maximales entre les éléments dans une seule matrice, alors que la CCA trouve la direction de corrélation maximale entre une paire de matrices. D'un point algébrique de vision linéaire, CCA mesure les similitudes entre deux sous-espaces (ceux engendré par les colonnes de chacune des deux matrices analysées). géométriquement, CCA calcule le cosinus des principaux angles entre les deux sous-espaces.

Pour les vecteurs aléatoires de grande dimension donnée $x \in R^P$ et $y \in R^Q$. Soit x_i et $y_i, i = 1, \cdots, N$, sont N réalisations indépendantes de x et y, respectivement. Comme l'analyse de données multi-variables, CCA extrait des vecteurs canoniques u et v tel que $u^T x$ et $y \times v^T$ possèdent un coefficient de corrélation maximale. Ces paires de vecteurs révèlent différentes associations linéaires qui sont encapsulés dans les x et y.

CCA a été appliquée avec succès dans de nombreuses applications d'apprentissage automatique, par exemple la réduction de la dimension, le clustering, l'apprentissage de plongements de mots, la classification des sentiments, l'apprentissage discriminant, reconnaissance d'objets, l'analyse des données fonctionnelle et la bioinformatique [Chang 2013].

3.5.2.4 Positionnement Multi-Dimensionnel

La méthode de positionnement multidimensionnel (en anglais : MultiDimensional Scaling, MDS) [Messick 1954] permet de construire une représentation des points de l'espace dans une dimension réduite. Son objectif est de construire, à partir d'une matrice de distances calculées sur chaque paire de points, une représentation euclidienne des individus dans un espace de dimension réduite qui préserve au maximum ces distances. Cette méthode est utilisée souvent dans le domaine de la visualisation d'information pour explorer les similarités et les dissimilarités dans les données.

Un algorithme MDS commence avec une matrice de similarité entre individus, puis affecte une position à chaque individu dans un espace de N dimensions, où N est prédéfini. Pour N suffisamment petit, les positions peuvent être représentées à l'aide d'un graphe ou en $3D$.

Soit l'ensemble de données $X = x_1, x_2, \cdots, x_M$ composé de M observations où chaque observation $x_i = x_{i1}, x_{i2}, \cdots, x_{iN}$ est composée de N caractéristiques. Soit d une matrice symétrique de taille $M \times M$ où chaque élément d_{ij} représente la distance entre x_i et x_j. L'idée de MDS est de trouver une configuration de points $y_i, i = 1 \cdots M$ dans un espace de dimension plus réduite qui conserverait les distances entre les points initiaux x_i. Autrement dit, il cherche les points y_i dans un espace de dimension $q < N$ tels que $d(y_i, y_j) \approx d_{ij} = d(x_i, x_j)$.Donc, le problème sera de minimiser l'erreur quadratique suivante :

$$E = (d_{ij} - d(y_i, y_j))^2 \tag{3.16}$$

3.5.2.5 Cartographie des Caractéristiques Isométriques

La méthode de Cartographie des caractéristiques isométrique (Isomap : Isometric Feature Mapping) a été introduite par Tenenbaum et al dans [Tenenbaum 2000] et s'inspire de la méthode Positionnement Multidimensionnel (MDS) pour la connaissance d'une matrice de dissimilarité entre les paires des individus mais en lui donnant comme métrique la distance géodésique (ou curviligne) [Khoder 2013].

Dans ce cas, le but est de trouver une variété (*non linéaire*) contenant les données. La méthode Isomap estime la distance géodésique (voir la Figure 3.4) de la façon suivante : Dans un premier temps, le voisinage de chacun des points est calculé. Une fois les voisinages connus, un graphe est construit en reliant tous les points voisins. Chaque arête du graphe est ensuite pondéré par la distance euclidienne entre les deux extrémités de l'arête. Enfin, la distance géodésique entre deux points est estimée par la somme des longueurs des arêtes le long du plus court chemin entre ces points. En pratique, le plus court chemin entre deux sommets du graphe est calculé en utilisant l'algorithme de Dijkstra [Dijkstra 1971].

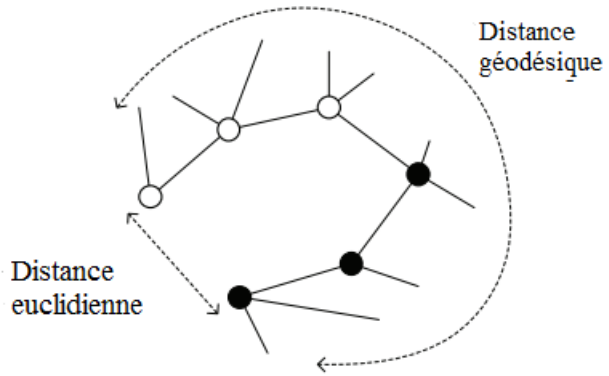

FIGURE 3.4 – La distance géodésique entre deux sommets d'un graphe est le nombre d'arêtes d'un chemin le plus court les reliant.

D'un point de vue algorithmique, l'Isomap projette une surface d'un espace à m dimensions, représentée par n points, de la façon suivante [Khoder 2013] :

1. Recherche des vecteurs de l'ensemble $X = \{x_1, \cdots x_n\} \in R^m$ voisins. D'après les distances euclidiennes d_{ij}, on détermine un voisinage pour chaque point x_i, soit avec le critère des k-plus proches voisins soit en considérant tous les points à l'intérieur d'une sphère de rayon α centré sur x_i. On considère que les distances euclidiennes proches géodésiquement quand les vecteurs se trouvent à petite distance.
2. On fait une estimation des distances géodésiques $d(G_{ij})$ entre tous les points x_i. Isomap construit un graphe dont les sommets sont les points et les arêtes sont les distances entre eux.
3. On applique l'algorithme de Dijkstra afin d'obtenir la matrice des distances D ; elle contient les distances géodésiques entre chaque point et ses voisins. Un sommet est adjacent à un autre seulement s'ils ont été définis comme voisins (Étape 1). La distance géodésique est estimée entre chaque paire de données par la distance la plus courte parcourue sur le graphe (Voir Figure 3.4).
4. Finalement, la méthode MDS est appliquée à la matrice de distances $(dG_{ij}), i, j = 1, \cdots, n$ pour obtenir un nouveau système de coordonnées euclidiennes $Y \in R^r, (r < m)$ qui préserve la géométrie intrinsèque de ensemble des points. Parfois, des valeurs propres de petite magnitude sont obtenues (et ignorées) à la sortie du MDS : elles sont dues à des inconsistances mineures du calcul des distances géodésiques et/ou au bruit dans X.

Isomap combine les principales caractéristiques algorithmiques de l'ACP et MDS à savoir ; l'efficacité de calcul, l'optimalité globale, et les garanties de convergence asymptotique avec la flexibilité d'apprendre une large variété des classes non linéaires.

Le problème avec Isomap, vu qu' il utilise l'incorporation des caractéristiques, est qu'il place les N points dans un espace réduit, mais il n'apprend pas une fonction de projection(mapping) générale qui permettra de projeter un nouveau point de test ; Le nouveau point doit être ajouté à l'ensemble de données et l'algorithme entier doit être exécuté une fois de plus en utilisant $N + 1$ instances [Alpaydin 2014].

3.6 Travaux récents dans le domaine de sélection des caractéristiques textuelles

Dans cette section, nous présentons quelques travaux qui ont proposé des méthodes de sélection de caractéristiques contenues dans les textes ou dans la page web et qu'ils ont été testé sur la base WebKB et des Reuters et des autres corpus. Roberto et al. ont proposé une méthode de sélection de caractéristiques, de type filtrage, appelée ALOFT (At Least One FeaTure) (au moins une caractéristique) [Pinheiro 2012]. Cette approche assure que chaque document dans l'ensemble d'apprentissage est représenté au moins par une caractéristique et le nombre de caractéristiques sélectionnées d'une manière dirigée par les données.

Dans [Pinheiro 2015], les mêmes auteurs proposent une version améliorée de l'approche précédente. Dans cette nouvelle version, deux méthodes de filtrage sont utilisées pour la sélection des caractéristiques dans la classification de textes, à savoir :maximum f caractéristiques par document et maximum f caractéristiques par document-Réduit. Les deux algorithmes déterminent le nombre f de caractéristiques sélectionnées d'une manière dirigée par les données en utilisant une fonction d'évaluation du classement global des caractéristiques (global ranking Feature Evaluation Function (FEF)). Alors que le deuxième algorithme analyse uniquement les documents contenant des caractéristiques à forte valeur FEF pour sélectionner moins de caractéristiques, par conséquent, d'éviter ceux qui sont inutiles.

Un autre projet a été proposé dans [Yang 2014], afin d'améliorer les méthodes classiques de sélection des caractéristiques de type filtre(§ 3.3.1). Cette approche essaye de diminuer l'influence des facteurs de déséquilibres qui se produisent dans le corpus. Cette approche de sélection des caractéristiques est composée de trois étapes. La première étape consiste à calculer l'importance d'une caractéristique t_k sur une catégorie donnée c_i. La deuxième étape consiste à combiner les scores spécifiques d'une caractéristique sur les différentes catégories en un seul score global. La dernière étape consiste à classer toutes les caractéristiques de la base d'apprentissage selon leurs importances globales, puis sélectionner le k top caractéristiques importantes.

En outre, une approche composée de deux algorithmes, l'algorithme de sélection équitable d'un sous-ensemble des caractéristiques (fair feature subset selection algorithm (FFSS)) et un réseau d'apprentissage flou adaptatif (RAFA) pour la classification, a été proposé par Lee et al. dans [Lee 2001]. L'algorithme de FFSS est

utilisé pour la réduction de dimension. Selon les auteurs, cet algorithme donne non seulement un traitement équitable à chaque catégorie, mais il a aussi la capacité d'identifier les caractéristiques utiles, y compris les caractéristiques positives et négatives. D'autre part, le RAFA fournit la capacité d'apprentissage extrêmement rapide pour modéliser le comportement incertain pour la classification de manière à corriger la matrice floue automatiquement.

Dasgupta et al. proposé une stratégie non supervisée pour la sélection des caractéristiques [Dasgupta 2007]. Cette stratégie donne le pire des cas de la garantie théorique sur la puissance de généralisation de la fonction de classification résultante f' par rapport à la fonction de classification f obtenue quand on garde toutes les caractéristiques.

Mladenic et al. ont proposé l'utilisation des SVMs (Machines à vecteurs de support) linéaire pour la sélection de caractéristiques [Brank 2002]. Tout d'abord, ils forment l'SVM linéaire sur un sous-ensemble de données d'apprentissage et ne retenir que les caractéristiques qui correspondent à des composants fortement pondérés de la normale à l'hyperplan résultant. L'espace de caractéristiques réduit est ensuite utilisé pour former le classificateur sur un base d'apprentissage plus grande.

Par ailleurs, Kim et al. adopté une méthode de réduction de dimension qui réduit la dimension des vecteurs de documents [Kim 2005]. Les auteurs ont également introduit une fonction de décision pour l'algorithme de classification basé sur le barycentre (centroïd). Cette méthode utilise un classificateur SVM pour traiter le problème où un document peut appartenir à plusieurs classes.

Chih-Ming et al. ont proposé un paradigme de classement floue avec une nouvelle mesure de la pertinence appelée la mesure de puissance discriminante, pour réduire la dimension d'entrée [Chen 2009]. Le premier algorithme est utilisé pour réduire la dimension des données d'entrée. Cet algorithme donne un traitement équitable à chaque catégorie et identifie les caractéristiques utiles. Le second est utilisé pour la classification.

Dans [Yang 2012], les auteurs ont proposé une nouvelle approche de sélection des caractéristiques des pages Web qui s'appelle "Comprehensive Measurement Feature Selection (CMFS)". Cette approche propose de mesurer de l'importance globale d'un terme dans un corpus. Cela signifie que CMFS calcule l'importance d'un terme à la fois de à l'intérieur d'une catégorie (intra-catégorie) et entre les catégories (inter-catégorie). La mesure complète pour chaque terme t_k par rapport à la catégorie c_i est définie comme suit :

$$\begin{aligned} CMFS(t_k, c_i) &= \frac{tf(t_k, c_i) + 1}{tf(t_k) + |C|} \times \frac{tf(t_k, c_i) + 1}{tf(t, c_i) + |V|} \\ &= \frac{(tf(t_k, c_i) + 1)^2}{(tf(t_k) + |C|)(tf(t, c_i) + |V|)} \end{aligned} \tag{3.17}$$

Où $tf(t_k, c_i)$ est la fréquence d'un terme t_k dans la catégorie c_i ; $Tf(t_k)$ est la fréquence d'un terme t_k dans la base d'apprentissage ; $tf(t, c_i)$ est la somme de la fréquence de tous les termes de la catégorie c_i ; $|C|$ est le nombre de catégories ; $|V|$ est le nombre total de termes dans l'espace de caractéristiques.

Une amélioration de cette approche a été proposée dans [Feng 2015] dont les auteurs ont proposé d'introduire la taille de catégorie et la distribution de terme comme des facteurs dans le calcule de CMFS.

3.7 Conclusion

Les techniques de réduction de la dimension fondées sur des projections et des sélections sont en croissance très rapide dans la dernière décennie. Dans ce chapitre, nous avons exploré le domaine de la réduction de dimension, en exposant les deux approches qui sont couramment utilisées dans l'apprentissage automatique ; la sélection et l'extraction des caractéristiques. Nous avons aussi constater que chaque approche a son domaine d'application et ses avantages et inconvénients.

Le chapitre suivant sera consacré à la présentation de nos contributions que nous voulons à apporter sur les classificateurs des pages Web. Ces contributions sont la sélection et la pondération des caractéristiques en se basant sur l'algorithme HITS de Kleinberg.

Chapitre 4

Réduction et Pondération des caractéristiques en utilisant HITS

Sommaire

4.1 Introduction

L'analyse de contenu de pages Web nécessite des efforts intellectuels incroyables qui dépassent les capacités humaines. En général, n'importe quel processus d'analyse de données Web doit gérer des millions de pages Web, des dizaines de milliers de caractéristiques ou des termes et des centaines de catégories. Mais, il doit signaler que l'immense majorité de ces données sont redondantes et contient beaucoup de bruits qui sont sans intérêt pour les utilisateurs et peuvent enterrer les résultats de recherche souhaités, ou influencer sur les précision de la classification. Nous nous sommes donc intéressés au problème du prétraitement de ces données dont il était nécessaire de proposer des méthodes innovantes pour la sélection de caractéristiques.

Dans ce chapitre, nous allons présenter nos contributions dans la sélection d'un sous ensemble de caractéristiques discriminantes et non redondantes. Cette contribution est basée sur la valeur autorité, qui est calculée à l'aide de l'algorithme

d'analyse des liens HITS. Notre deuxième contribution consiste à utiliser la valeur autorité pour calculer les cœfficients de pondération (les poids) des caractéristiques des pages Web.

Le reste de ce chapitre est organisé comme suit : dans la section suivante, nous allons exposer le principe de fonctionnement des machines à vecteurs des supports (SVM). La troisième section sera consacrée à l'exposition détaillée de déroulement de notre approche, dont nous allons expliquer chaque phase à part. Ce chapitre sera terminé par une conclusion.

4.2 Rappel sur les machines à vecteurs de support (SVM)

4.2.1 Motivation sur l'utilisation des SVM

Les machines à vecteurs de support (Support Vector Machines (SVM)) ou séparateurs à vaste marge sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression [Scholkopf 2001]. Les SVM sont une généralisation des classifieurs linéaires. Les SVM ont été développés dans les années 1990 sur la base du principe de minimisation du risque structurel de la théorie de l'apprentissage statistique de Vladimir Vapnik [Vapnik 1982, Vapnik 2013].

Les SVM fournissent une approche très intéressante de l'approximation statistique. Souvent, le nombre des exemples pour l'apprentissage est insuffisant pour que les estimateurs fournissent un modèle avec une bonne précision. D'un autre côté, l'acquisition d'un grand nombre d'exemples s'avère être souvent très coûteuse et peut même mener à des problèmes de sur-apprentissage dans le cas ou la capacité du modèle est très complexe. Pour ces deux raisons, il faut arriver à un compromis entre la taille des échantillons et la précision recherchée. Dans ces cas spécifiques comme la reconnaissance de formes, il serait intéressant de trouver une mesure de la fiabilité de l'apprentissage, et d'avoir une mesure du taux d'erreur qui sera commis durant la phase de test [Kharroubi 2002].

4.2.2 Principe de fonctionnement des SVM

L'idée principale des SVMs est de construire un hyperplan qui est considéré comme une surface de décision dont le but est de maximiser la marge de séparation entre les exemples positifs et négatifs [Burges 1998] :

$$\{(x_1, y_1), \cdots, (x_i, y_i)\} \quad tel \quad que \quad x_i \in R^n \quad et \quad y_i \in \{+1, -1\} \tag{4.1}$$

Supposons que nous avons un hyperplan qui sépare le exemples positifs x^+ des exemples négatifs x^- c-à-d un " hyperplan de séparation ". Donc, on doit trouver

une fonction linéaire f permettant de séparer les deux classes (Figure 4.1) :

$$f(x) = y = wx + b \tag{4.2}$$

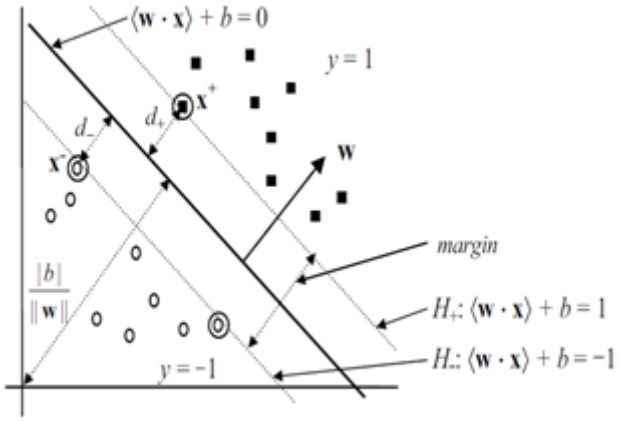

FIGURE 4.1 – Principe de machines à vecteurs de support [Liu 2007]

Les points qui se trouvent sur l'hyperplan satisfont $wx+b = 0$, où w est le normal de l'hyperplan, $||w||$ est la norme euclidienne de l'hyperplan p, $\frac{|b|}{||w||}$ est la distance perpendiculaire de l'hyperplan à l'origine. Soit d^+ (d^-) la distance la plus courte entre l'hyperplan de séparation et les exemples positifs les plus proches (négatifs). Donc La distance $d^+ + d^- = \frac{2}{||w||}$ s'appelle ***la marge***

Supposons que toutes les données de l'apprentissage satisfont les contraintes :

$$wx_i + b \geq 1 \quad si \quad y_i = 1 \tag{4.3}$$

$$wx_i + b \leq -1 \quad si \quad y_i = -1 \tag{4.4}$$

Les deux contraintes(4.3) et (4.4) peuvent être combinées en une seule contrainte :

$$y_i(wx_i + b) \geq 1 \quad si \quad \forall \quad y_i \tag{4.5}$$

Donc, l'objectif des SVMs est de trouver une marge la plus large possible entre les deux classes, ce qui peut être considéré comme un problème d'optimisation. Étant donné que la maximisation de la marge revient à minimiser $\frac{1}{2}||w||^2$.

Donc, l'apprentissage consiste à résoudre le problème de minimisation de la contrainte suivante :

$$\begin{cases} minimisr : & \frac{||w||^2}{2} \\ Avec : & y_i(wx_i + b) \geq 1 \quad si \quad \forall \quad y_i \end{cases} \tag{4.6}$$

Le problème (4.6) est un problème de programmation quadratique avec contraintes linéaires. Généralement, le nombre de variables est important ce qui ne permet pas d'utiliser les techniques classiques de programmation quadratique. Dans ce cas, le problème (4.6) est convertit en un problème dual équivalent sans contraintes de l'équation (4.6). On introduit les multiplicateurs de Lagrange positif $\alpha_i, i = 1, \cdots, L$, une pour chacune des contraintes (Eq.4.7). Les équations de contrainte sont multipliées par des multiplicateurs de Lagrange positifs et soustraites de la fonction objectif, pour former le lagrangien :

$$Q(w, b, \alpha) = \frac{||w||^2}{2} - \sum_{i=1}^{n} \alpha_i [y_i(wx_i + b) - 1] \tag{4.7}$$

La théorie de l'optimisation dit qu'une solution optimale de (Eq. 4.7) doit satisfaire à certaines conditions, appelées conditions de Karush-Kuhn-Tucker (KKT) [Kuhn 1951] :

$$\frac{\delta Q(w, b, \alpha)}{\delta w_j} = w_j - \sum_{i=1}^{n} y_i \alpha_i x_{ij} = 0, j = 1, \cdots, r \tag{4.8}$$

$$\frac{\delta Q(w, b, \alpha)}{\delta b} = -\sum_{i=1}^{n} y_i \alpha i = 0, i = 1, 2, \cdots, n \tag{4.9}$$

$$y_i(wx_i + b) - 1 \geq 0, i = 1, 2, \cdots, n \tag{4.10}$$

$$\alpha_i \geq 0, i = 1, 2, \cdots, n \tag{4.11}$$

$$\alpha_i(y_i(wx_i + b) - 1) = 0 \geq 0, i = 1, 2, \cdots, n \tag{4.12}$$

De (Eq.4.8) on déduit :

$$w_j = \sum_{i=1}^{n} y_i \alpha_i x_{ij}, j = 1, \cdots, r \tag{4.13}$$

$$\sum_{i=1}^{n} y_i \alpha_i = 0, i = 1, 2, \cdots, n \tag{4.14}$$

En remplaçant l'équation (Eq.4.13) dans l'équation (Eq.4.7), on obtient le problème dual à maximiser suivant :

$$\begin{cases} Maximiser : \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j < x_i, x_j > \\ Avec : \sum_{i=1}^{n} y_i \alpha_i = 0, \\ \quad \alpha_i \geq 0, i = 1, 2, \cdots, n \end{cases} \tag{4.15}$$

Après la résolution de l'équation (Eq.4.15), on obtient les valeurs de α_i, qui sont utilisés pour calculer le vecteur de poids w et le biais b en utilisant les équations (4.8) et (4.10) respectivement. Donc, la fonction de décision sera comme suit :

$$f(x) = wx + b = \sum_{i \in sv} y_i \alpha_i < x_i, x_i > +b = 0 \tag{4.16}$$

La classification d'un nouvel exemple z, consiste à calculer le signe de la fonction de décision (4.16)] , comme suit :

$$y = sign(\sum_{i \in sv} y_i \alpha_i < x_i, z > +b) \tag{4.17}$$

4.2.2.1 Marges Souples

Le cas linéairement séparable est la situation idéale. En pratique, les données d'apprentissage sont presque toujours bruité, c-à-d, contenant des erreurs dues à des différentes raisons. Par exemple, problème d'étiquetage.

Dans le cas où les données d'apprentissages ne sont pas linéairement séparables, les SVMs proposent de relâcher les contraintes de marge en introduisant des variables d'écart $\xi_i \geq 0$ par rapport aux frontières de la marge de séparation avec un paramètre de pénalisation C, comme suit (Figure 4.2) :

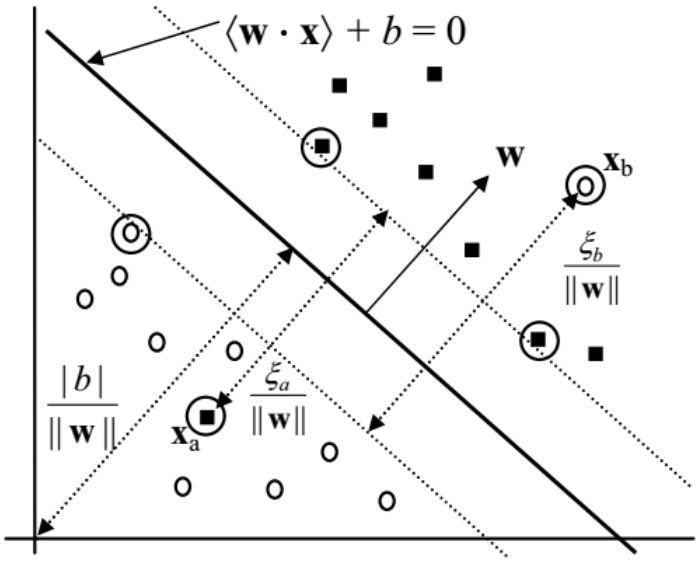

FIGURE 4.2 – Principe de machines à vecteurs de support dans le cas de la marge souple [Liu 2007]

$$wx_i + b \geq 1 - \xi_i \quad si \quad y_i = 1 \tag{4.18}$$

$$wx_i + b \leq -1 + \xi_i \quad si \quad y_i = -1 \tag{4.19}$$

Les deux contraintes(4.18) et (4.19) peuvent être combinés en une seule contrainte :

$$\begin{aligned} y_i(wx_i + b) &\geq 1 - \xi_i \quad , i = 1, 2, \cdots, n \\ \xi_i &\geq 0, \quad i = 1, 2, \cdots, n \end{aligned} \tag{4.20}$$

On a aussi également besoin de pénaliser les erreurs dans la fonction objective :

$$minimiser : \quad \frac{||w||^2}{2} + C(\sum_{i=1}^{n} \xi_i) \tag{4.21}$$

Où $C \geq 0$ est un variable introduit par l'utilisateur.

En combinant (4.20) avec l'équation (4.21), et on obtient ce qu'on appelle **SVM à marge souple** :

$$\begin{cases} minimisEr : \quad \frac{||w||^2}{+} C(\sum_{i=1}^{n} \xi_i) \\ Avec : y_i(wx_i + b) \geq 1 - \xi_i \quad , i = 1, 2, \cdots, n \\ \qquad\qquad\qquad\qquad \xi_i \geq 0, \quad i = 1, 2, \cdots, n \end{cases} \tag{4.22}$$

$$Q(w, b, \alpha) = \frac{||w||^2}{2} + C(\sum_{i=1}^{n} \xi_i) - \sum_{i=1}^{n} \alpha_i [y_i(wx_i + b) - 1 + \xi_i] - \sum_{i=1}^{n} \mu_i \xi_i \tag{4.23}$$

Où $\alpha_i, \mu_i \geq 0$ sont des multiplicateurs de Lagrange. Les conditions KKT pour le problème primal (Eq. 4.23) sont donc :

$$\frac{\delta Q(w, b, \alpha)}{\delta w_j} = w_j - \sum_{i=1}^{n} y_i \alpha_i x_{ij} = 0, j = 1, \cdots, r \tag{4.24}$$

$$\frac{\delta Q(w, b, \alpha)}{\delta b} = -\sum_{i=1}^{n} y_i \alpha i = 0, i = 1, 2, \cdots, n \tag{4.25}$$

$$\frac{\delta Q(w, b, \alpha)}{\delta \xi i} = C - \alpha_i - \mu_i = 0 \tag{4.26}$$

$$y_i(wx_i + b) - 1 + \xi_i \geq 0, i = 1, 2, \cdots, n \tag{4.27}$$

$$\alpha_i \geq 0, i = 1, 2, \cdots, n \tag{4.28}$$

$$\xi_i \geq 0, i = 1, 2, \cdots, n \tag{4.29}$$

$$\mu_i \geq 0, i = 1, 2, \cdots, n \tag{4.30}$$

$$\alpha_i(y_i(wx_i + b) - 1 + \xi_i) = 0, i = 1, 2, \cdots, n \tag{4.31}$$

$$\mu_i \xi_i = 0, i = 1, 2, \cdots, n \tag{4.32}$$

On déduit :

$$\begin{cases} w_j = \sum_{i=1}^{n} y_i \alpha_i x_{ij} \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \alpha_i + \mu_i = 0 \end{cases} \tag{4.33}$$

En remplaçant l'équation (4.33) dans 4.23, on obtient le problème dual suivant :

$$\begin{cases} Maximiser : \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j < x_i, x_j > \\ Avec : \sum_{i=1}^{n} y_i \alpha_i = 0, \\ \quad 0 \leq \alpha_i \leq C, i = 1, 2, \cdots, n \end{cases} \tag{4.34}$$

A partir des équations (4.24),(4.31) et (4.32), on peut conclure que les α_i ne peuvent pas dépasser C et ils peuvent être dans l'un des trois cas suivants :

$$\begin{cases} \alpha_i = 0 \Rightarrow y_i(wx_i + b) \geq 1 \quad and \quad \xi_i = 0 \\ 0 \leq \alpha_i \leq C \Rightarrow (y_i(wx_i + b) = 1 \quad and \quad \xi_i = 0 \\ \alpha_i = C \Rightarrow y_i(wx_i + b) \leq 1 \quad and \quad \xi_i \geq 0 \end{cases} \tag{4.35}$$

La fonction de décision est alors calculée de la même manière que dans le cas des SVMs à marge dure mais uniquement à base des exemples qui ont la valeur $\alpha_i > 0$ et $\alpha_i < C$, c-à-d des vecteurs supports non bornés :

$$f(x) = \sum_{i \in sv} y_i \alpha_i < x_i, x_i > + b \tag{4.36}$$

Enfin, Il faut préciser que la définition de la valeur du paramètre C reste toujours un problème. Alors, la solution simple consiste à construire plusieurs classificateurs avec différentes valeurs de paramètre C Ensuite, on choisi celui qui donne le meilleur résultat de classification

4.2.2.2 Cas non linéairement séparable

La procédure de recherche de l'hyperplan séparateur telle qu'elle est présentée ci-dessus ne permet de résoudre que des problèmes linéairement séparables. Cependant, pour plusieurs ensembles de données réelles, les limites de décision ne sont pas linéaires. Afin de remédier ce problème, les SVMs proposent une solution consiste à transformer les données dans l'espace d'entrée X dans un espace caractéristique F, de dimension supérieure (éventuellement de dimension infinie) [Burges 1998], à l'aide d'une fonction non linéaire ϕ :

$$\begin{aligned} \phi : & X \longrightarrow F \\ & x \mapsto \phi(x) \end{aligned} \tag{4.37}$$

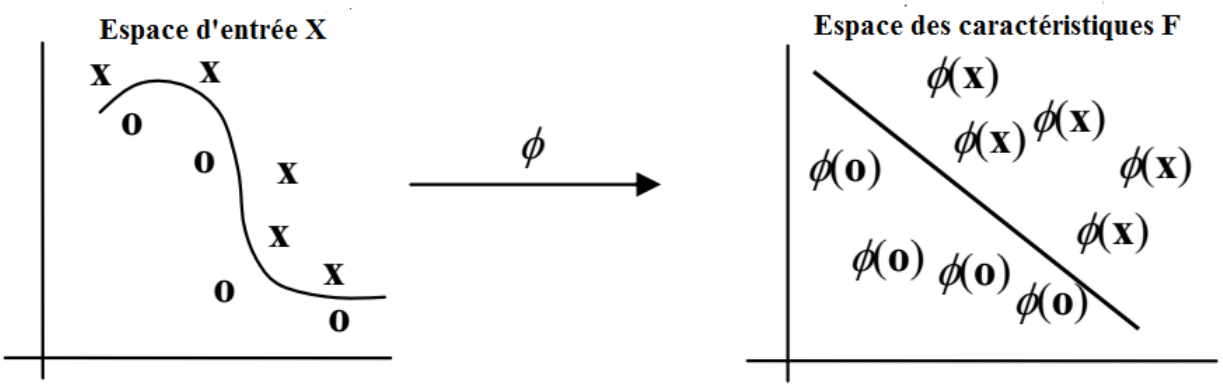

FIGURE 4.3 – Les données sont non linéairement séparables dans l'espace de données X mais sont linéairement séparables dans l'espace de caractéristiques F [Liu 2007]

Avec la transformation, le problème d'optimisation (4.22) devient :

$$\begin{cases} minimisEr : \quad \dfrac{||w||^2}{+} C(\sum_{i=1}^{n} \xi_i) \\ Avec : y_i(w\phi(x_i) + b) \geq 1 - \xi_i \quad , i = 1, 2, \cdots, n \\ \qquad\qquad\qquad\qquad\qquad \xi_i \geq 0, \quad i = 1, 2, \cdots, n \end{cases} \tag{4.38}$$

Son problème duale correspondant est :

$$\begin{cases} Maximiser : \sum_{i=1}^{n} \alpha_i - \dfrac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j < \phi(x_i), \phi(x_j) > \\ Avec : \sum_{i=1}^{n} y_i \alpha_i = 0, \\ \qquad 0 \leq \alpha_i \leq C, i = 1, 2, \cdots, n \end{cases} \tag{4.39}$$

La règle de la décision finale est :

$$f(x) = \sum_{i \in sv} y_i \phi_i < \phi(x_i), \alpha(z) > +b \tag{4.40}$$

Les transformations explicites peuvent être évitées si l'on remarque que, dans toutes les représentations duales de la construction de l'hyperplan optimal dans F (Eq.4.39) et l'évaluation de la fonction décision correspondante (Eq.4.40)ne nécessitent que l'évaluation de produit scalaire $< \phi(x), \phi(z) >$. Cette stratégie consiste à utiliser directement une fonction de noyau pour remplacer les produits scalaire dans l'espace de caractéristiques est appelée *l'astuce du noyau*(en anglais : Kernel Trick).

l'astuce du noyau « Tout modèle ne nécessitant dans sa construction que la manipulation de produit scalaire entre observations (et non leur cordonnées explicites) peut être construit implicitement dans un espace de Hilbert en remplaçant chaque produit scalaire par l'évaluation d'un noyau défini positif sur un espace quelconque ».

A partir de ce qui précède, on conclu que pour qu'une fonction K soit un noyau, il faut qu'elle respecte les conditions de Mercer [Mercer 1909, Cristianini 2000], c-à-d qu'elle doit être semi-définie positive (symétrique et n'a pas de valeurs propres négatives).

Une fois le noyau a été choisi, la fonction objective (Eq.4.40) peut être écrit comme suit :

$$f(x) = \sum_{i \in sv} y_i \alpha_i < \phi(x_i), \phi(z) > + b = \sum_{i \in sv} y_i \alpha_i k(x_i, z) + b \tag{4.41}$$

4.2.2.3 Exemples des fonctions noyaux

Les fonctions noyaux les plus fréquemment utilisées dans la littérature sont : le noyau linéaire, polynomial, triangulaire et le noyau Gaussien [Cristianini 2000]

— **Le noyau linéaire** :

$$K(V_1, V_2) =< V_1, V_2 > \tag{4.42}$$

Cette fonction exprime le produit scalaire usuel et elle nous permet de nous comparer à une utilisation sans approche noyau. Par exemple, elle exprimera la distance Euclidienne si la mesure de similarité est calculée à travers une distance Euclidienne.

— **Polynomiale** : de degré m :

$$K(V_1, V_2) =< V_1, V_2 >^m \tag{4.43}$$

— **Radial Basis Function (RBF)** :

$$K(V_1, V_2) = e^{\frac{-dist(V_1, V_2)}{2\sigma^2}} \tag{4.44}$$

tel que σ est le paramètre de l'échelle. Ce paramètre permet de définir d'une manière implicite l'espace de projection à partir duquel on déduit la mesure de similarité. Le noyau Gaussien dépend de la distance $dist(V_1, V_2)$, ce qui nous permettra de choisir la distance la plus appropriée aux données traitées. Avec distance euclidienne :

$$K(V_1, V_2) = e^{\frac{-||V_1 - V_2||^2}{2\sigma^2}} \tag{4.45}$$

4.3 Une architecture globale d'un classificateur des pages Web basée HITS

Un classificateur des pages Web est composé essentiellement de deux étapes successives : La préparation et l'apprentissage.

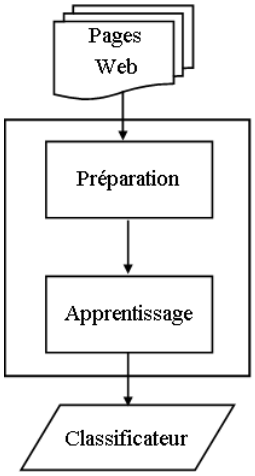

FIGURE 4.4 – Une vue globale d'un classificateur des pages Web

4.3.1 Préparation

Le rôle de cette phase est de recevoir les données et les préparer et les représenter sous une forme appropriée à la phase de l'apprentissage. Donc, la phase de préparation s'occupe de lecture des données (dans notre cas les pages Web) et de les faire subir par les différentes tâches de nettoyage afin d'obtenir un "sac à mots" (bag of words). La deuxième tâche qui est incluse dans la phase de préparation est la sélection de caractéristiques appropriées à la classification. La dernière étape de préparation consiste à la pondération des caractéristiques. Le rôle de cette étape est la transformation des caractéristiques symboliques (les termes) en d'autres numériques (les poids).

Il faut préciser que nos contributions font parties entrées dans la phase de préparation :

1. La représentation de la relation entres les pages Web et les termes qu'elles contiennent par un graphe biparti, où les documents pointent vers termes.
2. L'adaptation de l'algorithme de l'analyse de lien HITS pour être applicable sur les graphes bipartis.
3. La sélection des termes selon la valeur d'autorité.
4. L'utilisation des valeurs autorités pour le calcul des poids de pondération des caractéristiques .

4.3.2 Apprentissage

La classification des pages Web se situe dans le domaine de l'apprentissage automatique, où l'apprentissage se fait sur des pages Web. L'apprentissage sur les pages Web est similaire à l'apprentissage du texte puisque les pages Web peuvent être traitées comme des documents textuels. Néanmoins, il est clair que l'apprentissage sur les pages Web a de caractéristiques supplémentaires [Choi 2005] :

1. Les pages Web sont des documents texte semi-structurés qui sont habituellement écrits en HTML.
2. Les pages Web sont reliées entre elles formant des graphes orientés à l'aide des hyperliens.
3. Les pages Web sont souvent courtes et l'analyse de ces pages en utilisant seulement le texte peut être insuffisante.
4. Les sources des pages Web sont nombreuses, hétérogènes, distribuées et dynamiquement changeantes.

Au cours des dernières années, plusieurs méthodes de classification et des techniques d'apprentissage automatique ont été appliquées à la classification des pages Web, nous citons par exemple ; les modèles de régression multivariée, le plus proche voisin, les approches probabilistes de Bayes, arbres de décision, les réseaux de neurones, les machines à vecteurs de supports ...etc.

Enfin, Il faut préciser que le classificateur résultant doit être valider (tester) sur plusieurs ensembles de données (des pages Web) avant l'utiliser définitivement.

4.4 Une architecture détaillée améliorée d'un classificateur des pages Web

Notre approche est composée de six étapes successives(Figure 4.5) permettant de réduire la taille de vecteur des caractéristiques en utilisant un algorithmes d'analyse de lien qui est HITS. Ce dernier originalement est utilisé pour le classement des pages Web mais cette fois-ci nous proposons de l'utiliser pour le classement des caractéristiques contenues dans ces pages Web.

4.4.1 Préparation

Cette phase est composée de plusieurs étapes qui sont :

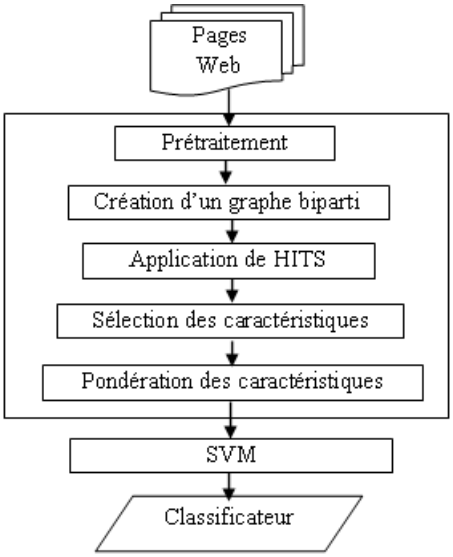

FIGURE 4.5 – Sélection de caractéristiques en utilisant HITS

4.4.1.1 Prétraitemts

La première étape de n'importe quel classificateur des pages Web c'est le prétraitement. Cette étape est considérée comme une première phase d'extraction des caractéristiques. Dans cette étape, plusieurs tâches ont été effectuées dont l'objectif est l'extraction de contenu d'une page Web. Afin de récupérer les caractéristiques textuelles importantes, les pages Web sont d'abord prétraitées pour annuler les données les moins importantes. Le prétraitement comporte les étapes suivantes :

1. Suppression des balises HTML : les balises HTML indiquent la structure et le format de pages Web. Par exemple, le contenu dans <title> et </ title> c'est le titre d'une page Web ; le contenu inclus dans <table> et </ table> est un tableau. Ces balises HTML peuvent indiquer l'importance de leur contenu et ils peuvent ainsi aider à pondérer leur contenu. Les balises elles-mêmes seront supprimées après la pondération de leur contenu.
2. Suppression de toutes les ponctuations dans le texte.
3. Suppression de tous les nombres existants dans le texte.
4. Mettre tout le texte dans la page Web en minuscule.
5. Suppression des mots vides (en anglais : Stopwords) : les mots vides sont des mots fréquents qui portent peu d'informations, telles que les prépositions, les pronoms et conjonctions. Ils sont éliminés en comparant le texte d'entrée avec une liste des mots vides.
6. Extraction de radicale des mots : cela se fait en regroupant les mots qui ont le même radicale ou de racine, tels que "computer", "compute", et "compu-

ting". L'algorithme Porter est un algorithme bien connu pour effectuer cette Tâche [Porter 1980].

7. Sélection des mots qui sont composés de plus de deux lettres.

4.4.1.2 Création d'un graphe biparti

Dans la théorie des graphes, un graphe est appelé biparti s'il y a une partition de l'ensemble de nœuds en deux sous-ensembles U et V, où chaque arête a une extrémité en U et l'autre en V (figure 4.6).

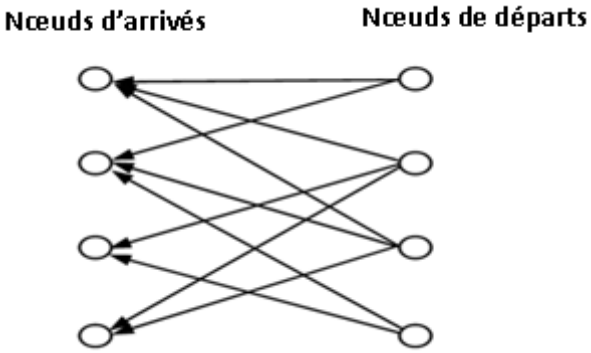

FIGURE 4.6 – Un graphe biparti dont chaque arête a une extrémité dans l'ensemble des nœuds de départs et l'autre dans l'ensemble des nœuds d'arrivés

A l'origine, l'algorithme HITS a été proposé pour analyser un graphe des pages Web, où les nœuds (pages Web) peuvent être des hubs ou autorités. Notre approche se démarre en posant l'hypothèse qui suppose l'existence d'une relation entre les pages Web et leurs caractéristiques. Cette relation peut être représentée par un graphe biparti dont les pages Web sont les hubs et les caractéristiques sont des autorités.

Dans cette phase, nous avons représenté les pages Web nettoyées obtenues à l'étape précédente, par un graphe biparti. Les nœuds de départ sont les pages Web et les nœuds d'arrivé sont les termes (les caractéristiques). La figure 4.7 représente une matrice générale (A [n, m]) qui représente notre graphe biparti. Où :

— N : représente le nombre de pages Web,
— M : représente la taille du vecteur de caractéristique.

Les valeurs de la matrice sont des valeurs binaires, où :
— $A[i, j]$ est mis 1 si la caractéristique j existe dans la page i.
— $A[i, j]$ est mis 0 autrement.

	t_1	t_2	t_3	t_4	.....	t_f
d_1	1	1	0	0	.....	0
d_2	0	1	1	1	.....	0
.					.....	.
.					.....	.
.					.....	.
.					.....	.
d_n	1	0	0	0		1

FIGURE 4.7 – Une représentation matricielle d'un graphe biparti dont les lignes sont les pages Web et les colonnes sont les termes.

4.4.1.3 Application de l'algorithme HITS

Originalement, l'algorithme HITS est appliqué pour calculer les poids associés aux pages Web pour les classer par la suite. Dans ce travail, nous proposons d'utiliser HITS pour calculer les poids des caractéristiques contenues dans les pages Web, dont chaque page Web est considérée comme hub et chaque caractéristique est un autorité, où les hubs pointent vers les autorités.

Donc, les poids des pages Web peuvent être décrits par un vecteur h de dimension $N(N = Nombrededocuments)$, où h_i est le poids hub de la page p_i et les poids des caractéristiques peuvent être décrits par un vecteur a de dimension m, où a_i indique la valeur d'autorité de la caractéristique. Alors, il y a deux formules à appliquer :

$$\begin{aligned} a_i &= \sum_{i \longrightarrow j} h_j \\ h_j &= \sum_{j \longrightarrow i} a_i \end{aligned} \tag{4.46}$$

Les deux relations précédentes (4.46) peuvent être écrites comme suit :

$$\begin{aligned} a &= A^t.h \\ h &= A.a \end{aligned} \tag{4.47}$$

Où A est la matrice représentant du graphe biparti (qui a été construit dans la phase précédente).

Une implémentation possible de l'algorithme HITS peut être comme suit :

A la fin, cet algorithme produit un vecteur a traduisant l'importance de chaque caractéristique dans le corpus d'entrée.

HubAvg HubAvg [Borodin 2005] est un autre algorithme d'analyse de liens basé sur le principe de renforcement mutuel entre les hubs et les autorités. Cependant,

Algorithme 4 Solution itérative de l'algorithme HITS

$h_0 \leftarrow (1, 1, \cdots, 1)$
$a_0 \leftarrow (1, 1, \cdots, 1)$
$k \leftarrow 0$
répéter
  $k \leftarrow k + 1$
  $a_k \leftarrow A^t h_{k-1}$
  $h_k \leftarrow A * a_k$
  $Normaliser\ les\ deux\ vecteurs\ (a_k\ et\ h_k)$
jusqu'à $|a_k - a_{k-1}| < \xi_a\ et\ |h_k - h_{k-1}| < \xi_h$
retour $a_k\ et\ h_k$

les auteurs de HubAvg utilisent une définition différente que celle de Kleinberg pour la notion d'un bon hub. En effet, alors que HITS considère qu'un bon hub est un nœud qui pointe vers de **bonnes** autorités.L'intuition de l'algorithme de HubAvg est qu'un bon hub doit pointer uniquement ou (au moins principalement) les bonnes autorités. En d'autres termes, cette définition signifie que les hubs qui pointent vers des nœuds ayant un faible degré d'autorité doivent être pénalisés.

Le degré d'hubité d'une nœuds j est calculé par HubAvg de la façon suivante :

$$\begin{cases} a_i = \sum_{i \longrightarrow j} h_j \\ h_j = \dfrac{1}{|F(i)|} \sum_{j \longrightarrow i} a_i \end{cases} \tag{4.48}$$

Où $F(i)$ est l'ensemble des nœuds qui sont pointé par le nœud i. l'algorithme HubAvg peut être résumé comme suit ;

Algorithme 5 Solution itérative de l'algorithme HubAvg

$h_0 \leftarrow (1, 1, \cdots, 1)$
$a_0 \leftarrow (1, 1, \cdots, 1)$
$k \leftarrow 0$
répéter
  $k \leftarrow k + 1$
  pour $i \leftarrow 1 \quad à \quad N$ **faire**
    $a_i k \leftarrow \sum_{i \longrightarrow j} h_{jk}$
  pour $j \leftarrow 1 \quad à \quad N$ **faire**
    $h_j k \leftarrow \dfrac{1}{|F(i)|} \sum_{j \longrightarrow i} a_{ik}$
  $Normaliser\ les\ deux\ vecteurs\ (a_k\ et\ h_k)$
jusqu'à $|a_k - a_{k-1}| < \xi_a\ et\ |h_k - h_{k-1}| < \xi_h$
retour $a_k\ et\ h_k$

Mais le problème avec HubAvg s'avère quand un nœud est en même temps un

hub fort sur un sujet et un hub faible sur un autre sujet. Ces hubs sont pénalisés par l'algorithme de HubAvg.

4.4.1.4 Algorithme seuil d'autorités AT(k))

Afin de réduire l'effet des autorités faibles sur le calcul du poids de hub, tout en conservant en même temps l'effet positif des autorités fortes. Borodin et al dans [Borodin 2005] ont proposé une solution qui consiste à appliquer un seuil qui ne retient que les poids de plus hautes autorités.

L'algorithme seuil d'autorités (Authority-Threshold AT(k)) qui définit le poids des nœuds comme la somme des k plus grand poids autorité des nœuds autorités pointées par le nœud i. Cela revient à dire qu'un nœud est un bon hub s'il pointe vers au moins k bonnes autorités. La valeur de k est passée en paramètre à l'algorithme.

$$\begin{cases} a_i = \sum_{i \longrightarrow j} h_j \\ h_j = \sum_{i \longrightarrow j / i \in F_k(i)} a_i \end{cases} \tag{4.49}$$

$F_k(i)$ désigne le sous-ensemble de $F(i)$ qui contient des k nœuds avec les poids d'autorité supérieure. L'algorithme AT(k) calcule les poids de l'autorité et du hub comme suit :

Algorithme 6 Solution itérative de l'algorithme AT(k)

$h_0 \leftarrow (1, 1, \cdots, 1)$
$a_0 \leftarrow (1, 1, \cdots, 1)$
$N \leftarrow 0$
répéter
  $P \leftarrow P + 1$
  pour $i \leftarrow 1 \quad à \quad N$ **faire**
    $a_i P \leftarrow \sum_{i \longrightarrow j} h_{jP}$
  pour $j \leftarrow 1 \quad à \quad N$ **faire**
    $h_j P \leftarrow \frac{1}{|F_(i)|} \sum_{j \longrightarrow i} a_{iP}$
  $Normaliser\ les\ deux\ vecteurs\ (a_P\ et\ h_P)$
jusqu'à $|a_P - a_{P-1}| < \xi_a\ et\ |h_P - h_{P-1}| < \xi_h$
retour $a_P\ et\ h_P$

4.4.1.5 Réduction de dimension du vecteur de caractéristiques

Au démarrage de cette phase, il faut d'abord trier le vecteur d'autorité dans l'ordre décroissant et ensuite déterminer un seuil t_a. Enfin, on supprime toutes les caractéristiques qui ont des valeurs autorité inférieure au seuil t_a.

La réduction de dimension du vecteur de caractéristiques peut conduire à un cas où il existe des pages Web qui n'ont aucun terme présent dans ce nouveau vecteur de caractéristiques alors il faut les supprimer de la base d'apprentissage. A la fin de cette phase, nous aurons effectué deux types de filtrage ; verticale et horizontale. Cette dernière est valable surtout dans le cas où on a éliminé un grand nombre de caractéristiques, ce qui produit une minimisation de la base d'apprentissage sur les deux axes ; côté caractéristiques et côté exemples d'apprentissage. Cette minimisation va accélérer les phases suivantes et surtout la phase d'apprentissage.

4.4.1.6 Calcul des poids des Caractéristiques

Un document dans le modèle vectoriel (voir § 2.8.3.2) est représenté par un vecteur des poids dans lequel chaque composant est un poids d'une caractéristique. A l'origine, ces poids sont calculés à l'aide modèle TFIDF.

Dans la cinquième étape de cette approche, nous proposons de calculer les poids en utilisant le vecteur de l'autorité qui est la sortie de l'algorithme HITS au lieu d'utiliser le schéma TFIDF(voir le paragraphe 2.8.3.2).

Soit a_i la valeur de l'autorité du terme (caractéristique) t_i et n_{ij} est le nombre d'occurrences du terme t_i dans le document d_j. Alors, le poids du terme t_i dans le document d_j (notée w_{ij}) est donné par :

$$w_{ij} = \begin{cases} log(a_i \times n_{ij}) & si \quad n_{ij} \neq 0 \\ 0 \quad si \qquad n_{ij} = 0 \end{cases} \tag{4.50}$$

4.4.2 Apprentissage de SVM

Après l'étape de prétraitement, nous passons à la phase de création d'un classificateur automatique des pages Web en utilisant un algorithme d'apprentissage donné. Parmi les algorithmes d'apprentissage et de la classification des données, nous proposons l'utilisation des SVM (Support Vector machine) [Vapnik 1982, Vapnik 2013], parce que par rapport à d'autres types, les classificateurs SVM sont à la fois efficaces et compétents. Les classificateurs basés sur les SVM ont montré des résultats prometteurs dans la classification de textes et des pages Web [Joachims 1998, Sun 2002].

Avant de lancer l'apprentissage, il faut procéder d'abord une tâche de normalisation des poids, (résultant de la phase précédente) c'est-à-dire la transformation des valeurs de poids pour être appartient à l'intervalle $[-1, 1]$. La valeur de normalisation $norm_w$ d'un poids w est obtenu par la formule suivante :

$$norm_w = 2 \times \frac{w_{ij} - min_w}{max_w - min_w} - 1 \tag{4.51}$$

4.5 Conclusion

Dans ce chapitre, nous avons introduit notre nouvelle approche de création d'un classificateur des pages Web en se basant sur nos contributions. Notre première contribution consiste dans l'utilisation de l'un des algorithmes les plus connus de l'analyse des liens, qui est HITS, dans la sélection des caractéristiques. Nous avons proposé de sélectionner les caractéristiques contenues dans les pages selon leurs valeurs d'autorités. Dans notre contribution, nous avons proposé l'utilisation des valeurs autorités des caractéristiques restantes pour calculer les poids de ces caractéristiques. À première vue, il semble que notre proposition a un certain nombre d'avantages :

1. La simplicité puisque HITS est un algorithme simple à comprendre, à expliquer et à implémenter.
2. La sélection des caractéristiques et la pondération qui se basent sur le même algorithme c-à-d nous n'avons pas besoin d'importer d'autres techniques de sélection des caractéristiques. Par contre dans le cas du modèle TFIDF, nous avons besoin d'importer des techniques de réduction, comme par exemple chi-square ou information gain.

Pour valider notre proposition, le chapitre suivant sera consacré à la présentation des résultats obtenus de différentes expérimentations que nous avons effectué sur quelques ensembles de documents.

Chapitre 5

Expérimentations et discussion

Sommaire

5.1 Introduction

Dans le chapitre précédent, nous avons présenté une approche de construction d'un classificateur des pages Web. Nous avons proposé comme contribution l'utilisation de l'algorithmes HITS pour la sélection et la pondération des caractéristiques contenues des pages Web. Dans ce chapitre, nous allons exposer les différentes expérimentations que nous avons effectué sur un ensemble des corpus référentiels.

Dans la section suivante, nous allons présenter les différents ensembles des documents qui nous avons utiliser pour valider notre approche. Ensuite, dans la section (5.3), nous exposons le matériel et les différents outils que nous avons utilisé pour l'implémentation de notre système. les résultats des différentes expérimentations seront analysés et évalués dans la section (5.4). Ce chapitre se termine par une conclusion.

5.2 Description des données

Nos expériences ont été conduites sur un ensemble de pages Web extraites du projet WebKB [Craven 1998]. Ce dernier contient des pages Web collectées auprès des départements d'informatique de 04 universités (*Cornell, Texas, Washington et Wisconsin*) en Janvier 1997. Les pages Web de ce projet ont été classées manuellement en 07 catégories : *student, faculty, staff, department, course, project et others.* Parmi ces classes, Nous avons choisi de tester notre classificateur sur trois classes, à savoir ; *course*, *students*, *Faculty* (voir table (5.1)).

La deuxième base de données utilisée est Reuters-21578 (R8 et R52) [1]. Cette base de documents est très connue dans le domaine de la catégorisation des textes et Web mining. L'ensemble Reuters-21578 R8 se compose de 7674 documents regroupés en 08 classes de sujets (acq, crude, earn, grain, interest, money-fx, ship et trade). Parmi ces dernières, nous avons choisi arbitrairement la classe *acq* comme classe positive et le reste en tant que classe négative (voir table (5.1)). L'ensemble Reuters-21578 R52 est un ensemble ds document contient 9100 documents concernent 52 catégories de sujets (acq, earn, etc ...). Dans ce cas, la classe *earn* était la classe positive.

TABLE 5.1 – Description des bases d'apprentissage et de tests des différents corpus utilisés dans nos expérimentations

	base d'apprentissage				Base des Testes		
	Total	Positives	Négatives	Termes	Total	Positives	Négatives
Webkb-course	2785	620	2165	7139	1383	306	1077
Webkb-faculty	2785	745	2040	7139	1383	372	1011
Webkb-student	2785	1085	1700	7139	1383	540	843
Reuters-21578 R8	5485	1596	14623	3889	2198	696	1493
Reuters-21578 R52	6532	2840	15869	3691	2568	1083	1485

5.3 Outils de validation

5.3.1 Matériel

Les expériences ont été effectuées sur une machine qui comporte un processeur modèle *Intel i5*, avec une mémoire centrale de 4 Go sous le système d'exploitation Microsoft Windows 7. Les modules de notre approche on été implémentés en langage C (en utilisant l'environnement Borland Builder C++).

1. http ://www.cs.umb.edu/ smimarog/textmining/datasets/

5.3.2 Algorithmes d'apprentissage

Nous avons utilisé les algorithmes d'apprentissage (SVM, Arbre décisionnel et bayésien Naïf) et les méthodes de sélection de caractéristiques ("Chi-square" et "Information gain"), qui sont implémentées dans le logiciel de data mining Weka [2] [Witten 2005]. Ce dernier est une suite de logiciels d'apprentissage automatique écrite en langage Java et développée à l'université de Waikato en Nouvelle-Zélande. Weka est un logiciel libre disponible sous la Licence publique générale GNU (GPL).

Nous utilisé le SMO [Platt 1998] comme une implémentation des SVM. Après une série de tests, nous avons sélectionné le meilleur modèle de notre SVM. Ce modèle est composé d'un noyau, où nous avons utilisé essentiellement le noyau polynomial (et RBF), la valeur de l'exposant δ et la valeur du paramètre C qui a été fixé à 1.

5.3.3 Mesure de validation

Les classificateurs des page Web sont évalués en utilisant des mesures standard qui à l'origine sont utilisées pour l'évaluation des systèmes de recherche d'information qui sont ; F-mesure (voir la section 2.8.4) et le taux de réussite qui nous avons le définit comme suit :

$$Taux\ de\ réussite = \frac{Nombre\ des\ document\ bien\ classés}{Nombre\ total\ des\ documents} \tag{5.1}$$

5.4 Expérimentations

5.4.1 Utilisation des SVM pour la classification des Pages Web

L'objectif de cette expérimentation est de vérifier quel algorithme d'apprentissage est approprié et donne des meilleurs résultats pour la classification des pages Web. Pour cela, nous avons proposé de construire trois classificateurs des pages Web, où chacun utilise algorithme d'apprentissage différent. Ces algorithmes sont : bayésien Naïf , l'arbre de décision et SVM.

Dans cette expérimentation, nous avons utilisé le modèle vectoriel pour représenter les documents Web dont les poids sont calculés en utilisant le modèle TFIDF. Les résultats obtenus de cette expérimentation sont résumés dans (table 5.2) :

A partir de la table 5.2, nous trouvons que, pour :

2. https ://fr.wikipedia.org/wiki/Weka_(informatique)

TABLE 5.2 – Récapitulatif des résultats obtenus par les classificateurs des pages Web basés sur le modèle TFIDF

		course	student	faculty	R8	R52
Bayes Naïf	Précision	84.53%	86.19%	76.50%	96.07%	82.87%
	F-measure	0.854	0.862	0.776	0.961	0.829
Arbre de Décision	Précision	95.01%	90.02%	89.30%	94.88%	96.69%
	F-measure	0.95	0.9	0.89	0.948	0.967
SMO	SV	665	1186	1459	1261	1237
	Précision	97.04%	92.19%	92.99%	98.36%	98.95%
	F-measure	0.97	0.921	0.929	0.984	0.989

— **L'ensemble WebKb-course :** le classificateur des pages Web basé SVM donne les meilleurs résultats de précision (97.04%) par rapport à ceux qui se basent sur l'algorithme naïf bayésien (84.53%) ou l'arbre décisionnel (95.01%).
— **L'ensemble WebKb-student :** la précision de classificateur basé SVM (92.19%) est meilleure que celle d'un classificateur qui se base sur l'arbre décisionnel(90.02%) ou celle de l'autre qui se base sur l'algorithme naïf bayésien (86.19%).
— **L'ensemble WebKb-faculty :** le classificateur des pages Web basé SVM donne une meilleure précision (97.04%) par rapport à ceux qui se basent sur l'algorithme naïf bayésien (84.53%) ou l'arbre décisionnel(95.01%).
— **L'ensemble Reuters-R8 :** le classificateur des pages Web basé SVM est plus précis (98.36%) que ceux utilisant les algorithmes de classification naïf bayésien (96.07%) et l'arbre décisionnel(94.88%).
— **L'ensemble Reuters-R52 :** le classificateur des pages Web basé SVM donne une précision (98.95%) meilleure que ceux qui se basent sur l'algorithme naïf bayésien (82.87%) ou l'arbre décisionnel(96.69%).

A partir de ce qui précède, nous pouvons conclure que les SVMs sont beaucoup plus appropriées pour la classification des pages Web.

5.4.2 La pondération des caractéristiques des pages Web par HITS

Dans cette expérimentation, nous avons essayé de répondre à la question : Est ce qu'il est possible d'utiliser les sorties de l'algorithme HITS pour calculer les poids des caractéristiques contenues dans les pages Web ? Pour cela, nous avons répété l'expérimentation précédente mais cette fois-ci nous avons remplacé le schéma TFIDF par un autre schéma qui utilise le vecteur autorité pour calculer les poids des caractéristiques, comme suit :

$$w_{ij} = \begin{cases} log(a_i \times n_{ij}) & si \quad n_{ij} \neq 0 \\ 0 \quad si \qquad n_{ij} = 0 \end{cases} \tag{5.2}$$

Les résultats obtenus de cette expérimentation sont résumés dans la table 5.3 :

TABLE 5.3 – Récapitulatif des résultats obtenus par les classificateurs des pages utilisant le modèle HITS pour la pondération des caractéristiques

		course	student	faculty	R8	R52
Bayes Naïf	Précision	88.21%	83.66%	78.81%	94.20%	89.68%
	F-measure	0.889	0.835	0.798	0.942	0.897
Arbre de Décision	Précision	95.08%	88.36%	88.86%	94.84%	96.85%
	F-measure	0.95	0.883	0.888	0.984	0.969
SMO	SV	432	1503	1236	848	1043
	Précision	97.69%	94.65%	94.58%	98.04%	98.87%
	F-measure	0.977	0.946	0.944	0.98	0.989

En analysant les tableaux (table 5.2 et table 5.3), nous trouvons que, pour :

- **la base *course*** : Les classificateurs (SVM, arbre décisionnel et Bayes) basés sur HITS dominent ceux qui se basent sur le modèle TFIDF.
- **la base *Student*** : Le classificateur basé HITS est plus précis dans le cas de l'algorithme SVM (94.65% *vs* 92.19%) mais un peu moins précis dans le cas de l'arbre de décision (88.36% Vs 90.02%) et de Bayes (83.66% vs 86.19%).
- **la base *faculty*** : Le classificateur basé sur HITS donne la meilleure précision pour l'algorithme Bayes (78.81% Vs 76.50%) et SVM (94.58% Vs 92.99%) mais avec l'arbre de décision le classificateur basé TFIDF est plus précis (89.30%) par rapport à celui basé HITS (88.86%).
- **la base *Reuters-R8*** : Notre proposition donne un taux de classification moins si nous utilisons l'algorithme Bayes (94.20% Vs 96.07%). Les taux de classification obtenus par les deux approches en utilisant l'arbre décisionnel sont approximativement égaux (94.84% Vs 94.88%). En utilisant SVM, l'approche basée TFIDF génère un classificateur précis légèrement (98.36%) par rapport au classificateur basé HITS (98.04%).
- **la base *Reuters-R52*** : Le classificateur basé HITS est plus précis, si nous utilisons Bayes(89.68% Vs 82.87%) et arbre de décision (96.85% Vs 96.69%) et moins précis si nous utilisons SVM (98.87% Vs 98.95%).

D'après la table 5.2 et la table 5.3, nous pouvons conclure que le vecteur autorité peut être utilisé normalement pour le calcul des poids des caractéristiques contenues dans les pages Web. Donc, l'algorithme HITS peut être considéré comme concurrent du modèle TFIDF.

5.4.3 La réduction des caractéristiques des pages Web par HITS

Dans cette section, nous comparons l'algorithme HITS, comme un algorithme de sélection de caractéristiques avec les méthodes "Chi-square" (χ^2) et "Information

gain". Dans cette expérimentation, nous avons testé notre approche avec plusieurs valeurs de seuils (Ta). Pour les deux autres méthodes, nous avons fixé le seuil à zero, c-à-d nous avons éliminé toutes les caractéristiques qui ont des valeurs de classement (Ranking value) inférieure ou égale à zéro. En outre, pour cette comparaison, nous avons utilisé également deux systèmes de pondération différents qui sont TFIDF et HITS.

Pour effectuer une meilleure comparaison, nous avons appliqué ces méthodes de sélection (notre approche, "Chi-square", Information Gain) sur les ensembles de données précédents (table 5.1), avec différents algorithmes de classification (Naïf Bayésien, arbre de décision, SVM). Nous avons rapporté les résultats obtenus dans les tables (5.4 à 5.13).

TABLE 5.4 – Résultats de classification de la base des documents course après la réduction en utilisant le modèle de pondération TFIDF.

		Chi-2 TH=0	Info-gain TH=0	HITS					
				8×10^{-5}	14×10^{-5}	18×10^{-5}	23×10^{-5}	28×10^{-5}	47×10^{-5}
Pages Web		2779	2779	2785	2785	2784	2783	2781	2780
Termes		1012	1012	2160	1399	1128	913	784	463
Bayes Naïf	Précision	92.48%	92.48%	84.45%	87.49%	88.21%	89.73%	91.18%	91.47%
	F-measure	0.928	0.928	0.853	0.881	0.887	0.901	0.914	0.917
Arbre de décision	Précision	94.79%	94.79%	95.01%	95.01%	95.01%	95.08%	95.08%	94.87%
	F-measure	0.948	0.948	0.95	0.95	0.95	0.951	0.951	0.948
SMO	SV	397	397	586	631	538	455	422	378
	Précision	97.18%	97.18%	97.69%	97.25%	97.54%	97.40%	97.54%	96.96%
	F-measure	0.972	0.972	0.977	0.972	0.975	0.974	0.975	0.969

A partir de la table 5.4, nous remarquons que :

— **Dans le cas de *Naïf Bayesienne*** : "Chi-square" et "Information gain" ont atteint un taux de précision égal à 92,48% qui est meilleur par rapport à notre méthode qui a obtenu une précision dans l'intervalle[84.45%,91,47%]. Néanmoins, la précision retournée par notre proposition est meilleure que celle avant la réduction (Voir table 5.2 et table 5.4).

— **Dans le cas de *l'arbre de décision*** : Notre approche donne une meilleure précision ([94.87%-95.08%]) par rapport les deux autres méthodes (94.79%). Si nous prenons l'exemple où le seuil (Ta) est égale à 0.00047, c-à-d le nombre des termes a diminué jusqu'à 463, alors notre approche nous a donné une précision égale à 94.87%. Par contre, les deux autres méthodes avec un nombre des termes égale à 1012 ont obtenu une précision de (94.79%).

— **Dans le cas de *SVM*** : Nous trouvons qu'après la réduction des caractéristiques par notre proposition nous arrivons à de meilleures précisions

([97.69%-97.18%]) que celles obtenues par "Chi-square" et Information Gain (97.18%).

Selon les tables (5.2, 5.4) : Nous pouvons confirmer que la réduction des caractéristiques en utilisant HITS est possible et même améliore la précision de classificateur à condition que nous choisissons des bons seuils sinon il est possible de perturber un peu les performances du classificateur.

TABLE 5.5 – Résultats de classification de la base des documents course après la réduction en utilisant le modèle de pondération HITS.

		Chi-2 TH=0	Info-gain TH=0	HITS					
				8×10^{-5}	14×10^{-5}	18×10^{-5}	23×10^{-5}	28×10^{-5}	47×10^{-5}
Bayes Naïf	Précision	94.36%	94.36%	88.58%	91.68%	93.42%	94.65%	95.44%	95.44%
	F-measure	0.945	0.945	0.892	0.92	0.935	0.947	0.955	0.955
Arbre de décision	Précision	95.08%	95.08%	95.08%	95.08%	95.16%	95.16%	95.16%	95.16%
	F-measure	0.95	0.95	0.95	0.95	0.951	0.951	0.951	0.951
SMO	SV	824	824	762	716	751	870	878	724
	Précision	98.26%	98.26%	97.98%	97.98%	97.90%	98.05%	98.26%	97.90%
	F-measure	0.982	0.982	0.979	0.979	0.979	0.98	0.982	0.979

La table 5.5 présente les précisions de classification obtenues en remplaçant le TFIDF par HITS (vecteur autorité) et en utilisant HITS pour la sélection des caractéristiques.

Donc, nous remarquons que :

— **Dans le cas de *Bayes Naïf*** : Notre approche améliore beaucoup le taux de réussite de la classification (Il peut aller jusqu'à 95.44%) par rapport aux "Chi-square" et "Information gain" qui ont arrivé à un taux de (94.36%).
— **Dans le cas de l'*arbre de décision*** : La même remarque(comme le cas de Bayes) peut être signalée sauf que cette fois-ci le taux de reconnaissance augmente tout en minimisant le nombre des termes. Par exemple, avec un nombre de caractéristiques égale à (2160) la précision est égale à (95.08%) tandis que avec avec un nombre de caractéristiques égale à (463) la précision est égale à (95.16%).
— **Dans le cas de *SVM*** : Nous sommes arrivés au même taux que de "Chi-square" et "Information gain" (98.26%) mais avec un ensemble plus réduit de caractéristiques (784 termes).

La table 5.6 décrit les résultats obtenus après l'application de notre proposition sur l'ensemble des pages Web *student* (voir la table 5.1).

A partir de la table 5.6, nous trouvons que :

TABLE 5.6 – Résultats de classification de la base des documents student après la réduction en utilisant le modèle de pondération TFIDF.

		Chi-2 TH=0	Info-gain TH=0	HITS					
				8×10^{-5}	14×10^{-5}	18×10^{-5}	23×10^{-5}	28×10^{-5}	47×10^{-5}
Pages Web		2782	2782	2785	2785	2784	2784	2781	2780
Termes		1314	1314	2160	1399	1127	913	784	463
Bayes Naïf	Précision	87.56%	87.56%	85.76%	85.18%	84.24%	83.95%	83.22%	82.00%
	F-measure	0.876	0.876	0.858	0.853	0.844	0.841	0.834	0.822
Arbre de Décision	Précision	90.24%	90.24%	90.02%	89.44%	87.64%	88.58%	88.00%	87.35%
	F-measure	0.902	0.902	0.9	0.895	0.876	0.885	0.88	0.873
SMO	SV	841	841	1055	984	744	951	802	724
	Précision	92.48%	92.48%	92.55%	92.77%	92.19%	92.26%	91.61%	91.40%
	F-measure	0.925	0.925	0.925	0.927	0.922	0.922	0.916	0.914

— ***Avec Bayes*** : La sélection des caractéristiques avec "Chi-square" ou "Information gain" améliore la précision de classificateur (87.56 %) par rapport à notre proposition qui donne une précision (Par exemple. 85.76%) moins que celle avant la réduction (86.19%).
— **Avec l'*arbre décisionnel*** : Notre approche garde la même précision (90.02%) du classificateur sans réduction (90.02 %) mais elle est moins que celle obtenue par "Chi-square" et "Information gain" (90.24%).
— **Dans le cas d'utilisation de *SVM*** : Notre approche peut aboutir à des résultats meilleurs que ceux qui sont obtenus par "Chi-square" et "Information gain" à condition que nous définissons le meilleur seuil.

TABLE 5.7 – Résultats de classification de la base des documents student après la réduction en utilisant le modèle de pondération HITS.

		Chi-2 TH=0	Info-gain TH=0	HITS					
				8×10^{-5}	14×10^{-5}	18×10^{-5}	23×10^{-5}	28×10^{-5}	47×10^{-5}
Bayes Baïf	Précision	86.55%	86.55%	83.01%	80.48%	79.97%	79.32%	80.33%	79.18%
	F-measure	0.865	0.865	0.828	0.801	0.796	0.791	0.801	0.791
Arbre de Décision	Précision	90.89%	90.89%	88.43%	89.37%	89.01%	88.50%	86.62%	85.76%
	F-measure	0.909	0.909	0.884	0.894	0.89	0.885	0.866	0.857
SMO	SV	1141	1141	1391	1342	1264	1240	1204	1147
	Précision	94.50%	94.50%	94.79%	94.79%	94.29%	94.22%	94.29%	93.71%
	F-measure	0.945	0.945	0.948	0.948	0.943	0.942	0.943	0.937

La table 5.7 décrire les résultats obtenus après l'application du modèle HITS pour calculer le poids des caractéristiques des pages Web de l'ensemble *student*.

D'après la table 5.7, nous remarquons que :

— **dans le cas de *Bayes Naïf*** : La précision de la classification avant la réduction (86.19%) a diminué (83.01%) après la réduction avec notre approche, par contre les deux techniques la précision ont augmenté (86.55%) en utilisant les deux techniques "Chi-square" et "Information gain".
— **dans le cas de *arbre décisionnel*** : La précision de la classification avant la réduction (90.02%) a diminué à (89.37%) après la réduction avec notre approche. Les deux techniques "Chi-square" et "Information gain" ont amélioré la précision (90.89%).
— **dans le cas de *SVM*** : Notre approche a amélioré le taux de classification (94.79%) par rapport au classificateur sans réduction (94.65%) et même par rapport au classificateur qui utilise "Chi-square" ou "Information gain" pour la sélection des caractéristiques (94.50%).

TABLE 5.8 – Résultats de classification de la base des documents Faculty après la réduction en utilisant le modèle de pondération TFIDF.

		Chi-2 TH=0	Info-gain TH=0	HITS					
				8×10^{-5}	14×10^{-5}	18×10^{-5}	23×10^{-5}	28×10^{-5}	47×10^{-5}
Pages Web		2782	2782	2785	2785	2784	2784	2781	2780
Termes		714	714	2160	1399	1127	913	784	463
Bayes Naïf	Précision	77.51%	77.51%	74.98%	75.05%	76.43%	78.96%	77.66%	72.89%
	F-measure	0.785	0.785	0.762	0.763	0.776	0.799	0.786	0.743
Arbre de Décision	Précision	89.88%	89.88%	89.44%	89.95%	89.88%	89.23%	89.23%	88.58%
	F-measure	0.896	0.896	0.894	0.897	0.89 7	0.891	0.891	0.885
SMO	SV	746	746	1170	1205	1201	1052	1192	1095
	Précision	92.84%	92.84%	92.84%	92.84%	93.13%	91.97%	92.91%	93.06%
	F-measure	0.928	0.928	0.928	0.926	0.93	0.92	0.928	0.929

La table 5.8 résume les résultats que nous avons obtenu après classification des pages Web de la base *faculty* (voir la table 5.1), dont le modèle de calcul des poids des caractéristiques était TFIDF. Donc, nous trouvons que :

— **Pour le classificateur basé sur *Bayes Naïf*** : Notre approche atteint une précision (78.96%) meilleure que celle obtenue avant la réduction (76.50%) ou que ceux obtenues par les classificateurs utilisant les techniques de réduction des caractéristiques "Chi-square" et "Information gain" (77.51%).
— **Pour le classificateur basé sur*l'arbre de décision*** : La précision du classificateur utilisant notre approche (89.95%) est la meilleure par rapport aux autres classificateurs (sans réduction (89.30%), utilisant "Chi-square" ou "Information gain"(89.88%)).
— **Pour le classificateur basé sur*SVM*** : Notre approche et avec un ensemble

très réduit des caractéristiques (463 termes) est arrivée à améliorer les performances du classificateur, de (92.99%) sans réduction à (93.06%) après la réduction ; par rapport aux autres qui utilisent "Chi-square" ou "Information gain" (92.84%).

TABLE 5.9 – Résultats de classification de la base des documents Faculty après la réduction en utilisant le modèle de pondération HITS.

		Chi-2 TH=0	Info-gain TH=0	HITS					
				8×10^{-5}	14×10^{-5}	18×10^{-5}	23×10^{-5}	28×10^{-5}	47×10^{-5}
Bayes Baîf	Précision	82.79%	82.79%	78.16%	80.33%	80.84%	81.34%	79.18%	80.33%
	F-measure	0.835	0.835	0.791	0.809	0.813	0.819	0.8	0.813
Arbre de Décision	Précision	89.15%	89.15%	88.86%	90.31%	90.31%	89.88%	89.73%	89.59%
	F-measure	0.891	0.891	0.888	0.902	0.902	0.898	0.896	0.895
SMO	SV	1146	1146	1137	1141	1098	1111	1091	1099
	Précision	94.50%	94.50%	94.14%	94.29%	94.29%	94.29%	94.14%	93.93%
	F-measure	0.945	0.945	0.94	0.942	0.942	0.942	0.94	0.938

Selon les résultats qui sont résumés dans la table 5.9, nous remarquons que :

— **Dans le cas de l'*arbre décisionnel*** : Notre approche améliore la précision de la classification par rapport à celle de classificateur sans réduction, ou dans le cas de la sélection des termes en utilisant "Chi-square" ou "Information gain".
— **Dans le cas de *Bayes*** : Par contre notre approche n'améliore pas beaucoup la précision de classification comme la sélection par "Chi-square" (ou "Information gain") la fait.
— **Dans le cas de *SVM*** : Malheureusement, la sélection de caractéristiques dans cet exemple, soit en utilisant notre approche ou "Chi-square" ou "Information gain" altère légèrement la précision de la classification.

La table 5.10 résume les résultats de la classification de l'ensemble reuters-21578 R8 par différents algorithmes d'apprentissage dont nous avons pondéré les termes en utilisant TFIDF. D'après ces résultats, nous remarquons que :

— **Dans le cas de *SVM*** : Notre approche donne une précision de classification [98.63%-98.26%] meilleure que celle obtenue par "Chi-square" ou "Information gain" (98.31%). Prenons à titre d'exemple le cas où le seuil est égal à 0.00047 alors après la réduction, il ne reste que 478 termes, mais le classificateur en utilisant SVM présente un taux de (98.26%). Ce dernier est meilleur que celui du classificateur sans réduction (98.04%) et inférieur légèrement par rapport au classificateur avec "Chi-square" et "Information gain" (98.32%) avec 1528 termes.
— **Dans le cas de *Bayes*** : Notre approche n'a pas pu améliorer la précision

TABLE 5.10 – Résultats de classification de la base des documents Reuters R8 après la réduction en utilisant le modèle de pondération TFIDF.

		Chi-2 TH=0	Info-gain TH=0	HITS					
				7×10^{-5}	9×10^{-5}	11×10^{-5}	14×10^{-5}	25×10^{-5}	47×10^{-5}
Pages Web		5485	5485	5485	5485	5485	5485	5485	5485
Termes		1528	1528	1819	1585	1399	1195	800	478
Bayes Naïf	Précision	96.48%	96.48%	95.61%	95.57%	94.84%	94.88%	95.11%	93.97%
	F-measure	0.965	0.965	0.956	0.956	0.949	0.949	0.952	0.94
Arbre de Décision	Précision	95.11%	95.11%	94.88%	94.88%	94.79%	94.88%	94.88%	94.66%
	F-measure	0.951	0.951	0.948	0.948	0.947	0.948	0.948	0.946
SMO	SV	892	892	969	913	894	869	685	642
	Précision	98.31%	98.31%	98.63%	98.49%	98.58%	98.54%	98.49%	98.26%
	F-measure	0.983	0.983	0.986	0.985	0.986	0.985	0.985	0.983

comme elle la fait dans les méthodes "Chi-square" et "Information gain".

— **Avec *l'arbre de décision*** : Notre proposition garde le même taux du classificateur sans réduction (94.88%). Par contre les deux autres méthodes améliorent un peu ce taux (95.11%).

TABLE 5.11 – Résultats de classification de la base des documents Reuters R8 après la réduction en utilisant le modèle de pondération HITS.

		Chi-2 TH=0	Info-gain TH=0	HITS					
				7×10^{-5}	9×10^{-5}	11×10^{-5}	14×10^{-5}	25×10^{-5}	47×10^{-5}
Bayes Baïf	Précision	94.97%	94.97%	94.70%	94.79%	94.97%	95.34%	95.20%	94.79%
	F-measure	0.95	0.95	0.947	0.948	0.95	0.954	0.952	0.948
Arbre de Décision	Précision	95.16%	95.16%	94.84%	94.84%	94.88%	94.88%	94.88%	95.61%
	F-measure	0.951	0.951	0.948	0.948	0.948	0.948	0.948	0.956
SMO	SV	852	852	985	1096	1106	841	773	726
	Précision	98.08%	98.08%	98.17%	98.13%	97.99%	97.85%	97.76%	97.03%
	F-measure	0.981	0.981	0.982	0.981	0.98	0.978	0.978	0.97

Selon le table 5.11, la réduction de nombre des caractéristiques en utilisant notre approche avec un schéma de pondération basé sur HITS améliore les performances des classificateurs utilisant n'importe quel algorithme d'apprentissage (SVM, arbre de décision et Naïf Bayesien) :

— **Dans le cas de *SVM*** : Notre approche a amélioré la précision du classificateur sans réduction (98.04%) et elle est arrivée à une précision égale à (98.13%) meilleure que celle de "Chi-square" et "Information gain" (98.08%).

— **Concernant *l'arbre de décision*** : Notre approche a réussi à arriver à un

taux de reconnaissance (95.61%) plus grand que celui du classificateur sans réduction (94.84%) et à celui qui a été produit par le classificateur utilisant les méthodes "Chi-square" et "Information gain" (95.16%).

— **Avec *Bayes*** : Notre approche est arrivée, dans tous les tests que nous avons conduit sur cette base des documents, à des taux de précision [94.70%- 95 34%] plus grand que la précision de classificateur sans réduction (94.20%), et même meilleur que celui de "Chi-square" et "Information gain" (94.97%).

TABLE 5.12 – Résultats de classification de la base des documents Reuters R52 après la réduction en utilisant le modèle de pondération TFIDF

		Chi-2 TH=0	Info-gain TH=0	HITS					
				6×10^{-5}	8×10^{-5}	0×10^{-5}	12×10^{-5}	25×10^{-5}	47×10^{-5}
Pages Web		6531	6531	6531	6531	6531	6531	6531	6531
Termes		2087	2087	1999	1694	1574	1310	765	452
Bayes Naïf	Précision	85.36%	85.36%	84.97%	85.43%	85.90%	87.07%	89.41%	92.37%
	F-measure	0.854	0.854	0.85	0.855	0.86	0.871	0.895	0.924
Arbre de Décision	Précision	97.00%	97.00%	96.85%	96.85%	96.85%	96.81%	96.96%	96.96%
	F-measure	0.97	0.97	0.968	0.969	0.969	0.968	0.97	0.97
SMO	SV	870	870	835	791	731	722	666	769
	Précision	98.95%	98.95%	99.18%	99.18%	99.14%	99.18%	99.14%	99.06%
	F-measure	0.989	0.989	0.992	0.992	0.991	0.992	0.991	0.991

La dernière base de documents que nous avons testé était Reuters-21578 R52. Les résultats obtenus sont résumés dans la table 5.12, dans le cas d'utilisation de TFIDF comme un schéma de pondération et dans la table 5.13 pour le schéma de pondération HITS. Donc, nous remarquons que :

— **Avec *Naïve Bayesienne*** : Notre approche améliore beaucoup la précision de la classification avec un écart de (7.48%) par rapport à la classification sans réduction, et avec un écart de (7.01%) par rapport à la sélection en utilisant "Chi-square" et "Information gain".

— **Avec *SVM*** : Notre approche améliore le taux de classification (99.18%) tandis que le taux était (98.95%) avant la réduction ou pour la classification en utilisant "Chi-square" et "Information gain" pour la sélection des caractéristiques.

— **Avec l'*arbre de décision*** : Nous sommes arrivées à des taux [96.85%,96.96%] qui sont mieux que ceux obtenus par la classification sans réduction (96.81%), mais un peu inférieurs par rapport à la classification en utilisant "Chi-square" et "Information gain" dont le taux égal à (97.00%).

En analysant les résultats contenus dans la table 5.4, nous trouvons que notre approche produit des classificateurs plus précis que ceux sans réduction en utilisant

différents algorithmes d'apprentissage

TABLE 5.13 – Résultats de classification de la base des documents Reuters R52 après la réduction en utilisant le modèle de pondération HITS.

		Chi-2 TH=0	Info-gain TH=0	HITS					
				6×10^{-5}	8×10^{-5}	0×10^{-5}	12×10^{-5}	25×10^{-5}	47×10^{-5}
Bayes Baîf	Précision	90.42%	90.42%	89.95%	90.11%	90.26%	90.62%	92.56%	93.59%
	F-measure	0.905	0.905	0.9	0.901	0.902	0.906	0.926	0.936
Arbre de Décision	Précision	96.92%	96.92%	96.92%	96.88%	96.88%	96.73%	96.81%	96.88%
	F-measure	0.969	0.969	0.969	0.969	0.969	0.967	0.968	0.969
SMO	SV	1032	1032	1057	985	1011	867	789	916
	Précision	98.79%	98.79%	98.75%	98.68%	98.68%	98.68%	98.56%	98.48%
	F-measure	0.988	0.988	0.988	0.987	0.987	0.987	0.986	0.985

A partir de la table 5.13, nous remarquons que si nous utilisons :

— ***Bayes Naïf*** : La classification de l'ensemble Reuters-21578 R52, avec notre approche et en utilisant HITS comme modèle de pondération, a été améliorée par rapport à la classification sans réduction (93.59% Vs 89.76%) ou même avec la réduction en utilisant "Chi-square" ou "Information gain".

— ***L'arbre décisionnel*** : Notre approche a réussi à améliorer les taux de précision des classificateurs comme elle ont fait les autres méthodes de sélection. L'avantage est que pour plusieurs cas de tests, l'amélioration de notre approche est meilleure par rapport à celle de "Chi-square" et "Information gain".

— ***SVM*** : La réduction de la dimension de vecteur des caractéristiques a influencé négativement sur la précision de la classification, c-à-d elle a diminué légèrement le taux de reconnaissance, soit en utilisant notre approche soit en utilisant les deux autres méthodes.

5.5 Complexité des étapes proposées

Dans cette thèse, nous avons proposé l'amélioration des performances des classificateurs des pages Web basés sur SVM en ajoutant deux étapes supplémentaires par rapport à l'approche standard. Ces étapes sont : l'utilisation de l'algorithme HITS dans la sélection et la pondération des caractéristiques contenues dans les pages Web.

Le problèmes qui se pose avec la classification des pages Web est la taille volumineuse de vecteur des caractéristiques à cause de nombre important des termes qui sont choisis d'être des caractéristiques discriminantes. Les vecteurs descriptifs, de chaque document à part, sont caractérisés par la dispersion des termes (words

sparsing), c-à-d la plupart des éléments de chaque vecteur sont mis à zéro (0). Cet caractéristique va minimiser le nombre des arcs dans graphe.

Le deuxième avantage est l'utilisation d'un graphe biparti, c-à-d qu'il n'existe pas des arcs entre les nœuds du même groupe (hubs ou autorité). Cette caractéristique rend l'interaction entre les nœuds plus faible,

Alors, ces deux avantages vont diminuer le temps nécessaire pour l'exécution de l'algorithme HITS, dont la complexité de chaque itération est estimé $O(M + N)$, où M est le nombre de caractéristiques et N est le nombre de documents. Dans les différentes expériences que nous avons effectué avec ($\xi = 10^{-7}$), le nombre d'itérations ne dépasse pas 4.

De plus, nous avons observé au cours de la comparaison entre notre méthode et les techniques de gain d'information et "Chi-square" que notre méthode réduit considérablement le temps de traitement de la sélection des caractéristiques. La table 5.14 illustre le temps consommé par chaque méthode de sélection de caractéristiques contenues dans les différentes bases des pages Web (voir table 5.1). La complexité de l'étape de réduction des caractéristiques est estimée $O(M)$.

TABLE 5.14 – Comparaison du temps consommé par notre approche et les méthode "Chi-square" et "Information gain" pour la sélection de caractéristiques des pages Web.

	Chi-2	IG	HITS
courses	10 sec	10 sec	252 msec
student	13 sec	13 sec	297 msec
faculty	12 sec	11 sec	229 msec
R8	43 sec	43 sec	291 msec
R52	59 sec	59 sec	310 msec

Enfin, nous confirmons que le temps nécessaire pour l'exécution des étapes que nous avons proposé de les ajouter (l'algorithme HITS et la sélection des caractéristiques) est négligeable par rapport à la performance globale du système, et que ces étapes accélèrent considérablement le processus de classification des pages Web.

5.6 Conclusion

Dans ce chapitre, nous avons exposé les différentes expérimentations que nous avons effectué afin de valider notre approche qui propose d'explorer et exploiter la relation qui existe entre le document et son contenu. Cette relation peut être représentée par un graphe qui va être analysé par la suite.

Nous pouvons conclure que le vecteur des autorités qui est un des deux sorties

de l'algorithme HITS peut être utilisé pour calculer les poids des caractéristiques contenues dans les pages Web. Aussi, ce vecteur peut être utilisé avec succès comme un paramètre de sélection des caractéristiques.

Néanmoins, cette approche présente un inconvénient qui est la dépendance avec la valeur du seuil c-à-d si nous arrivons à choisir le meilleur seuil nous pouvons améliorer la précision du classificateur sinon cette précision va être dégradée légèrement.

Chapitre 6

Conclusion générale et perspectives

L'exploration du Web est un domaine de recherche qui s'intéresse à l'analyse des données du Web. Le Web Mining est défini comme étant l'application des techniques de Data mining sur les données Web. Selon les données étudiées, le Web mining s'est répertorié en trois classes, qui sont : L'exploration de contenu de Web, l'exploration de la structure de Web et l'exploration de l'usage du Web.

La classification des pages Web est le domaine d'exploration de contenu de Web, l'objectif de La classification (ou la catégorisation) des pages Web est l'attribution automatique de catégories prédéfinies aux documents Web.

Le grand problème correspondant à la classification de pages Web est la grande dimensionnalité de l'espace de caractéristique, qui se compose des termes (mots ou phrases) qui se trouvent dans les documents Web, qui peuvent être des dizaines ou des centaines de milliers pour une collection de pages Web de taille moyenne. Ceci présente un grand défi pour plusieurs algorithmes d'apprentissage, il est alors, hautement demandé de trouver des mécanismes de réduction automatique de l'espace des caractéristiques sans perturber la précision de catégorisation.

La réduction de dimension de documents Web est un domaine de recherche qui a donné lieu à plusieurs études et à de nouvelles approches. Le but de la réduction de la dimensionnalité est de diminuer le nombre de caractéristiques sans dégrader la performance du système. Une approche efficace pour la réduction de la dimension est la sélection des caractéristiques (FS) qui élimine les caractéristiques non pertinentes pour sélectionner un bon sous-ensemble de caractéristiques d'origine. Un point fort la sélection des caractéristiques est que l'interprétation des caractéristiques importantes dans l'ensemble original n'est pas altérée dans le processus de réduction de dimensionnalité.

Dans cette thèse avons proposé deux contributions afin d'améliorer les performances des classificateurs des pages Web, dans le côté de l'accélération du processus de construction de classificateur en sélectionnant les meilleures caractéristiques et en éliminant les caractéristiques non pertinentes et redondantes. Dans l'autre côté, nous visons à améliorer la précision des classificateurs.

La première contribution développée dans cette thèse traite le problème de sélection de caractéristiques, de type filtres. D'abord, nous avons proposé de représenter le corpus des pages Web sous forme d'un graphe biparti, afin d'appliquer l'algorithme HITS pour trier les termes et les pages Web selon leurs importance dans le corpus. Dans l'étape qui suit, nous avons proposé de trier les caractéristiques (termes) selon leur importance dans le corpus. La notion d'importance est présentée par les valeurs d'autorité qui est l'une des deux sorties de l'algorithme HITS.

Les poids des caractéristiques contenues dans les pages Web, dans le modèle vectoriel, sont généralement calculés en utilisant le modèle TFIDF (qui est originalement inspirée du domaine recherche d'information). Notre deuxième contribution, consiste à remplacer le modèle TFIDF par un autre schéma qui se base sur l'algorithme HITS, pour lequel nous avons utilisé le vecteur des autorités comme un paramètre dans la formule calculant ces poids.

Nous avons réalisé de nombreuses expérimentations pour évaluer l'approche proposée par un les SVMs et l'algorithme Bayésien Naïf et l'arbre décisionnel en utilisant 05 ensembles de pages Web. Les résultats obtenus lors de la comparaison entre notre approche et les deux autres méthodes Chi-square et Information gain ont démontré que notre approche est performante en terme de sélection des sous-ensembles de caractéristiques et avec un temps d'exécution plus réduit.

La comparaison de notre modèle de pondération avec le modèle TFIDF a mis en évidence que celui-ci rivalise très bien avec les modèles de référence du point de vue du taux de classification, F-mesure et même minimise le nombre des vecteurs supports en cas d'utilisation des machines à vecteurs de support (SVM) comme un algorithme d'apprentissage et de classification.

Cette étude confirme que la sélection de caractéristiques permet effectivement de réduire les données inutiles et les bruits pour améliorer le taux de classification et même accélère le temps d'apprentissage de classificateur et le temps de classification d'un nouvel exemple.

Perspectives de recherche

Les travaux réalisés au cours de cette thèse nous ont permis de conclure que le problème de la classification des pages Web est très prometteur, surtout la phase de réduction de dimensions et la sélection de caractéristiques des pages Web.

De ce fait, nos perspectives de recherche porteront, essentiellement, sur la proposition des techniques complémentaires permettant d'améliorer nos résultats sur la sélection de caractéristiques pour la classification de pages Web, nous pouvons citer par exemple l'ajout des nouveaux paramètres comme nombre d'occurrences,

information sur les catégories ...etc.

Comme nous avons vu dans les chapitres précédents le seuil Th, est un paramètre important dans nos contributions de sélection des caractéristiques est spécifié manuellement. Alors, dans le future nous visons à trouver des solutions afin de minimiser l'intervention humaine pour que ce taux soit spécifier d'une manière automatique. Pour cela, nous allons essayer de trouver des formules permettant de calculer ce taux, ou faire appel à l'optimisation multi-objectifs comme par exemple les algorithmes génétiques, NSGA II, les abeilles,...etc, pour trouver tous les bons paramètres du système (Paramètres des SVM et Seuil Th).

En premier lieu, nous avons appliqué nos contributions sur un classificateur binaire des pages Web. Après les résultats encourageants que nous avons obtenu, nous envisageons aussi à étendre ce classificateur pour être applicable dans la classification multi-classes des pages Web.

Publications et Communications

Communications nationales

1. Meadi Mohamed Nadjib, Djeffal Abdelhamid, Babahenini Mohamed Chaouki. *Accélération des SVMs binaires par sélection des vecteurs supports potentiels.* Journées d'études de Laboratoire D'Informatique Oran (JDLIO'2011), Université d'Oran, Algérie.2011.

Communications internationales

1. MEADI Mohamed Nadjib et BABAHENINI Mohamed Chaouki,*Réduction des caractéristiques des pages Web basée sur l'algorithme HITS*, The First International Symposium on Informatics and its application (ISIA'14) Université de Msila, Algérie, 2014.
2. MEADI Mohamed Nadjib, BABAHENINI Mohamed Chaouki et TALEB AHMED Abdelmalik, *Accelerating the web page classifier : An approach using HITS algorithm*, les 21èmes Rencontres de la Société Francophone de Classification, CNRST-Rabat Maroc, 2014.
3. MEADI Mohamed Nadjib et BABAHENINI Mohamed Chaouki,*Accelerating the web page classifier using The HITS algorithm*, The 15th International Arab Conference on Information Technology (ACIT'2014), Université de Nizwa, Oman, 2014.

Publications

1. Mohamed Nadjib MEADI, Mohamed Chaouki BABAHENINI, Abdelmalik TALEB AHMED, *New use of the HITS algorithm for fast Web page classification*, TURKISH JOURNAL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCES,2016

Bibliographie

[Aggarwal 2014] Charu C Aggarwal. Data classification : algorithms and applications. CRC Press, 2014.

[Aggarwal 2015] Charu C Aggarwal. Data mining : the textbook. Springer, 2015.

[Aha 1996] David W Aha et Richard L Bankert. *A comparative evaluation of sequential feature selection algorithms.* In Learning from Data, pages 199–206. Springer, 1996.

[Alpaydin 2014] Ethem Alpaydin. Introduction to machine learning. MIT press, 2014.

[Androutsopoulos 2000] Ion Androutsopoulos, John Koutsias, Konstantinos V Chandrinos, George Paliouras et Constantine D Spyropoulos. *An evaluation of naive bayesian anti-spam filtering.* arXiv preprint cs/0006013, 2000.

[Bellia 2008] Heddadji Zoulikha Bellia. *Modélisation et classification de textes. Application aux plaintes liées à des situations de pollution de l ?air intérieur.* PhD thesis, Thèse de doctorat, Université de Paris DESCARTES, 2008.

[Berners-Lee 1994] Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Nielsen Henrik Frystyk et Secret Arthur. *The world-wide web.* Communications of the ACM, vol. 37, pages 76–82, 1994.

[Berners-Lee 2000] Tim Berners-Lee, Mark Fischetti et Michael L Foreword By-Dertouzos. Weaving the web : The original design and ultimate destiny of the world wide web by its inventor. HarperInformation, 2000.

[Borodin 2005] Allan Borodin, Gareth O Roberts, Jeffrey S Rosenthal et Panayiotis Tsaparas. *Link analysis ranking : algorithms, theory, and experiments.* ACM Transactions on Internet Technology (TOIT), vol. 5, no. 1, pages 231–297, 2005.

[Bouramoul 2011] Abdelkrim Bouramoul. *Recherche d'information contextuelle et sémantique.* PhD thesis, Université de Constantine, 2011.

[Brank 2002] Janez Brank, Marko Grobelnik, Natasa Milic-Frayling et Dunja Mladenic. *Feature selection using support vector machines.* WIT Transactions on Information and Communication Technologies, vol. 28, 2002.

[Brin 1997] S Brin et L Page. *Pagerank : Bringing order to the web.* Technical report, Stanford Digital Library Project, 1997.

[Burges 1998] Christopher JC Burges. *A tutorial on support vector machines for pattern recognition.* Data mining and knowledge discovery, vol. 2, no. 2, pages 121–167, 1998.

[Carreras 2001] Xavier Carreras et Lluis Marquez. *Boosting trees for anti-spam email filtering.* arXiv preprint cs/0109015, 2001.

[Ceri 2013] Stefano Ceri, Alessandro Bozzon, Marco Brambilla, Emanuele Della Valle, Piero Fraternali et Silvia Quarteroni. Web information retrieval. Springer Science & Business Media, 2013.

[Chakrabarti 1998] Soumen Chakrabarti, Byron Dom et Piotr Indyk. *Enhanced hypertext categorization using hyperlinks.* In ACM SIGMOD Record, volume 27, pages 307–318. ACM, 1998.

[Chakrabarti 1999] Soumen Chakrabarti, Martin Van den Berg et Byron Dom. *Focused crawling : a new approach to topic-specific Web resource discovery.* Computer Networks, vol. 31, no. 11, pages 1623–1640, 1999.

[Chakrabarti 2002] Soumen Chakrabarti. Mining the web : Discovering knowledge from hypertext data. Elsevier, 2002.

[Chakrabarti 2008] Soumen Chakrabarti, Earl Cox, Eibe Frank, Ralf Hartmut Güting, Jiawei Han, Xia Jiang, Micheline Kamber, Sam S Lightstone, Thomas P Nadeau, Richard E Neapolitan*et al.* Data mining : know it all. Morgan Kaufmann, 2008.

[Chang 2013] Billy Chang, Uwe Krüger, Rafal Kustra et Junping Zhang. *Canonical Correlation Analysis based on Hilbert-Schmidt Independence Criterion and Centered Kernel Target Alignment.* In ICML (2), pages 316–324, 2013.

[Chekuri 1997] Chandra Chekuri, Michael H Goldwasser, Prabhakar Raghavan et Eli Upfal. *Web search using automatic classification.* In Proceedings of the Sixth International Conference on the World Wide Web, 1997.

[Chen 2000] Hao Chen et Susan Dumais. *Bringing order to the web : automatically categorizing search results.* In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pages 145–152. ACM, 2000.

[Chen 2009] Chih-Ming Chen, Hahn-Ming Lee et Yu-Jung Chang. *Two novel feature selection approaches for web page classification.* Expert systems with Applications, vol. 36, no. 1, pages 260–272, 2009.

[Chikhi 2010] Nacim Fateh Chikhi. *Calcul de centralité et identification de structures de communautés dans les graphes de documents.* PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2010.

[Choi 2005] Ben Choi et Zhongmei Yao. *Web page classification.* pages 221–274, 2005.

[Chouaib 2011] Hassan Chouaib. *Sélection de caractéristiques : méthodes et applications.* PhD thesis, Université Paris Descartes, 2011.

[Christopher 2008] D Manning Christopher, Raghavan Prabhakanr et Schutze Hinrich. Introduction to information retrieval. Cambridge University Press, 2008.

[Cios 2007] KJ Cios, W Pedrycz, RW Swiniarski et L Kurgan. *Data mining : A knowledge discovery approach*, 2007.

[Cooley 1997] Robert Cooley, Bamshad Mobasher et Jaideep Srivastava. *Web mining : Information and pattern discovery on the world wide web.* In Tools

with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on, pages 558–567. IEEE, 1997.

[Craven 1998] Mark Craven, Andrew McCallum, Dan PiPasquo, Tom Mitchell et Dayne Freitag. *Learning to extract symbolic knowledge from the World Wide Web.* Rapport technique, DTIC Document, 1998.

[Cristianini 2000] Nello Cristianini et John Shawe-Taylor. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press, 2000.

[Dasgupta 2007] Anirban Dasgupta, Petros Drineas, Boulos Harb, Vanja Josifovski et Michael W Mahoney. *Feature selection methods for text classification.* In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 230–239. ACM, 2007.

[Dash 1997] Manoranjan Dash et Huan Liu. *Feature selection for classification.* Intelligent data analysis, vol. 1, no. 3, pages 131–156, 1997.

[Deguchi 2014] Tsuyoshi Deguchi, Katsuhide Takahashi, Hideki Takayasu et Misako Takayasu. *Hubs and authorities in the world trade network using a Weighted HITS algorithm.* PloS one, vol. 9, no. 7, page e100338, 2014.

[Desikan 2002] Prasanna Desikan, Jaideep Srivastava, Vipin Kumar et Pang-Ning Tan. *Hyperlink Analysis–Techniques & Applications.* Army High Performance Computing Center Technical Report, 2002.

[Diamantini 2015] Claudia Diamantini, Alberto Gemelli et Domenico Potena. A geometric approach to feature ranking based upon results of effective decision boundary feature matrix, pages 45–69. Springer, 2015.

[Dijkstra 1971] Edsger Wybe Dijkstra. A short introduction to the art of programming, volume 4. Technische Hogeschool Eindhoven Eindhoven, 1971.

[Djoukoué 2014] Marius Kwémou Djoukoué. *Réduction de dimension en régression logistique, application aux données Actu-Palu.* PhD thesis, Université d ?Evry Val d ?Essonne, 2014.

[Drucker 1999] Harris Drucker, Donghui Wu et Vladimir N Vapnik. *Support vector machines for spam categorization.* IEEE Transactions on Neural networks, vol. 10, no. 5, pages 1048–1054, 1999.

[Easley 2010] David Easley et Jon Kleinberg. Networks, crowds, and markets : Reasoning about a highly connected world. Cambridge University Press, 2010.

[Etzioni 1996] Oren Etzioni. *The World-Wide Web : quagmire or gold mine ?* Communications of the ACM, vol. 39, no. 11, pages 65–68, 1996.

[Fayyad 1996] Usama Fayyad, Gregory Piatetsky-Shapiro et Padhraic Smyth. *From data mining to knowledge discovery in databases.* AI magazine, vol. 17, no. 3, page 37, 1996.

[Feng 2015] LiZhou Feng, WanLi Zuo et YouWei Wang. *Improved Comprehensive Measurement Feature Selection Method for Text Categorization.* In Network and Information Systems for Computers (ICNISC), 2015 International Conference on, pages 125–128. IEEE, 2015.

[Fisher 1936] Ronald A Fisher. *The use of multiple measurements in taxonomic problems.* Annals of eugenics, vol. 7, no. 2, pages 179–188, 1936.

[Fodor 2002] Imola K Fodor. *A survey of dimension reduction techniques*, 2002.

[Getoor 2005] Lise Getoor et Christopher P Diehl. *Link mining : a survey.* ACM SIGKDD Explorations Newsletter, vol. 7, no. 2, pages 3–12, 2005.

[Grefenstette 1994] Gregory Grefenstette. *Explorations in Automatic Thesaurus Discovery, volume 278 of Kluwer International Series in Engineering and Computer Science*, 1994.

[Gu 2011] Quanquan Gu, Zhenhui Li et Jiawei Han. *Linear discriminant dimensionality reduction.* In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 549–564. Springer, 2011.

[Gu 2012] Quanquan Gu, Zhenhui Li et Jiawei Han. *Generalized fisher score for feature selection.* arXiv preprint arXiv :1202.3725, 2012.

[Gudivada 1997] Venkat N Gudivada, Vijay V Raghavan, William I Grosky et Rajesh Kasanagottu. *Information retrieval on the world wide web.* IEEE Internet Computing, vol. 1, no. 5, page 58, 1997.

[Guyon 2003] Isabelle Guyon et André Elisseeff. *An introduction to variable and feature selection.* Journal of machine learning research, vol. 3, no. Mar, pages 1157–1182, 2003.

[Hall 1997] Mark A Hall et Lloyd A Smith. *Feature subset selection : a correlation based filter approach.* 1997.

[Hall 1999] Mark A Hall. *Correlation-based feature selection for machine learning.* PhD thesis, The University of Waikato, 1999.

[Han 2011] Jiawei Han, Jian Pei et Micheline Kamber. Data mining : concepts and techniques. Elsevier, 2011.

[Haveliwala 2002] Taher H Haveliwala. *Topic-sensitive pagerank.* In Proceedings of the 11th international conference on World Wide Web, pages 517–526. ACM, 2002.

[Hotelling 1936] Harold Hotelling. *Relations between two sets of variates.* Biometrika, vol. 28, no. 3/4, pages 321–377, 1936.

[Joachims 1998] Thorsten Joachims. *Text categorization with support vector machines : Learning with many relevant features.* In European conference on machine learning, pages 137–142. Springer, 1998.

[John 1994] George H John, Ron Kohavi, Karl Pfleger*et al. Irrelevant features and the subset selection problem.* In Machine learning : proceedings of the eleventh international conference, pages 121–129, 1994.

[Johnson 2012] Faustina Johnson et Santosh Kumar Gupta. *Web content mining techniques : a survey.* International Journal of Computer Applications, vol. 47, no. 11, 2012.

[Jolliffe 2002] Ian Jolliffe. Principal component analysis. Wiley Online Library, 2002.

[Käki 2005] Mika Käki. *Findex : search result categories help users when document ranking fails.* In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 131–140. ACM, 2005.

[Kantardzic 2011] Mehmed Kantardzic. Data mining : concepts, models, methods, and algorithms. John Wiley & Sons, 2011.

[Katz 2003] Boris Katz, Jimmy J Lin, Daniel Loreto, Wesley Hildebrandt, Matthew W Bilotti, Sue Felshin, Aaron Fernandes, Gregory Marton et Federico Mora. *Integrating Web-based and Corpus-based Techniques for Question Answering.* In TREC, pages 426–435, 2003.

[Kharroubi 2002] Jamal Kharroubi. *Etude de techniques de classement" Machines à vecteurs supports" pour la vérification automatique du locuteur.* PhD thesis, Télécom ParisTech, 2002.

[Khoder 2013] Jihan Khoder. *Nouvel algorithme pour la réduction de la dimensionnalité en imagerie hyperspectrale.* PhD thesis, Université de Versailles-Saint Quentin en Yvelines, 2013.

[Kim 2004] Yongdai Kim et Jinseog Kim. *Gradient LASSO for feature selection.* In Proceedings of the twenty-first international conference on Machine learning, page 60. ACM, 2004.

[Kim 2005] Hyunsoo Kim, Peg Howland et Haesun Park. *Dimension reduction in text classification with support vector machines.* Journal of Machine Learning Research, vol. 6, no. Jan, pages 37–53, 2005.

[Kira 1992] Kenji Kira et Larry A Rendell. *The feature selection problem : Traditional methods and a new algorithm.* In AAAI, volume 2, pages 129–134, 1992.

[Kleinberg 1999] Jon M Kleinberg. *Authoritative sources in a hyperlinked environment.* Journal of the ACM (JACM), vol. 46, no. 5, pages 604–632, 1999.

[Kohavi 1997] Ron Kohavi et George H John. *Wrappers for feature subset selection.* Artificial intelligence, vol. 97, no. 1, pages 273–324, 1997.

[Kohlschütter 2007] Christian Kohlschütter, Paul-Alexandru Chirita et Wolfgang Nejdl. *Utility analysis for topically biased PageRank.* In Proceedings of the 16th international conference on World Wide Web, pages 1211–1212. ACM, 2007.

[Kononenko 1994] Igor Kononenko. *Estimating attributes : analysis and extensions of RELIEF.* In European conference on machine learning, pages 171–182. Springer, 1994.

[Kosala 2000] Raymond Kosala et Hendrik Blockeel. *Web mining research : A survey.* ACM Sigkdd Explorations Newsletter, vol. 2, no. 1, pages 1–15, 2000.

[Kuhn 1951] H.W. Kuhn et A.W. Tucker. *Nonlinear programming.* In J. Neyman, editeur, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, pages 481–492. University of California Press, Berkeley, California, 1951.

[Kwok 2001] Cody Kwok, Oren Etzioni et Daniel S Weld. *Scaling question answering to the web.* ACM Transactions on Information Systems (TOIS), vol. 19, no. 3, pages 242–262, 2001.

[Ladha 2011] L Ladha et T Deepa. *Feature selection methods and algorithms.* International journal on computer science and engineering, vol. 1, no. 3, pages 1787–1797, 2011.

[Langville 2005] Amy N Langville et Carl D Meyer. *A survey of eigenvector methods for web information retrieval.* SIAM review, vol. 47, no. 1, pages 135–161, 2005.

[Langville 2011] Amy N Langville et Carl D Meyer. Google's pagerank and beyond : The science of search engine rankings. Princeton University Press, 2011.

[Lee 2001] Hahn-Ming Lee, Chih-Ming Chen et Chia-Chen Tan. *An intelligent web-page classifier with fair feature-subset selection.* In IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th, volume 1, pages 395–400. IEEE, 2001.

[Liu 2007] Bing Liu. Web data mining : exploring hyperlinks, contents, and usage data. Springer Science & Business Media, 2007.

[Markov 2007] Zdravko Markov et Daniel T Larose. Data mining the web : uncovering patterns in web content, structure, and usage. John Wiley & Sons, 2007.

[Maxwell 2014] K Tamsin Maxwell. *Term Selection in Information Retrieval.* PhD thesis, University of Edinburgh, 2014.

[Mercer 1909] James Mercer. *Functions of positive and negative type, and their connection with the theory of integral equations.* Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character, vol. 209, pages 415–446, 1909.

[Messick 1954] Samuel J Messick et Robert P Abelson. *The additive constant problem in multidimensional scaling.* ETS Research Bulletin Series, vol. 1954, no. 1, pages 1–25, 1954.

[Miner 2012] Gary Miner. Practical text mining and statistical analysis for non-structured text data applications. Academic Press, 2012.

[Mladenić 2006] Dunja Mladenić. *Feature selection for dimensionality reduction.* In Subspace, latent structure and feature selection, pages 84–102. Springer, 2006.

[Molina 2002] Luis Carlos Molina, Lluís Belanche et Àngela Nebot. *Feature selection algorithms : a survey and experimental evaluation.* In Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on, pages 306–313. IEEE, 2002.

[Nadri 2016] Ali Asghar Nadri, Farhad Rad et Hamid Parvin. *A Framework for Categorize Feature Selection Algorithms for Classification and Clustering.* Bulletin de la Société Royale des Sciences de Liège, 2016.

[Nelson 1965] TH Nelson. *A file structure for the complex, the changing abd the indeterminate.* In i proceedings fra ACM 20th National Conference-New York, Association for Computing Machinery, 1965.

[Nettey 2006] Clive Nettey. *Link-Based Methods for Web Information Retrieval.* PhD thesis, Universiteit van Amsterdam, 2006.

[Ogura 2009] Hiroshi Ogura, Hiromi Amano et Masato Kondo. *Feature selection with a measure of deviations from Poisson in text categorization.* Expert Systems with Applications, vol. 36, no. 3, pages 6826–6832, 2009.

[Pal 2002] Sankar K Pal, Varun Talwar et Pabitra Mitra. *Web mining in soft computing framework : relevance, state of the art and future directions.* IEEE Transactions on Neural Networks, vol. 13, no. 5, pages 1163–1177, 2002.

[Pinheiro 2012] Roberto HW Pinheiro, George DC Cavalcanti, Renato F Correa et Tsang Ing Ren. *A global-ranking local feature selection method for text categorization.* Expert Systems with Applications, vol. 39, no. 17, pages 12851–12857, 2012.

[Pinheiro 2015] Roberto HW Pinheiro, George DC Cavalcanti et Tsang Ing Ren. *Data-driven global-ranking local feature selection methods for text categorization.* Expert Systems with Applications, vol. 42, no. 4, pages 1941–1949, 2015.

[Platt 1998] John C Platt. *Sequential minimal optimization : a fast algorithm for training support vector machines. 1998.* 1998.

[Porter 1980] Martin F Porter. *An algorithm for suffix stripping.* Program, vol. 14, no. 3, pages 130–137, 1980.

[Qi 2009] Xiaoguang Qi et Brian D Davison. *Web page classification : Features and algorithms.* ACM Computing Surveys (CSUR), vol. 41, no. 2, page 12, 2009.

[Robertson 1976] Stephen E Robertson et K Sparck Jones. *Relevance weighting of search terms.* Journal of the American Society for Information science, vol. 27, no. 3, pages 129–146, 1976.

[Salton 1969] Gerard Salton. *A comparison between manual and automatic indexing methods.* American Documentation, vol. 20, no. 1, pages 61–71, 1969.

[Salton 1975] Gerard Salton, Anita Wong et Chung-Shu Yang. *A vector space model for automatic indexing.* Communications of the ACM, vol. 18, no. 11, pages 613–620, 1975.

[Salton 1983] Gerard Salton, Edward A Fox et Harry Wu. *Extended Boolean information retrieval.* Communications of the ACM, vol. 26, no. 11, pages 1022–1036, 1983.

[Scholkopf 2001] Bernhard Scholkopf et Alexander J Smola. Learning with kernels : support vector machines, regularization, optimization, and beyond. MIT press, 2001.

[Shang 2007] Wenqian Shang, Houkuan Huang, Haibin Zhu, Yongmin Lin, Youli Qu et Zhihai Wang. *A novel feature selection algorithm for text categorization.* Expert Systems with Applications, vol. 33, no. 1, pages 1–5, 2007.

[Singh 2010a] Brijendra Singh et Hemant Kumar Singh. *Web data mining research : a survey.* In Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on, pages 1–10. IEEE, 2010.

[Singh 2010b] Sanasam Ranbir Singh, Hema A Murthy et Timothy A Gonsalves. *Feature Selection for Text Classification Based on Gini Coefficient of Inequality.* FSDM, vol. 10, pages 76–85, 2010.

[Stańczyk 2015] Urszula Stańczyk. Feature evaluation by filter, wrapper, and embedded approaches, pages 29–44. Springer, 2015.

[Sun 2002] Aixin Sun, Ee-Peng Lim et Wee-Keong Ng. *Web classification using support vector machine.* In Proceedings of the 4th international workshop on Web information and data management, pages 96–99. ACM, 2002.

[Tang 2014] Jiliang Tang, Salem Alelyani et Huan Liu. *Feature selection for classification : A review.* Data Classification : Algorithms and Applications, page 37, 2014.

[Tenenbaum 2000] Joshua B Tenenbaum, Vin De Silva et John C Langford. *A global geometric framework for nonlinear dimensionality reduction.* science, vol. 290, no. 5500, pages 2319–2323, 2000.

[Thelwall 2006] Mike Thelwall. *Interpreting social science link analysis research : A theoretical framework.* Journal of the American Society for Information Science and Technology, vol. 57, no. 1, pages 60–68, 2006.

[Tibshirani 1996] Robert Tibshirani. *Regression shrinkage and selection via the lasso.* Journal of the Royal Statistical Society. Series B (Methodological), pages 267–288, 1996.

[Vapnik 1982] Vladimir Naumovich Vapnik et Samuel Kotz. Estimation of dependences based on empirical data, volume 40. Springer-Verlag New York, 1982.

[Vapnik 2013] Vladimir Vapnik. The nature of statistical learning theory. Springer Science & Business Media, 2013.

[Wasserman 1994] Stanley Wasserman et Katherine Faust. Social network analysis : Methods and applications, volume 8. Cambridge university press, 1994.

[Witten 2005] Ian H Witten et Eibe Frank. Data mining : Practical machine learning tools and techniques. Morgan Kaufmann, 2005.

[Xu 2009] Xiujuan Xu, Chunguang Zhou et Zhe Wang. *Credit scoring algorithm based on link analysis ranking with support vector machine.* Expert Systems with Applications, vol. 36, no. 2, pages 2625–2632, 2009.

[Xu 2010] Guandong Xu, Yanchun Zhang et Lin Li. Web mining and social networking : techniques and applications, volume 6. Springer Science & Business Media, 2010.

[Yang 1997] Yiming Yang et Jan O Pedersen. *A comparative study on feature selection in text categorization.* In ICML, volume 97, pages 412–420, 1997.

[Yang 2004] Hui Yang et Tat-Seng Chua. *Web-based list question answering.* In Proceedings of the 20th international conference on Computational Linguistics, page 1277. Association for Computational Linguistics, 2004.

[Yang 2012] Jieming Yang, Yuanning Liu, Xiaodong Zhu, Zhen Liu et Xiaoxu Zhang. *A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization.* Information Processing & Management, vol. 48, no. 4, pages 741–754, 2012.

[Yang 2014] Jieming Yang, Zhaoyang Qu et Zhiying Liu. *Improved feature-selection method considering the imbalance problem in text categorization.* The Scientific World Journal, vol. 2014, 2014.

[Yue 2007] Xun Yue, Ajith Abraham, Zhong-Xian Chi, Yan-You Hao et Hongwei Mo. *Artificial immune system inspired behavior-based anti-spam filter.* Soft Computing, vol. 11, no. 8, pages 729–740, 2007.

[Zhu 2015] Wei-Dong Zhu, Bo Wang et Yong-Min Lin. *An Algorithm of Feature Selection in Text Categorization Based on Gini-index.* 2015.

Technique basée HITS/SVM pour la réduction et la pondération des caractéristiques des pages Web

Résumé : Le nombre de pages Web publiées sur le World Wide Web est estimé des centaines de millions. La fouille de ces pages demande un effort intellectuel incroyable qui dépasse les capacités humaines. Pour ce problème, il est conseillé d'utiliser de classificateurs automatiques qui permettent d'organiser et d'obtenir des informations de ces importantes ressources.
En général, les classificateurs automatiques de pages Web doivent gérer des millions de pages web, des dizaines de milliers de caractéristiques (des termes) et des centaines de catégories. La plupart des classificateurs utilisent le modèle vectoriel pour représenter l'ensemble des pages Web. Ce modèle produit des vecteurs des caractéristiques de taille importante, ce qui va ralentir le temps de traitement et augmenter les demandes de ressources.
Par conséquent, il y a une demande croissante pour atténuer ces problèmes en réduisant la dimension des données d'entrée sans dégrader les performances des classificateurs. La littérature compte plusieurs travaux de réduction de dimensions, mais le problème est que ces propositions importent des techniques qui consomment beaucoup de temps qui peuvent influencer sur le temps de l'apprentissage des classificateurs.
Dans cette thèse, nous proposons une approche novatrice qui améliore les classificateurs de la page Web en réduisant la dimension des données d'entrée c-à-d sélection de caractéristiques, en sélectionnant les plus importantes. Nous avons présenté l'importance d'une caractéristiques par une valeur qui s'appelle " valeur d'autorité " Cette dernière est l'une de deux sorties de l'algorithme HITS (Hypertext Induced Topic Search). Cet algorithme est très connu dans le domaine de l'analyse des liens où il est utilisé pour classer les pages Web selon leur importance dans le corpus d'entrée.
Pour valider notre approche, nous l'avons comparée avec deux algorithmes de sélection des caractéristiques, qui sont chi-square et information gain, et nous sommes arrivé à des résultats très encourageants qui confirment la possibilité d'utiliser notre proposition comme un sélecteur des caractéristiques.
Nous proposons aussi d'utiliser le vecteur des autorités pour calculer les poids des caractéristiques restantes. Nous avons évalué la précision de notre approche en la comparant au classificateur TFIDF en tant qu'un modèle de pondération et nous sommes arrivés à des résultats très compétitifs. Ces résultats confirment que notre approche peut être utilisée comme un schéma de pondération.
D'après les expérimentations que nous avons effectué sur plusieurs ensembles des pages Web, nous avons remarqué que notre approche réduit considérablement le temps nécessaire pour la classification.
Mots clés : Web mining, Classification des pages Web, Sélection des caractéristiques, Analyse des liens, HITS, SVM

Technique based HITS/SVM for reduction and weighting of Web page features

Abstract : The number of web pages published on the World Wide Web is estimated hundreds of millions. The mining of these pages requires incredible intellectual effort that exceeds human capacities. For this reason, it is recommended to use the automatic classifiers that allow organizing and obtaining information from these large resources.
Typically, automatic web pages classifiers handle millions of web pages, tens of thousands of features and hundreds of categories. Most of the classifiers use the Vector Space Model (VSM) to represent the web page dataset. This model produces vectors of large-scale dimensions that will slow down the processing time and increase resource demands.
Therefore, there is an increasing demand to alleviate these problems by reducing the size of the input data without influencing the classification results.
The literature counts several dimension reduction works, but the problem is that these proposals use a consuming-time techniques that can increase the learning time of classifiers.
In this thesis, we propose a novel approach that improves Web page classifiers accuracy by reducing the size of the input data and using a new ponderation scheme. We presented the importance of the feature by a value that called the authority value. The latter is one of HITS (Hypertext Induced Topic Search) algorithm outputs. This algorithm is well known in the field of link analysis that is used to rank web pages according to their importance in the input corpus.
To validate our approach, we compared it with two famous feature selection algorithms, which are ; Chi-square and Information gain, and we have obtained very encouraging results, which confirm the possibility of use our proposal as a features selector.
We also propose to use the vector of the authorities to calculate the weights of remaining features. We evaluate the accuracy of our approach by comparing it with classifiers that us the TFIDF as a weighting model and we have arrived at very competitive results. These results confirm that our approach can be used normally as a weighting schema.
According to the experiment that we conducted on several sets of Web pages, we noticed that our approach significantly reduces the time required for classification.
Keywords : Web mining, Web page Classification, feature selection, Link analysis, HITS, SVM.