Une approche générique pour l'analyse croisant contenu et usage des sites Web par des méthodes de bipartitionnement

Malika Charrad

'Marcheur, ce sont tes traces
ce chemin, et rien de plus ;
Marcheur, il n'y a pas de chemin,
Le chemin se construit en marchant.
En marchant se construit le chemin,
Et en regardant en arrière
On voit la sente que jamais
On ne foulera à nouveau.
Marcheur, il n'y a pas de chemin,
Seulement des sillages sur la mer.'

Antonio Machado

Remerciements

Le tout jeune docteur chassa encore quelques résidus de coquille, fit quelques pas, et fut confusément conscient de ne pas être arrivé là tout seul...

Je tiens tout d'abord à remercier mon directeur de thèse Mohamed BEN AHMED, professeur émérite à l'École Nationale des Sciences de l'Informatique, qui a guidé mes premiers pas et beaucoup des suivants sur la voie difficile de la recherche et dont les nombreuses critiques m'ont permis d'avancer dans la réalisation de ma thèse. Je le remercie vivement pour sa disponibilité, malgré son emploi de temps chargé, ses encouragements et ses précieux conseils prodigués tout au long de ce travail. Il a fait preuve d'un soutien sans faille et a toujours su exprimer sa confiance en moi, non seulement dans les moments de succès mais aussi dans les moments difficiles.

Je remercie également le professeur Gilbert SAPORTA, mon directeur de thèse au Conservatoire National des Arts et Métiers d'avoir accepté de diriger ma thèse. Je le remercie doublement car, outre son aide précieuse dans la réalisation de ce travail, il n'a pas hésité de sacrifier de son temps pour me décharger de certaines lourdeurs administratives (notamment à la fin de ma thèse).

Je tiens à remercier tout particulièrement Yves LECHEVALLIER, mon directeur de recherche à l'INRIA, qui m'a énormément aidé pendant mes quatre ans de thèse. Ses qualités scientifiques et pédagogiques m'ont guidé et m'ont toujours

permis d'aller plus loin. Je le remercie vivement pour sa disponibilité, ses conseils et pour les discussions que l'on a pu avoir et qui se sont toujours révélées très intéressantes et instructives.

Je souhaite également adresser mes remerciements à Gérard GOVAERT et Sadok BEN YAHIA d'avoir accepté d'être les rapporteurs de ma thèse et de l'intérêt qu'ils avaient porté à mon travail.
Je remercie aussi Elisabeth METAIS et Mohamed NADIF de m'avoir fait l'honneur d'être membres du jury de ma thèse.

Mes remerciements vont aussi à tous mes amis et collègues, en particulier à Siwar CHABIR et Alzennyr Da SILVA qui m'ont encouragé, soutenu et remotivé pendant ces années de travail et à tous les membres du projet Axis de l'INRIA pour leur accueil et leur amabilité.

J'exprime ma gratitude et ma profonde reconnaissance à tous les membres de ma famille qui ont toujours été là, même quand ça n'allait pas. Merci à mes parents, mes frères et soeurs et à mon beau-frère pour leurs encouragements et le soutien moral et financier qu'ils m'ont accordé tout au long de mes études.

Enfin, une attention toute spéciale à Mohamed pour son soutien, sa compréhension et ses encouragements.

Résumé

Dans cette thèse, nous proposons une nouvelle approche WCUM (Web Content and Usage Mining based approach) permettant de relier l'analyse du contenu à l'analyse de l'usage d'un site Web afin de mieux comprendre le comportement général des visiteurs du site. Ce travail repose sur l'utilisation de l'algorithme CROKI2 de classification croisée implémenté selon deux stratégies d'optimisation différentes que nous comparons à travers des expérimentations sur des données générées artificiellement.
Afin de pallier le problème de détermination du nombre de classes sur les lignes et les colonnes, nous proposons de généraliser certains indices proposés initialement pour évaluer les partitions obtenues par des algorithmes de classification simple, aux algorithmes de classification simultanée. Pour évaluer la performance de ces indices nous proposons un algorithme de génération de biclasses artificielles pour effectuer des simulations et valider les résultats. Des expérimentations sur des données artificielles ainsi qu'une application sur des données réelles ont été réalisées pour évaluer l'efficacité de l'approche proposée.
Mots clés : Fouille du contenu du Web, fouille de l'usage du Web, classification simultanée.

Abstract

In this thesis, we propose a new approach WCUM (Web Content and Usage Mining based approach) for linking content analysis to usage analysis of a website to better understand the general behavior of the web site visitors. This work is based on the use of the block clustering algorithm CROKI2 implemented by two different strategies of optimization that we compared through experiments on artificially generated data.
To mitigate the problem of determination of the number of clusters on lines and columns, we suggest to generalize the use of some indices originally proposed to evaluate the partitions obtained by simple classification algorithms to evaluate bipartitions obtained by simultaneous clustering algorithms. To evaluate the performance of these indices on data with biclusters structure, we proposed an algorithm for generating artificial data to perform simulations and validate the results. Experiments on artificial data as well as on real data were realized to estimate the efficiency of the proposed approach.
Key words : Web Content Mining, Web Usage Mining, simultaneous clustering.

Table des matières

Table des figures

Liste des tableaux

Introduction

0.1 Introduction au Web Mining

L'Extraction de Connaissances à partir de Données (ECD) (Knowledge Discovery in Databases en anglais) est un domaine aujourd'hui très en vogue. On la définit comme" un processus non-trivial d'identification de structures inconnues, valides et potentiellement exploitables dans les bases de données" [Fay96]. Ce processus, décrit dans la figure 1, englobe le stockage, le nettoyage et la sélection des données, l'analyse de celles-ci par différentes techniques, et enfin, l'interprétation et l'évaluation des connaissances acquises.

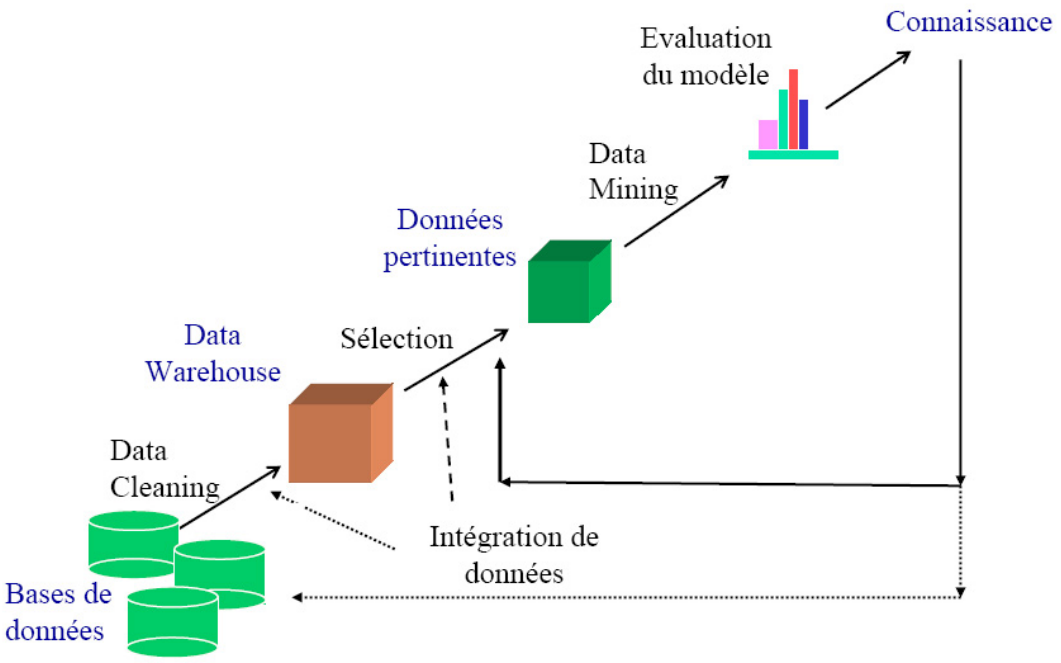


FIGURE 1 – Processus d'Extraction des Connaissances à partir des Données

Traditionnellement, la pratique de l'ECD fait appel aux techniques d'analyse

de données et aux statistiques exploratoires. De nos jours, avec l'abondance des ressources des données (entrepôts des données, archives, Web) et leur aspect polymorphe (texte, multimédia, base des données), le traitement des données sort de plus en plus des sentiers battus en traitant, non plus seulement des fichiers plats "individus x variables", mais également des données non structurées telles que les textes("text mining"), les images ("image mining"), le son ("sound mining") et la vidéo. On parle alors plus généralement du "multimédia mining".
Actuellement, une des principales sources des données est le Web. Cette source est en perpétuelle expansion due à la croissance colossale du nombre des documents en ligne d'une part et l'augmentation du nombre des utilisateurs d'autre part. En effet le nombre des utilisateurs d'Internet dans le monde a augmenté de 380.3 % au cours des neuf dernières années et a atteint plus de 1,733 millions au mois de septembre 2009 [1], ce qui correspond à un taux de pénétration (nombre d'utilisateurs d'Internet par rapport à la population) de 25.6 %. Parallèlement, plus de trois millions de sites Web sont crées mensuellement. Le nombre total des sites Web a atteint 233,6 millions au mois de novembre 2009 alors qu'il avoisinait 185 millions au mois de novembre 2008 selon l'enquête de Netcraft [2].
Face à cette densité en ligne, il est de plus en plus difficile d'exploiter efficacement ces informations. En effet, l'abondance de ces ressources, leur perpétuelle évolution et l'aspect polymorphe de leur contenu (texte, multimédia, base des données) sont autant d'écueils qu'il faut contourner. La nécessité de disposer de méthodes et d'outils avancés permettant d'y remédier a fait émerger de nouveaux domaines de recherches tels que le Web Mining (WM), défini comme l'application de techniques de data mining aux données du web pour l'extraction d'informations pertinentes et de connaissances. Il s'est développé à la fin des années 1990 afin d'analyser l'organisation et les performances des sites web et de personnaliser le contenu proposé aux internautes en tenant compte de leurs préférences et de leur profil.

0.1.1 Axes de développement du Web Mining

Les principales sources des données du Web Mining sont les pages web, les fichiers logs du serveur, les bases de données clients et les cookies qui permettent d'alimenter des data webhouses. Ces données sont classifiées en quatre types [SCDT00] :

- Données relatives au contenu : données contenues dans les pages Web (textes, graphiques),
- Données relatives à la structure : données décrivant l'organisation du contenu (structure de la page, structure inter-page),

1. http ://www.internetworldstats.com/stats.htm
2. http ://news.netcraft.com/

- Données relatives à l'usage : données sur l'usage telles que les adresses IP, la date et le temps des requêtes, fournies par les fichiers logs du serveur Web,
- Données relatives au profil de l'utilisateur : données fournissant des informations démographiques sur les utilisateurs du site Web.

En se basant sur ces types de données, les axes de développement actuels du Web sont :

- Le Web Content Mining (WCM) : consiste en une analyse textuelle avancée (traitement linguistiques, classification des pages, segmentation thématique...) intégrant les particularités du Web telles que la structure sémantique des pages.
- Le Web Usage Mining (WUM) : s'intéresse à l'analyse des comportements de navigation sur les sites Web notamment l'analyse du clickstream (l'ensemble des clics exécutés sur le site) afin de mesurer l'audience et la performance des sites Web (temps passé par page, nombre de visites, profil de l'utilisateur, horaires et fréquences des consultations,...) et d'enrichir les sources de données de l'entreprise (bases de données clients, bases marketing,...).
- Le Web Structure Mining (WSM) : consiste à analyser l'architecture des sites Web et des liens entre les différents sites afin d'améliorer leur ergonomie par la suppression ou l'ajout de nouveaux liens entre les pages.

0.1.2 Processus du Web Mining

Il y a trois phases dans le processus du web mining [Coo00] : le prétraitement (*preprocessing*), la découverte de modèles de navigation (*pattern discovery*) et l'analyse de ces modèles (*pattern analysis*). Ce processus générique (fig. 2) s'adapte à chacun des axes du Web mining selon la nature des données utilisées (texte, logs, liens...).

- Prétraitement : le prétraitement consiste à transformer les informations sur l'usage, le contenu et la structure du contenu à des abstractions de données nécessaires pour la découverte de modèles.
- Découverte des modèles : la découverte des modèles de navigation nécessite l'utilisation de plusieurs méthodes et algorithmes provenant de différentes branches de recherche telles les statistiques, le data mining et le machine learning.
- Analyse des modèles : l'analyse des modèles de navigation est la dernière étape dans tout le processus du Web Mining. Il s'agit d'analyser les résultats et d'extraire des connaissances.

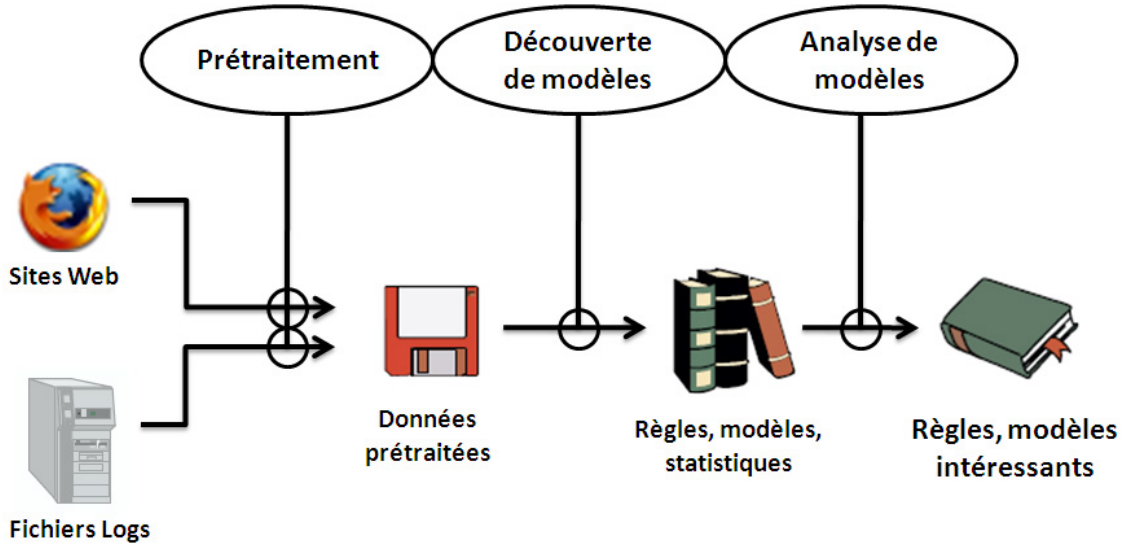


FIGURE 2 – Processus du Web Mining

0.2 Problèmes ouverts en Web Mining

Le Web, devenu un outil essentiel pour un accès facile et rapide à l'information, représente une source immense de données pour se renseigner, rechercher, apprendre et découvrir de nouvelles connaissances. Ainsi, on pense même que c'est le Web qui fût l'élément décisif du passage d'une société industrielle à une société d'information. Une société d'information est une société où la création, la distribution et la manipulation d'information est devenue une activité économique et culturelle significative. Le Web permet donc de mieux répondre au besoin toujours plus grandissant d'information et de connaissances. Mais en apportant une solution, il apporte par le fait même de nouveaux problèmes aussi bien pour les internautes que pour les concepteurs des sites web. Parmi ces problèmes :

- Trouver l'information recherchée : les internautes, à la recherche d'une information spécifique, utilisent des moteurs de recherche qui retournent un ensemble de pages assez similaires à la requête de l'utilisateur. Cependant, la performance des algorithmes utilisés par ces moteurs n'est pas assez satisfaisante à cause de la faiblesse de leur précision et leur rappel. La faiblesse de la précision est due au nombre élevé de documents non pertinents retrouvés pour une requête donnée alors que la faiblesse du rappel est due à la difficulté d'indexer toute l'information disponible sur le Web [Cha00].
- Créer des nouvelles connaissances à partir de l'information présente sur le Web : alors que le problème précédent est orienté recherche d'information, le second est orienté data mining. Il suppose la présence d'une collection

des données Web à partir desquelles de la connaissance devrait être extraite. Plusieurs travaux ont été menés dans ce sens ([MBNL99], [Coh99], [CDF+98]...) afin d'utiliser le Web comme une base des connaissances pour l'aide à la décision.

– Comprendre le comportement des consommateurs ou visiteurs des sites Web et personnaliser l'information : ce problème concerne les concepteurs et les gestionnaires des sites Web. Il consiste à identifier les visiteurs des sites, leurs préférences, leurs motifs de visite et apporter des modifications aux sites pour répondre à leurs attentes.

Ce dernier problème a fait l'objet de plusieurs travaux de recherches (tab. 1).

TABLE 1 – Travaux antérieurs sur la classification des pages et des utilisateurs

	Méthode	**Données**	**Références**
Classification des pages	- Classif. Floue	- Termes présents dans les liens(<a>tag)	[RHS09]
	- Classif. Floue	- Phrases-clés	[FKS+04]
	- Kmeans	- Mots-clés	[QS04]
	- Naïve Bayes	- Mots-clés	[TVJ06]
	- SVM et LSA	- Mots-clés	[CH06]
	- ARHP (Association Rule Hypergraph Partition)	- Fichiers Logs	[MHLN02]
	- Classification spectrale	- Topologie du site	[HZS06]
Classification des utilisateurs	- Classif. hiérarchique floue	- Fichiers Logs	[LMC03]
	- Classif. hiérarchique	- Fichiers Logs	[YP05]
	- Classif. simple	- Fichiers Logs	[PAV07]
	- Classification floue	- Fichiers Logs	[CFMT07]
	- PACT (Profile Aggregation based on Clustering Transactions)	- Fichiers Logs	[MHLN02]
Classification des utilisateurs et des pages		- Topologie du site + Fichiers logs	[PE98]
		- Fichiers logs + données sur le contenu	[SCDT00]
		- Fichiers logs	[ZCM02]
		- Fichiers logs	[LHY05]
		- Fichiers logs	[MRGG05]
		- Fichiers logs	[KV09]

[CCB02], par exemple, propose l'approche "*SurfMiner*" reposant sur l'analyse des fichiers Logs afin de découvrir les usages d'un site associés à des descriptions d'utilisateurs. Cette approche repose sur l'hypothèse qu'il existe une certaine corrélation entre les pratiques différentes des utilisateurs et leurs caractéristiques personnelles. Elle consiste à extraire des motifs fréquents de navigation des utilisateurs de référence et découvrir des relations entre les motifs découverts et des traits d'utilisateurs. [SCDT00] cherche dans les données extraites des fichiers logs deux types de classes : classes d'usagers et classes de pages. La classification des utilisateurs a pour objectif d'établir des groupes d'internautes ayant des comportements de navigation similaires. L'examen de ces groupes permet d'associer un profil à chaque classe d'utilisateurs. La classification des pages Web consultées par les internautes permet de découvrir des groupes de pages ce qui facilite la tâche des navigateurs et des robots. Dans ce cas, la classification des pages est basée sur les résultats de la classification des utilisateurs. La majorité des travaux qui associent la classification des pages à la classification des utilisateurs construisent les classes des pages à partir des classes des utilisateurs ou à partir des visites des internautes (pages visitées ensemble par plusieurs visiteurs). En d'autres termes, la classification des utilisateurs guide la classification des pages [KV09].
Dans les rares travaux où la classification des pages est effectuée en premier lieu, un algorithme de classification simple, DBSCAN dans [LHY05] par exemple, est appliqué aux données issues des fichiers logs pour découvrir des classes de pages puis les classes des utilisateurs sont construits à partir des classes des pages. Ainsi, la liaison entre l'usage et le contenu est effectuée à travers les données sur l'usage. Le contenu textuel ou multimédia des pages n'est pas utilisé dans la classification des pages.
D'autre part, afin de pallier le problème des matrices creuses, certains travaux ont associé la catégorisation des pages à l'analyse de l'usage. En effet, au cours d'une session ou une navigation, le nombre de pages visitées est très faible par rapport au nombre total des pages visitées par tous les utilisateurs. Par conséquent, le nombre de colonnes présentant des valeurs nulles est important ce qui peut réduire la qualité des résultats lors de l'application de certaines méthodes de fouille des données sur les données de l'usage.
Cependant, la majorité de ces travaux catégorise les pages en se basant soit sur les URLs soit sur les mots-clés utilisés pour indexer la page, ou encore en utilisant les données sur l'usage en supposant que des pages visitées ensemble appartiennent à la même classe. Le contenu textuel des pages n'intervient pas généralement dans de telles classifications. Il serait donc intéressant d'introduire le contenu des pages Web dans l'analyse de l'usage étant donné que le comportement des internautes sur un site dépend fortement du contenu proposé dans les pages. Ce contenu doit lui même être mis à jour et modifié selon les besoins des utilisateurs du site.

0.3 Problématique et contributions

Le présent travail s'inscrit dans le cadre de la recherche en Web Mining en ayant pour objectif de faire la liaison entre l'analyse du contenu et l'analyse de l'usage d'un site web. Il est principalement organisé autour de deux principaux axes du Web Mining à savoir le Web Content Mining qui s'intéresse à l'analyse du contenu et le Web Usage Mining qui s'intéresse à l'analyse de l'usage. Notre idée est d'exploiter les différentes informations relatives au contenu d'un site Web et de son usage en vue d'analyser le site.
Pour ce faire, nous proposons une approche WCUM (Web Content and Usage Mining based approach) permettant d'analyser le contenu d'un site Web indépendamment de son usage. Les résultats de cette analyse sont exploités par la suite pour comprendre les comportements de navigation sur le site. L'apport de ce travail réside principalement dans les points suivants :

- La première contribution consiste à proposer une approche reliant l'analyse du contenu à l'analyse de l'usage. La liaison entre ces deux axes permet d'évaluer la conception du site par la comparaison entre la structure sémantique et la structure logique et confronter cette structure logique à la perception des utilisateurs représentée par les traces de navigation enregistrées dans les fichiers logs. D'autre part, la classification des navigations sur le site permet de découvrir des modèles d'utilisation servant à mieux comprendre le comportement général des visiteurs du site. La première étape dans ce travail consiste à classifier les pages du site en pages de contenu présentant une information éventuellement utile aux utilisateurs, pages auxiliaires (ou de navigation) servant à faciliter la navigation sur le site et pages hybrides ayant les caractéristiques communes des deux classes précédentes. Cette étape a pour objectif d'exclure les pages auxiliaires de l'analyse. Un travail de prétraitement linguistique et de sélection de descripteurs est effectué sur les pages de contenu et les pages hybrides afin d'aboutir à une représentation matricielle du site.
  Un algorithme de block clustering est appliqué à la matrice pour découvrir des biclasses de pages et de descripteurs permettant d'attribuer un thème à chaque groupe de pages. L'ensemble des thèmes résultant de la classification simultanée ainsi que le fichier de sessions résultant d'une phase de prétraitement des fichiers logs sont les entrées d'un nouvel algorithme de classification appliqué à la matrice croisant les visites aux thèmes pour découvrir des patrons d'utilisation du site Web.
- La deuxième contribution consiste à étendre l'application des méthodes de block clustering, appliquées généralement en bioinformatique pour l'analyse des biopuces, au contexte Web mining afin de profiter de leur pouvoir classificatoire dans la découverte de biclasses homogènes à partir d'une parti-

tion des instances et une partition des attributs recherchées simultanément. Une biclasse est composée d'une classe sur les lignes et une classe sur les colonnes ayant une forte corrélation. L'apport de ces méthodes par rapport aux méthodes de classification automatique est qu'elles agissent d'une façon symétrique sur les deux ensembles de données (sur les lignes et les colonnes) permettant ainsi de découvrir des modèles locaux dans les données au lieu de modèles globaux obtenus par les méthodes de classification simple.
Cependant, en optant pour les méthodes de classification simultanée, nous sommes confrontés à plusieurs problèmes dont le plus important est le problème de détermination du nombre de classes sur les lignes et les colonnes. En effet, une des premières critiques que présentent les méthodes de block clustering, comme les méthodes de partitionnement simple, est la connaissance requise du nombre de classes dans la population. Différentes approches ont été proposées dans la littérature pour résoudre ce problème dans le cas de la classification simple mais aucune étude n'a été effectuée pour résoudre ce problème dans le cas de la classification simultanée.
Afin de pallier ce problème, nous avons proposé un nouvel indice reposant sur la méthode de la différentielle et adapté certains indices proposés initialement pour évaluer les partitions obtenues par des algorithmes de classification simple, aux algorithmes de classification simultanée. La pénurie (voir l'inexistence) de benchmarks de données présentant une structure en biclasses nous a incités à proposer un algorithme de génération de données artificielles pour effectuer des simulations et valider les résultats.

- La dernière contribution consiste à proposer une nouvelle variante de l'algorithme Croki2 de classification croisée des tableaux de contingence reposant sur une optimisation globale. Cette variante constitue une légère amélioration dans l'algorithme Croki2 permettant de gagner en temps de calcul. Les deux variantes sont implémentées sous R et testées sur des données artificielles et réelles.

0.4 Organisation du mémoire

Ce mémoire est composé de deux grandes parties. La première partie est composée de trois chapitres présentant un état de l'art sur le Web Content Mining, le Web Usage Mining et les algorithmes de bipartitionnement. La deuxième partie, composée également de trois chapitres, présente les contributions de cette thèse. Les différents chapitres de la thèse sont organisés comme suit :

- **Le chapitre 1** présente en détail le processus du Web Content Mining appliqué aux documents textuels. En effet, bien que le Web Content Mining s'intéressait à l'extraction des connaissances à partir des différentes

sources de données dans le Web (image, audio, vidéo, texte, hyperliens, métadonnées...) nous nous concentrons dans ce chapitre sur le processus WCM appliqué aux textes contenues dans les pages Web. Ce processus débute par un prétraitement linguistique (lemmatisation et étiquetage) suivi d'une sélection et extraction des descripteurs afin d'aboutir à une représentation terminologique du contenu textuel du site Web. Le processus se termine par l'extraction des connaissances à travers la classification ou la catégorisation.

- **Le chapitre 2** présente les différentes étapes du processus du Web Usage Mining, à savoir le prétraitement des données (nettoyage, transformation, structuration), la fouille des données et l'analyse des résultats. Il présente également un état de l'art sur les travaux existants en Web Usage Mining qui sont en liaison avec notre problématique et les enjeux majeurs y existant.
- **Le chapitre 3** présente un état de l'art sur les approches et les algorithmes de recherche de biclasses et l'apport d'une telle stratégie de classification par rapport à la classification simple. Il traite également des différences entre ces algorithmes au niveau de type de biclasses obtenues, la nature du tableau des données sur lequel ils s'appliquent et la nécessité de fixer à priori le nombre de classes sur les lignes et les colonnes.
- **Le chapitre 4** est consacré à la présentation de certaines contributions de cette thèse. Celles-ci concernent la génération automatique des données artificielles présentant une structure de biclasses ainsi qu'une nouvelle variante de l'algorithme Croki2 reposant sur l'optimisation globale. Les deux variantes de l'algorithme sont implémentés, testés sur des données générées artificiellement et comparés à travers un ensemble de critères.
- **Le chapitre 5** soulève le problème de détermination du nombre de classes dans les algorithmes de bipartitionnement. Une panoplie d'indices de validation basés sur des critères externes, relatifs ou internes, proposés initialement pour les algorithmes de classification simple sont adaptés au contexte de bipartitionnement. D'autres indices sont également proposés.
  Ces indices sont testés sur des données artificielles qui simulent la structure pages-descripteurs dans un site Web, en vue d'identifier l'indice le plus performant dans l'identification du bon nombre de classes. Ce même indice est appliqué par la suite aux données réelles afin d'estimer le nombre de classes dans les données représentant le contenu textuel d'un site Web.
- **Le chapitre 6** décrit l'approche WCUM que nous proposons pour l'analyse d'un site Web. Cette approche repose sur la combinaison entre l'analyse du contenu et l'analyse de l'usage d'un site Web.
  La méthodologie de l'analyse du contenu, basée sur le typage des pages, le block clustering et l'analyse sémantique latente, est décrite en détail ainsi que la méthodologie de l'analyse de l'usage basée sur la transformation, la

structuration et la classification des données de l'usage.
Les résultats de l'application de l'approche WCUM sur des données réelles issues d'un site web de tourisme sont présentés à la fin du chapitre ainsi que les avantages et les limites de cette approche.

– **Le chapitre 7** récapitule l'apport des travaux de recherche décrits dans ce mémoire et présente quelques pistes pouvant inspirer des travaux futurs intéressés par la même thématique.

Première partie

État de l'art

Chapitre 1

Le Web Content Mining appliqué aux documents textuels : État de l'art

"Documents are grouped because they are in some sense related to each other; but more basically, they are grouped because they are likely to be wanted together (...) "
Van Rijsbergen, 1979, p. 24

1.1 Introduction

Bien que le contenu du Web se présentait sous différentes formes : images, vidéo, audio, métadonnées ..., une bonne partie de l'information accessible est sous forme textuelle (bibliothèques électroniques, pages HTML, forums de discussion ...). Ainsi, l'analyse du contenu du Web ou le Web Content Mining (WCM) fait appel aux outils de fouille du texte, ou text mining afin de faciliter l'extraction des connaissances enfouies dans les documents web. Le text mining tel qu'il est défini dans [ISS04] est le "processus non trivial d'extraction d'informations implicites, précédemment inconnues, et potentiellement utiles, à partir de données textuelles non structurées dans de grandes collections de textes". Il représente ainsi l'opération d'analyse et de structuration de grands ensembles de documents par l'utilisation de techniques de traitement du langage naturel et des outils de fouille des données. Des exemples de ces techniques sont l'extraction d'information, la catégorisation de textes, la cartographie de textes et les modèles d'apprentissage automatique. Parmi les applications de text mining :

- La classification automatique des documents,
- Le résumé automatique des textes,
- L'alimentation automatique des bases de données,

– La veille sur des corpus documentaires importants,
– L'enrichissement de l'index d'un moteur de recherche pour améliorer la consultation des documents.

Ce chapitre se veut un tour d'horizon sur les techniques de text mining utilisées dans le processus du WCM appliqué aux documents textuels, ainsi que les techniques de catégorisation et de segmentation de documents utilisées généralement dans le processus de recherche d'information. Ainsi, le WCM appliqué aux textes est au croisement de chemins entre plusieurs domaines de recherches, tels que le text mining et la recherche d'information.

1.2 Processus du WCM appliqué aux textes

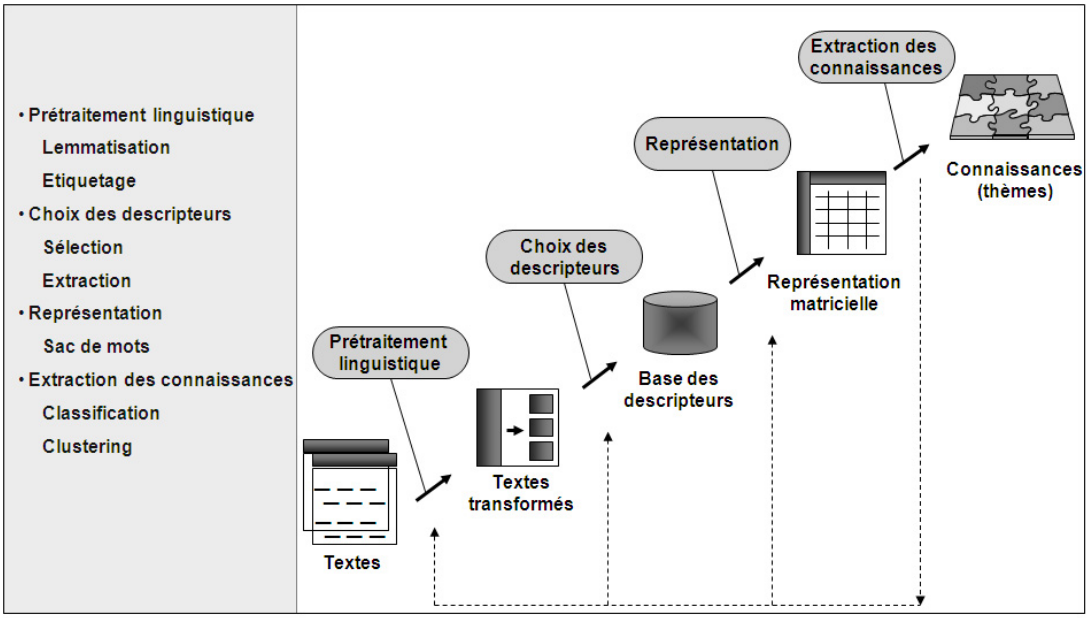


FIGURE 1.1 – Processus du Web Content Mining appliqué aux textes.

Le Web content mining a pour objectif d'extraire des connaissances à partir du contenu des pages Web. Ce contenu se présente sous différents types : texte, image, audio, vidéo, métadonnées et hyperliens. Le WCM décrit le processus d'extraction des informations à partir des différentes sources de données dans le Web. Ces sources de données sont structurées, telles que les tables et les bases des données, semi-structurées telles que les pages HTML ou non structurées telles que les textes. Le processus du WCM appliqué aux textes (fig.1.1)

comprend généralement la même succession d'étapes que tout processus d'extraction des connaissances à partir des données. En effet, la première étape est celle du prétraitement des données (nettoyage, structuration...), la deuxième est celle d'application des techniques de Data mining pour l'extraction des connaissances et la dernière est celle d'analyse et de validation. Cependant, la phase du prétraitement varie selon le type des données (textes, images, fichiers logs), de même le choix de la méthode de fouille des données varie selon l'objectif de l'analyse. Dans notre thèse, l'objectif de l'analyse textuelle est d'identifier le contenu thématique (i.e. les sujets ou les thèmes) des documents et de les regrouper en classes selon ces thèmes.

1.3 Prétraitements des documents textuels

La première étape des prétraitements linguistiques des documents textuels consiste en la lemmatisation et l'étiquetage morphosyntaxique des documents. L'étape suivante concerne l'élimination des mots vides (articles, pronoms, prépositions, etc.). À l'aide des étiquettes, généralement seuls les noms, les verbes et les adjectifs sont conservés. Plusieurs analyseurs linguistiques intégrant un étiqueteur morphosyntaxique et un lemmatiseur fonctionnant pour les documents textuels en français ont été construits ou adaptés au français. Nous citons à titre d'exemples cordial analyseur [1].

1.3.1 Lemmatisation

La lemmatisation consiste à ramener le terme à sa forme canonique conventionnelle (infinitif pour les verbes, masculin singulier pour les noms, etc.). Généralement, chaque flexion d'un mot dans le texte est considérée comme un descripteur différent, en particulier les différentes formes d'un verbe. Par exemple, les mots créer et créée sont considérés comme des descripteurs différents alors qu'il s'agit de deux formes conjuguées du même verbe qui ont *a priori* le même sens. Pour remédier à ce problème, on considère uniquement la racine des mots plutôt que les mots entiers (on parle de *stem* en anglais). Plusieurs algorithmes ont été proposés pour substituer les mots par leur racine ; l'un des plus connus pour la langue anglaise est l'algorithme de Porter [Por80].

1. Cordial analyseur, Site Web, http ://www.synapse-fr.com/Cordial-Analyseur/Presentation-Cordial-Analyseur.htm

1.3.2 Étiquetage

L'étiquetage consiste à associer à chaque occurrence de mot du corpus des informations concernant sa catégorie morphosyntaxique (nom, verbe, adjectif, etc.) et son état flexionnel (genre, nombre, temps et mode des verbes, etc.). Il facilite ainsi le choix des termes non utilisés dans l'analyse, par exemple ceux dont l'étiquette est "déterminant" ou "adverbe", etc.

1.4 Choix de descripteurs

Par définition, un terme est un mot ou une expression ayant un sens précis dans un contexte donné. [DGG+94] définit la notion de 'terme' comme représentant une unité signifiante constituée d'un mot (terme simple) ou de plusieurs mots (terme complexe), et qui désigne une notion de façon univoque à l'intérieur d'un domaine. Le choix des termes qui serviront des descripteurs aux textes est une étape incontournable étant donné le grand nombre de descripteurs potentiels. En effet, chacun des textes est représenté par un vecteur ayant autant de termes qu'il y'a de mots dans le vocabulaire. L'utilisation de tous ces mots peut influencer négativement la précision de la classification puisque plusieurs mots sont vides de sens, donc inutiles pour faire ressortir la sémantique d'un texte. D'autre part, un mot présent dans un nombre élevé de documents ne permet pas de décider de l'appartenance d'un texte qui le contient à l'une ou l'autre des catégories car son pouvoir de discrimination est faible. Des techniques de réduction de la dimension du vocabulaire ont été proposées dans la littérature. Ces techniques se divisent en deux grandes familles.

- Les techniques basées sur la sélection de descripteurs ("*feature selection*") : Ces techniques conservent seulement les descripteurs jugés utiles à la classification, selon une certaine fonction d'évaluation. Les autres sont rejetés. L'avantage de la sélection de descripteurs consiste à éliminer les descripteurs réellement inutiles ou les descripteurs erronés ("*noisy*") (mots mal orthographiés par exemple).
- Les techniques basées sur l'extraction de descripteurs ("*feature extraction*") : Ces techniques créent des nouveaux descripteurs à partir des descripteurs de départ, en faisant des regroupements ou des transformations. L'avantage de ces descripteurs est la réduction du nombre de descripteurs redondants.

1.4.1 Sélection de descripteurs

Plusieurs techniques de sélection de descripteurs ont été développées en vue de réduire la dimension de l'espace vectoriel. Chacune de ces techniques utilise des critères lui permettant de rejeter les descripteurs jugés inutiles à la tâche de

classification. Il en résulte un vocabulaire réduit, des textes représentés par des vecteurs de moindre dimension, un temps de calcul plus abordable et même dans certains cas une précision de classification accrue. Le processus de sélection de descripteurs débute généralement par la suppression de mots très fréquents tels que les mots grammaticaux ou les mots de liaisons. Les mots très rares, qui n'apparaissent qu'une ou deux fois sont également supprimés, car il n'est pas possible de construire de statistiques fiables à partir d'une ou deux occurrences seulement [Str00]. La racinisation ("stemming") et la lemmatisation peuvent être considérés comme des méthodes de réduction du vocabulaire. En effet, ils permettent de regrouper en un seul descripteur les multiples formes morphologiques de mots qui ont une sémantique commune. La substitution des mots par leur racine ou leur lemme réduit l'espace des descripteurs et permet de représenter par un même descripteur des mots qui ont le même sens. Cependant, ces substitutions peuvent dans certains cas augmenter l'ambiguïté des descripteurs en représentant par un même descripteur des mots avec des sens différents.

1.4.1.1 Recherche des mots fréquents

Selon [Str00], les mots très fréquents doivent être supprimés de la représentation des textes pour deux raisons :

- d'un point de vue linguistique, ces mots, communément appelés mots vides (ou *stop words* en anglais), ne contiennent aucune information sémantique et ne modifient pas le sens des mots qui les accompagnent tels que les prépositions.
- d'un point de vue statistique, ces mots qui se retrouvent sur l'ensemble des textes, ne sont d'aucune aide pour la classification. Leur pouvoir de discrimination est faible.

La suppression de ces mots nécessite l'utilisation d'une liste. [Sah98], par exemple, définit une liste de 570 mots courants en anglais, plus une liste de 100 mots très fréquents sur le Web. Néanmoins, l'établissement d'une telle liste peut poser des problèmes. D'une part, il n'est pas facile de déterminer le nombre de mots exacts qu'il faut inclure dans cette liste. D'autre part, cette liste est intimement liée à la langue utilisée et n'est donc pas transposable directement à une autre langue. La méthode que nous adoptons pour supprimer les mots fréquents est celle proposée par [Str00]. En effet, le rapport *R(m,t)* entre *TF(m,t)* l'occurrence d'un mot dans un texte t et *CF(m)* l'occurrence de ce mot sur le corpus permet de classer les mots par ordre décroissant.

$$R(m,t) = \frac{TF(m,t)}{CF(m)} \tag{1.1}$$

Plus le mot m est fréquent dans le corpus, plus le ratio est faible et, inversement, plus un mot est rare, plus le ratio est élevé. Dans le cas limite où un mot n'apparaît qu'une seule fois dans le corpus, ce ratio vaut 1 et le mot est classé en première place. La deuxième moitié de la liste des mots de la représentation du texte est supprimée de manière à ce que cette représentation ne contienne plus les mots fréquents du corpus.

1.4.1.2 Recherche des mots rares

En plus des mots fréquents, les auteurs cherchent également à supprimer les mots rares d'un corpus afin de réduire de façon appréciable la dimension des vecteurs utilisés pour représenter les textes. D'un point de vue linguistique, la suppression de ces mots n'est pas nécessairement justifiée. Certains mots peuvent être très rares, mais très informatifs. Néanmoins, ces mots ne peuvent pas être utilisés par des méthodes à bases d'apprentissage du fait de leur très faible occurrence. Une des méthodes communément retenues pour supprimer ces mots consiste à ne garder que les mots dont l'occurrence totale est supérieure à un seuil fixé préalablement. Dans notre cas, nous fixons ce seuil à 1 et nous supposons qu'un mot apparaissant plus qu'une fois dans le texte peut être utile à l'analyse.

1.4.1.3 Recherche des mots pertinents à la classification

La suppression de ces catégories de mots s'avère insuffisante car le nombre de candidats reste encore très élevé. Il est donc nécessaire d'utiliser une méthode statistique pour déterminer les mots utiles pour la discrimination entre les documents. Parmi les méthodes les plus souvent utilisées figurent le calcul de l'information mutuelle [Lew92] [Mou97] [DPHS98], la méthode du chi-2 [SHP95] [WPW95] ou des méthodes plus simples utilisant uniquement les fréquences d'apparition [Wie93] [YP97] ;

- L'information mutuelle ("mutual information") : cette façon d'évaluer la qualité d'un mot dans la prédiction de la classe d'un document est basée sur le nombre de fois qu'un mot apparaît dans une certaine catégorie. Plus un mot apparaît dans une catégorie, plus l'information mutuelle du mot et de la catégorie est jugée élevée. Plus un mot apparaît en dehors de la catégorie (et plus une catégorie apparaît sans le mot), moins l'information mutuelle est jugée élevée. Il faut ensuite faire une moyenne des scores du mot jumelé à chacune des catégories. La faiblesse de cette mesure est qu'elle est beaucoup trop influencée par la fréquence des mots. Pour une même probabilité conditionnelle sachant la catégorie, un terme rare va être avantagé, car il risque moins d'apparaître en dehors de la catégorie.

- La statistique du χ^2 : elle s'adapte bien à la sélection de descripteurs, puisqu'elle évalue le manque d'indépendance entre un mot et une classe. Elle utilise les mêmes notions de co-occurrence.
- La fréquence ("*document frequency*") : il s'agit d'éliminer les mots dont le nombre de documents dans lesquels ils apparaissent est en dessous d'un certain seuil. L'idée sous-jacente est que ces mots n'apportent pas d'information utile à la prédiction de la catégorie d'un texte ou qu'ils n'influencent pas la performance globale du classificateur. Il y' a aussi la possibilité que ces termes soient le résultat d'erreurs, comme un mot mal orthographié. Dans ce cas, leur élimination serait bénéfique.
- Le gain d'information ("*information gain*") : il mesure en quelque sorte le pouvoir de discrimination d'un mot, le nombre de bits d'information obtenue pour la prédiction de la catégorie en sachant la présence ou l'absence d'un mot. Cette méthode est souvent mise en pratique dans les arbres de décision, pour choisir le descripteur qui va le mieux diviser l'ensemble des instances en deux groupes homogènes.
- La force du terme ("*term strength*") : cette méthode se propose d'estimer l'importance d'un terme en fonction de sa propension à apparaître dans des documents semblables. Une première étape consiste à former des paires de documents dont la similarité cosinusoïdale est supérieure à un certain seuil. La force d'un terme est ensuite calculée à l'aide de la probabilité conditionnelle qu'il apparaisse dans le deuxième document d'une paire, sachant qu'il apparaît dans le premier.

Les résultats d'une étude comparative des cinq méthodes menée par [YP97] sur deux classificateurs différents démontrent la supériorité du gain d'information et de la statistique du χ^2 sur les autres. La sélection de descripteurs sur la base de la fréquence est avantageuse lorsque le temps de calcul importe. C'est la raison pour laquelle nous optons pour cette méthode.

1.4.2 Extraction de descripteurs

Le processus d'extraction de descripteurs consiste à créer à partir des descripteurs originaux un ensemble de descripteurs synthétiques qui maximisent l'efficacité de la classification. Le regroupement de termes ("*term clustering*") et l'analyse sémantique latente, qui sera présentée profondément dans la section 1.5.1.2, font partie de ces techniques.

- Term clustering : a pour but de grouper les mots qui ont une sémantique commune. Les groupes ("*clusters*") ainsi créés deviennent les descripteurs d'un nouvel espace vectoriel. En ce sens, [Blu67] rapporte des résultats intéressants à propos d'un tel regroupement basé sur la distribution des étiquettes de classes associées à chacun des mots. L'idée de départ est la

suivante : si deux mots différents apparaissent dans les mêmes classes et dans les mêmes proportions, les réunir en un seul descripteur qui affiche la distribution moyenne ne peut pas affecter négativement la performance de la classification.

– Latent Semantic Analysis : l'approche LSA proposée par [DDF+90], et connue sous le nom "Latent Semantic Indexing (LSI)" dans le domaine de la recherche d'information ("*Information retrieval*"), consiste à effectuer une décomposition en valeurs singulières de la matrice dont chaque colonne représente un document grâce à un vecteur des occurrences des termes qui le composent. Cette matrice est projetée dans un espace de dimensions plus faibles où les descripteurs considérés ne sont plus de simples termes. Avec cette méthode, les termes apparaissant ensemble sont projetés sur la même dimension. La représentation en utilisant LSA permet de résoudre partiellement le problème des synonymes et des termes polysémiques, d'où elle parait plus avantageuse que le term clustering.

1.4.3 Nombre de descripteurs retenus

Le nombre de descripteurs à conserver dans la liste des descripteurs retenus et ordonnés du plus important au moins important (la notion d'importance dépend de la méthode de classement considérée) dépend souvent du modèle. En effet, les machines à vecteurs supports sont, par exemple, capables de manipuler des vecteurs de grandes dimensions alors que, pour les réseaux de neurones, il est préférable de limiter la dimension des vecteurs d'entrées. Pour choisir le bon nombre de descripteurs, il faut déterminer si l'information apportée par les derniers descripteurs est utile, ou si elle est redondante avec l'information apportée par les premiers descripteurs. En considérant l'ensemble des termes du corpus Reuters, Joachims [Joa98] garde 9962 termes distincts pour représenter les textes en entrée de son modèle après suppression des mots les plus fréquents et l'utilisation de racines lexicales (les stems). Il considère que l'ensemble de ces termes apporte de l'information, et qu'il est indispensable de les inclure tous dans son modèle pour appliquer les machines à vecteurs supports. Cependant, [DPHS98], qui utilisent également les machines à vecteurs supports, ne considèrent que 300 descripteurs pour représenter les textes. Ils obtiennent néanmoins de meilleurs résultats que Joachims sur le même corpus ce qui laisse à penser que certains termes utilisés par Joachims n'étaient pas nécessaires. Dans leur étude sur la sélection de descripteurs, [YP97] considèrent que l'espace des descripteurs doit être de plus grande dimension contrairement à d'autres auteurs qui considèrent qu'un très petit nombre de descripteurs pertinents suffit pour construire un modèle performant. Dans notre cas, nous n'imposons pas de contraintes sur le nombre de descripteurs retenus d'une part parce que la méthode de classification que

nous utiliserons dans le chapitre 5 ne dépend pas de la dimension des vecteurs. D'autre part, nous considérons que le nombre de descripteurs est une résultante de l'application des techniques de réduction de la dimension.

1.5 Représentation terminologique des textes

Les deux méthodes de représentation les plus utilisées sont les ontologies [Sow00] et la représentation vetorielle. Nous nous intéressons dans ce chapitre à la seconde méthode.

1.5.1 Représentation vectorielle

1.5.1.1 Modèle de représentation vectorielle de Salton

Salton et al. [SWY75] ont proposé au milieu des années 70 un modèle de représentation vectorielle du texte qui reste à ce jour très utilisé et souvent désigné sous le nom de modèle de Salton ou de représentation de Salton. Ce modèle, appelé *Vector Space Model* (ou *term vector model*), est un modèle algébrique utilisé pour représenter un document textuel par un vecteur d'identifiants, comme par exemple des termes. Ce modèle a été utilisé en recherche d'information, en indexation de documents ou encore pour faire du filtrage d'information. Dans ce modèle, un document est représenté par un vecteur dont chacune des dimensions est un terme. Si ce terme apparaît dans le document, alors la valeur associée à cette dimension est non-nulle. À la base, les n termes sont les n différents mots apparaissant dans les textes de l'ensemble d'entraînement. Cette représentation des textes exclut toute analyse grammaticale et toute notion de distance entre les mots. C'est pourquoi elle est appelée "sac de mots" ou "*bag of words*". L'inconvénient de cette méthode de représentation est la perte d'une quantité considérable d'information contenue dans le document original vu que les paragraphes, les phrases et l'ordre des mots sont complètement écartés. Toute l'information sur les structures syntaxiques est perdue. [SM99] ont essayé une dizaine de façons de représenter les textes pour des fins de classification. Ils ont d'abord considéré les groupes nominaux, en extrayant des suites de noms et d'adjectifs pour construire les termes de l'espace vectoriel et en utilisant une application pour l'analyse de la nature grammaticale des mots du texte. Ils ont aussi évalué l'impact de regrouper les mots synonymes en un même méta-descripteur. Dans le même ordre d'idées, la notion d'hyperonymes a été mise à l'épreuve pour regrouper des mots de la même façon. Aucun de ces modes alternatifs n'a produit de résultats équivalents ou supérieurs à l'approche "*bag-of-words*". Par contre, une partie de leurs expérimentations a porté sur l'utilisation de plusieurs classificateurs utilisant des modes de représentation différents, combinés à l'aide d'un vote à majorité. Il en a

résulté une classification meilleure. [Lew92] n'a pas non plus obtenu de résultats satisfaisants en représentant les textes à l'aide de groupes nominaux. L'explication trouvée à ces résultats décevants est que malgré la qualité sémantique supérieure des groupes nominaux par rapport aux mots individuels ils possèdent une qualité statistique inférieure. En effet, un mot individuel est, en moyenne, amené à apparaître plus fréquemment dans un corpus de textes qu'un groupe nominal, fournissant ainsi au classificateur une meilleure image de la distribution statistique caractérisant le corpus.

Fonctions de pondération des termes

Plusieurs fonctions de pondérations ont été proposées dans la littérature. Ainsi, le poids W associé à chaque terme peut être :

- Une fonction binaire indiquant la présence ou l'absence du terme t dans le document d. On parle dans ce cas d'un modèle vectoriel binaire.

$$W(d,t) = B(d,t) = \begin{Bmatrix} 1 & si & t & \in & d \\ 0 & si & t & \notin & d \end{Bmatrix}$$

- Une fonction positive $TF(d,t)$ représentant le nombre d'occurrences du terme t dans le document d (Term Frequency). L'importance du terme est proportionnelle à sa fréquence d'apparition dans le document [Luh57]. Ainsi, le poids associé à chaque terme est

$$W(d,t) = TF(d,t)$$

  L'inconvénient de l'utilisation de cette fonction est qu'elle accorde un poids important aux termes qui apparaissent très souvent à travers toutes les classes des documents et qui sont peu représentatifs d'une classe en particulier.
- Une fonction IDF-binaire qui repose sur l'hypothèse que l'importance d'un terme est inversement proportionnelle au nombre de documents dans lesquels il apparaît. N est le nombre total des documents dans le site web et $df(t)$ est le nombre de documents contenant le terme t.

$$IDF(t) = log(\frac{N}{df(t)})$$

  Ainsi, le poids associé à chaque terme est

$$W(d,t) = B(d,t) \times IDF(t)$$

  avec $B(d,t) = \{0,1\}$

– Une fonction $TFIDF$ (acronyme de "*Term Frequency Inverse Document Frequency*") de la fréquence $TF(d,t)$ du terme t dans le document d et de l'inverse du nombre des documents contenant le terme t, $IDF(t)$. Issue du monde de la recherche d'information [Sal88], la fonction $TFIDF$ donne un faible poids aux termes ayant un faible pouvoir de discrimination entre les classes des documents, c'est à dire les termes appartenant à plusieurs documents.

1.5.1.2 Analyse Sémantique Latente (ASL)

Développée par [LD97a], la méthode ASL est utilisée pour représenter les données textuelles dans le but de regrouper les documents thématiquement proches. Elle consiste à construire un espace sémantique de très grande dimension à partir de l'analyse statistique de l'ensemble des co-occurrences dans un corpus de textes. Elle repose en effet sur la décomposition en valeurs singulières, une propriété des matrices rectangulaires, proposée par Eckart et Young en 1936, qui est à la base des méthodes factorielles d'analyse de données [LMP00]. La méthode ASL repose sur l'hypothèse que des mots qui apparaissent dans un même contexte sont sémantiquement proches. Le corpus est représenté sous forme matricielle. Les lignes sont relatives aux mots et les colonnes représentent les différents contextes choisis (un document, un paragraphe, une phrase, etc.). Chaque cellule de la matrice représente le nombre d'occurrences des mots dans chacun des contextes du corpus. Deux mots proches au niveau sémantique sont représentés par des vecteurs proches. La mesure de proximité est généralement définie par le cosinus de l'angle entre les deux vecteurs. Pour dériver d'un tableau lexical les relations sémantiques entre les mots, la simple analyse des co-occurrences brutes se heurte à un problème majeur. En effet, même dans un grand corpus de textes, la plus grande partie des mots sont relativement rares. Il s'ensuit que les co-occurrences le sont encore plus. Leur rareté les rend particulièrement sensibles à des variations aléatoires [BLL98] [RB97]. L'ASL résout ce problème en remplaçant le tableau de fréquences original par une approximation qui produit une sorte de lissage des associations. Une décomposition en valeurs singulières est appliquée au tableau de fréquences avant d'être recomposé à partir d'une fraction seulement de l'information qu'il contient. Les milliers de mots caractérisant les documents sont ainsi remplacés par des combinaisons linéaires ou 'dimensions sémantiques' sur lesquelles peuvent être situés les mots originaux. Contrairement à une analyse factorielle classique, les dimensions extraites sont très nombreuses (plusieurs centaines) et non interprétables. Les mots et les documents originaux sont positionnés dans le nouvel espace sémantique, ce qui permet de mesurer leur proximité par le cosinus entre les vecteurs qui les représentent.

Etapes d'une analyse sémantique latente

Les différentes étapes nécessaires pour dériver un espace sémantique d'un tableau lexical sont illustrées dans la figure 1.2 .

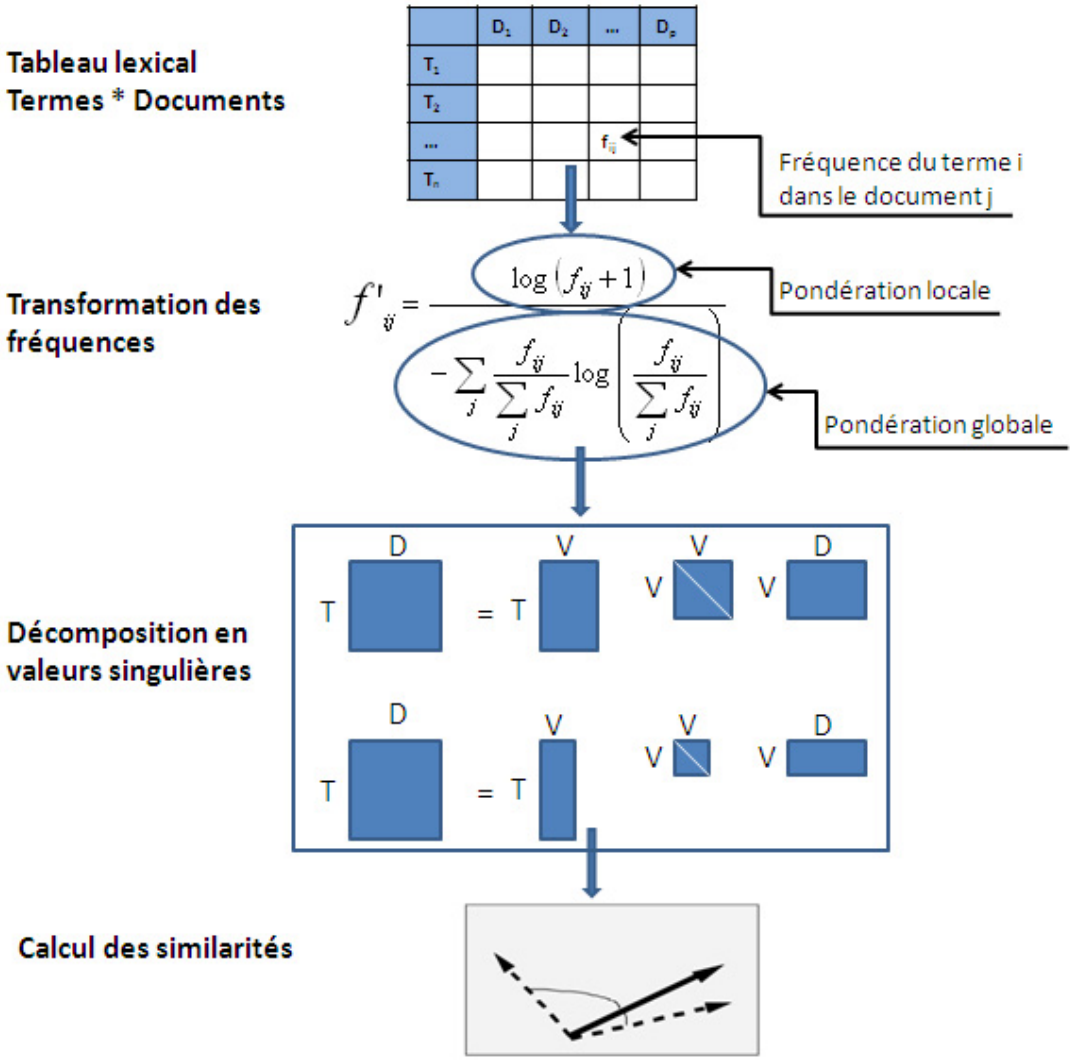


FIGURE 1.2 – Etapes de l'analyse sémantique latente

Le point de départ dans une analyse sémantique latente est un tableau lexical croisant les termes aux documents. Chaque cellule du tableau correspond au nombre d'occurrences du terme i dans le document j. La première étape est la transformation des fréquences afin de privilégier les termes les plus informatifs. Les nouvelles fréquences sont calculés à partir d'une combinaison d'une pondération locale qui réduit l'impact des mots très fréquents dans un document et

d'une pondération globale qui réduit l'impact des mots "peu informatifs", c'est-à-dire les mots qui apparaissent avec une fréquence constante dans les documents. L'étape suivante est celle de la décomposition en valeurs singulières. Cette étape permet de compresser l'information par la sélection des k dimensions orthogonales les plus importantes ($100 \leq k \leq 300$). La dernière étape est celle de calcul de similarités entre deux termes ou deux documents. Cette similarité est mesurée par le cosinus entre les vecteurs correspondants.

Caractéristiques théoriques de l'ASL

La théorie sur laquelle s'appuie l'ASL est la décomposition en valeurs singulières (SVD). Une matrice $A = [a_{ij}]$ où a_{ij} est la fréquence d'apparition du mot i dans le contexte j, se décompose en un produit de trois matrices USV^T. U et V sont des matrices orthogonales et S une matrice diagonale. Soit S_k où $k < r$ la matrice produite en enlevant de S les r-k colonnes qui correspondent aux plus petites valeurs singulières. Soient U_k et V_k les matrices obtenues en enlevant les colonnes correspondantes des matrices U et V. La matrice $U_k S_k V_k^T$ peut alors être considérée comme une version compressée de la matrice originale A.

Avantages et limites de l'ASL

Le principal intérêt de l'ASL est la notion d'indépendance par rapport à la langue du corpus étudié et le fait de se dispenser de connaissances linguistiques et de celles du domaine. Ainsi, elle permet la construction automatique de connaissances sémantiques génériques (domain-independent knowledge). Ces connaissances peuvent être employées quel que soit le domaine dont sont issus les textes pour développer des techniques d'analyse du langage comme celles employées en indexation automatique de documents ou en segmentation des textes [CWHM01] [DB03] [KWH02]. Bien que cette approche soit prometteuse, il n'en demeure pas moins que son utilisation soulève des contraintes. Notons par exemple l'importance de la taille des contextes choisis. En effet, [RSW+98] ont montré lors de leurs expérimentations que si les contextes possèdent moins de 60 mots, les résultats s'avèrent être décevants.[RC06] a également montré que l'efficacité de l'ASL est fortement influencée par la proximité du vocabulaire utilisé.

1.6 Extraction des connaissances pour la catégorisation

Une fois les textes représentés, des techniques de fouille des données sont appliqués aux matrices des données croisant les documents aux descripteurs. Dans le but d'attribuer un thème ou une étiquette à chaque document en fonction des descripteurs retenus, la classification, la catégorisation et la segmentation sont

les méthodes souvent utilisées pour l'extraction des connaissances. Cependant une confusion entre ces termes persiste à ce jour [Réh05]. Plusieurs définitions, souvent complémentaires, parfois contradictoires, y sont données.

1.6.1 Étymologie et définitions

La première définition de la classification date de 1787 [Fér87]. Le terme apparaît pour la première fois dans la cinquième édition du dictionnaire de l'Académie Française en 1798 [DAF98] sous la définition : "*distribution en classes et suivant un certain ordre*". En Analyse des Données, on distingue la détermination de classes à partir de données (classification) et le rangement d'individus dans des classes déjà constituées (classement). Le vocabulaire employé dans d'autres disciplines est parfois le même avec un autre sens. Ainsi, classification peut avoir le sens de rangement dans des classes, ce que d'autres appellent aussi identification. Parfois le vocabulaire est différent, mais le sens est le même. Ainsi la classification s'appelle taxonomie en biologie, typologie en sciences humaines, segmentation en marketing... La classification est donc, en quelque sorte, le travail préliminaire au classement, à savoir la recherche de classes "naturelles" dans le domaine étudié [Rou85]. Pour éviter toute confusion, la classification est dite supervisée ou non supervisée selon qu'on dispose ou non d'une information *a priori* sur la structure des données. Ainsi, le classement correspond à la classification supervisée. A chaque observation est associée une classe *a priori* et l'objectif de la classification est de respecter au mieux ces classes *a priori*. On parle aussi d'un problème de discrimination ou d'apprentissage avec professeur [Lec04]. Dans le cas où aucune information n'est disponible sur les classes et l'objectif de la classification est de regrouper les individus dans des classes homogènes en fonction de l'ensemble de variables sélectionnées, on parle d'un problème de classification automatique ou d'apprentissage sans professeur. Le terme "catégorisation" n'existe pas dans le dictionnaire de l'Académie Française, contrairement au mot "catégorie" d'origine grecque et du bas latin *katêgoria* (ou *katègoriai* : "qualité attribuée à un objet"). La catégorisation peut être définie comme étant "l'action de créer des catégories ou le résultat de cette action" [Nak07]. Aristote définit les catégories comme étant les "espèces les plus générales de ce qui est signifié par un mot simple". Mais [Réh05] rappelle que dans la littérature scientifique, les termes "classification" et "catégorisation" sont indifféremment utilisés. La segmentation est définie comme étant l'action de segmenter c.à.d Diviser ou partager en segments [DAF32]. Dans d'autres dictionnaires, c'est l'opération de découper une population hétérogène en sous-ensembles plus homogènes disposant de caractéristiques communes. Dans ce cas, la segmentation est étroitement liée à la classification non supervisée ou le clustering.

1.6.2 Classification, Catégorisation et Segmentation des documents

Dans le contexte de l'analyse textuelle, le même problème se pose. En effet, dans les travaux de la littérature, les définitions de ces termes présentent une certaine ambigüité et diffèrent des définitions précédemment évoquées.

1.6.3 Classification des documents

Dans [JMF99] et [MS99], la classification des documents est une opération non supervisée visant à regrouper un ensemble de documents sur la base d'un ou de plusieurs critères de similarité. Ainsi, cette tâche de regroupement correspond à l'opération de clustering ou classification non supervisée. [BYR99] soulignent que "(Clustering is) the grouping of documents which satisfies a set of common properties. The aim is to assemble together documents which are related among themselves".

1.6.4 Catégorisation des documents

[Jal03] définit la catégorisation de textes comme étant la recherche d'une relation bijective qui consiste à "chercher une liaison fonctionnelle entre un ensemble de textes et un ensemble de catégories (étiquettes, classes)". [Seb05] ajoute la notion de classes cibles prédéfinies et voit la catégorisation de textes comme étant une tâche de tri. Dans [Seb05] et [MS99], la catégorisation des documents est définie comme un processus d'organisation supervisé dans le cadre duquel une ou plusieurs catégories thématiques (il s'agit donc de critères de similarité externes aux documents) sont attribuées à chacun des documents. En d'autres termes, cette tâche réside dans la projection d'une taxinomie (c'est-à-dire un ensemble de catégories structurées) sur des documents afin d'attribuer à chaque document une ou plusieurs étiquettes thématiques représentant le contenu de chacun de ces documents [For06]. Dans [Seb02] l'auteur associe étroitement la catégorisation automatique à l'identification des thèmes, " text categorization (text classification, or topic spotting), the activity of labelling natural language texts with thematic categories from a predefined set, is one such task ". D'ailleurs, nombreux sont les auteurs pour lesquels la catégorisation automatique est synonyme de l'identification thématique comme en témoigne l'extrait suivant : " The goal in text categorization is to classify the topic or theme of a document. " [MS99]. Dans son application, le processus de catégorisation présuppose l'élaboration a priori d'une taxinomie ou d'une hiérarchie de catégories thématiques adaptées au contenu et aux spécificités des documents à catégoriser. Ce processus nécessite aussi l'identification, à partir de l'ensemble des documents, des caractéristiques

(linguistiques et statistiques) qui serviront de base à la catégorisation et le choix d'une méthode d'attribution des catégories aux différents documents. À cet égard, plusieurs méthodes ont été explorées dans la littérature. Les plus fréquemment employées sont les réseaux de neurones artificiels [Str00], les arbres de décision tel que l'algorithme C4.5 [Qui93], les machines à vecteurs supports [BWS03] [Joa02] et les modèles de Markov cachés [AZG00]. Le processus de catégorisation peut être formalisé par une fonction d'appariement [Seb02] $\Phi : D \times C \rightarrow \{0, 1\}$ où C est l'ensemble de catégories prédéfinies et D est un ensemble de documents. La catégorisation consiste à attribuer une seule catégorie ou plusieurs catégories par document.

1.6.5 Segmentation des documents

Dans la littérature, la tâche de segmentation des textes est traitée sous plusieurs angles selon la finalité de l'utilisateur : reconnaissance du texte ou bien extraction et recherche d'information. Elle consiste à segmenter le texte en unités (mots, phrases, paragraphes, etc). On distingue alors différents types de segmentation [Oue01].

- Segmentation en une suite de mots,
- Segmentation en phrases,
- Segmentation en paragraphes,
- Segmentation thématique.

Les méthodes de segmentation thématique procèdent à l'identification des différents thèmes véhiculés par le texte, pour le segmenter en unités homogènes formant des blocs thématiques. Dans [VP07], la segmentation de textes est définie comme étant une tâche de reconnaissance thématique qui peut s'apparenter à une forme d'indexation. L'idée est de dégager des parties de textes offrant une certaine cohérence, et de les distinguer les unes des autres soit en les nommant soit en en délimitant les contours. Dans ce dernier cas, la segmentation de texte est assimilable à la détection des ruptures thématiques. Certaines méthodes de segmentation thématique sont supervisées d'autres sont non supervisées. Parmi les méthodes supervisées, [DA97] proposent de construire un modèle probabiliste exponentiel qui à chaque phrase fait correspondre la probabilité qu'il y ait une frontière entre cette phrase et la phrase suivante. La distribution de probabilité est choisie en construisant de façon incrémentale un modèle log-linéaire. Dans [BMEBS98], un certain nombre de thématiques sont apprises sur une base d'apprentissage. Les auteurs emploient alors un modèle de langage pour étiqueter chaque paragraphe d'un texte suivant sa thématique de plus forte probabilité. Lorsque la valeur de la probabilité du meilleur thème décroît, une frontière thématique possible est détectée. La sélection définitive des frontières est effectuée suivant une méthode de programmation dynamique. [YCG+98] présentent une

méthode de segmentation basée sur les Modèles de Markov Cachés (MMC). Les méthodes de segmentation thématique non supervisées se basent principalement sur la notion de cohésion lexicale observée à travers la répétition de termes. La distinction principale entre ces méthodes réside dans le fait que certaines de ces approches, telles que celles présentées dans [Cho00], [Hea97] et [UI01] sont basées sur l'information contenue dans le texte à segmenter comme la cohésion lexicale, alors que d'autres reposent sur des connaissances sémantiques complémentaires extraites de dictionnaires et de thésaurus, comme dans [Koz93], [LNCC04] et [MH91]. Ces dernières méthodes pallient le problème d'absence des mots communs entre des phrases relevant du même thème. Parmi les méthodes non supervisées proposées pour la segmentation thématique, TextTiling [Hea97], C99 [Cho00], Dotplotting [Rey98] et Segmenter [KKM98] sont les plus connues.

1.7 Conclusion

En conclusion, dans le contexte de l'analyse textuelle, on entend par la classification des documents la classification non supervisée ou le clustering. La catégorisation des documents correspond par contre à une classification supervisée. Quand à la segmentation des documents, elle peut être supervisée ou non supervisée selon le type de la méthode utilisée. Dans notre cas, sachant qu'un site Web est généralement lié à un domaine spécifique (commerce électronique, tourisme, éducation...), nous disposons d'une information à priori sur les catégories auxquelles les pages du site pourraient appartenir. Il s'agit donc d'un problème de catégorisation des pages Web. L'intérêt de cette catégorisation est de résumer l'information présentée dans le site Web à étudier afin de faciliter par la suite l'analyse de l'usage du site et la découverte des modèles comportementaux. Le chapitre suivant sera donc consacré à la présentation d'un état de l'art sur les techniques du Web Usage Mining et du processus d'extraction de connaissances à partir des fichiers Logs, notre seconde source d'information (en plus du contenu textuel du site) pour l'analyse d'un site Web.

Chapitre 2

Le Web Usage Mining : état de l'art

2.1 Introduction

La fouille de données d'usage du Web (Web Usage Mining (WUM), en anglais) est définie comme étant l'application du processus d'Extraction des Connaissances à partir de bases de Données (ECD) aux données issues des fichiers Logs HTTP afin d'extraire des modèles comportementaux d'accès au Web en vue de répondre aux besoins des visiteurs de manière spécifique et adaptée (personnaliser les services) et faciliter la navigation [Tan05]. Comme les analyses se font à partir des fichiers logs (traces) de serveurs Web, on parle également de Web Log Mining. L'objectif de cette analyse est d'étudier le comportement de l'utilisateur dans son interaction avec le site Web. Elle est centrée sur l'ensemble de clics effectués lors d'une visite au site. On parle aussi d'analyse de clickstream. L'intérêt de cette analyse est d'enrichir les sources de données utilisateur afin d'affiner les profils utilisateurs et les modèles comportementaux. Le passage en revue des techniques de traitement des fichiers Logs ainsi que des travaux existants sur l'exploitation de l'information contenue dans les fichiers logs constituent le corps de ce chapitre.

2.2 Intérêts et applications du WUM

L'intérêt principal du *Web Usage Mining* est qu'il fournit des informations sur la manière dont les utilisateurs naviguent réellement sur le site Web. Ces informations sont exploitées pour faciliter la navigation sur le site. Dans [SCDT00], [Tan05] et [FL05] les auteurs considèrent qu'il y a cinq applications du Web Usage Mining :

- Évaluer et caractériser de manière générale l'activité d'un site Web : l'objectif étant ici de faire de l'observation (pas de la modélisation ou de l'induction) et de mettre en évidence les fonctionnalités les plus ou les moins

utilisées. Les techniques d'analyse sont souvent simples, elles relèvent en effet du dénombrement et des statistiques simples (moyennes, histogramme, indice, tri croisés).

- Améliorer les modes d'accès aux informations : le Web Usage Mining est alors utilisé pour comprendre comment les utilisateurs se servent d'un site, s'il y a des failles dans la sécurité ou des accès non autorisés identifiés.
- Modifier la structure : dans le même esprit que précédemment mais avec des techniques différentes, l'amélioration du site est effectuée par une restructuration des pages et des liens. Les pages considérées comme similaires par des techniques de classification, seront reliées de manière hypertextuelle.
- Personnaliser la consultation : il s'agit de faire des recommandations dynamiques à un utilisateur en se basant sur son profil et une base de connaissance d'usages connus [ZEH07] [RVZB09]. C'est l'un des enjeux les plus importants de bon nombre d'applications Internet ou de sites de e-commerces.
- Faire de l'intelligence économique : pour les sites marchands, l'objectif est ici de comprendre : quand et comment l'utilisateur a été attiré vers le site, qu'est ce qui l'a retenu, quelles sont les ventes croisées que l'on doit lui proposer et qu'est ce qui a motivé son départ ?

2.3 Processus de WUM

Le WUM consiste en "l'application des techniques de fouille des données pour découvrir des patrons d'utilisation à partir des données du Web dans le but de mieux comprendre et servir les besoins des applications Web" [Coo00]. La première étape dans le processus de WUM, une fois les données collectées, est le prétraitement des fichiers Logs qui consiste à nettoyer et transformer les données. La deuxième étape est la fouille des données permettant de découvrir des règles d'association, un enchainement de pages Web apparaissant souvent dans les visites et des " clusters " d'utilisateurs ayant des comportements similaires en terme de contenu visité. L'étape d'analyse et d'interprétation clôt le processus du WUM. Elle nécessite le recours à un ensemble d'outils pour ne garder que les résultats les plus pertinents.

2.3.1 Collecte des données

La première phase dans le processus du WUM consiste à collecter les données du Web à analyser. Les deux sources principales des données collectées sont les données enregistrées au niveau du serveur et les données enregistrées au niveau du client. Une autre source consiste aux données enregistrées au niveau du serveur Proxy, intermédiaire dans la communication client-serveur.

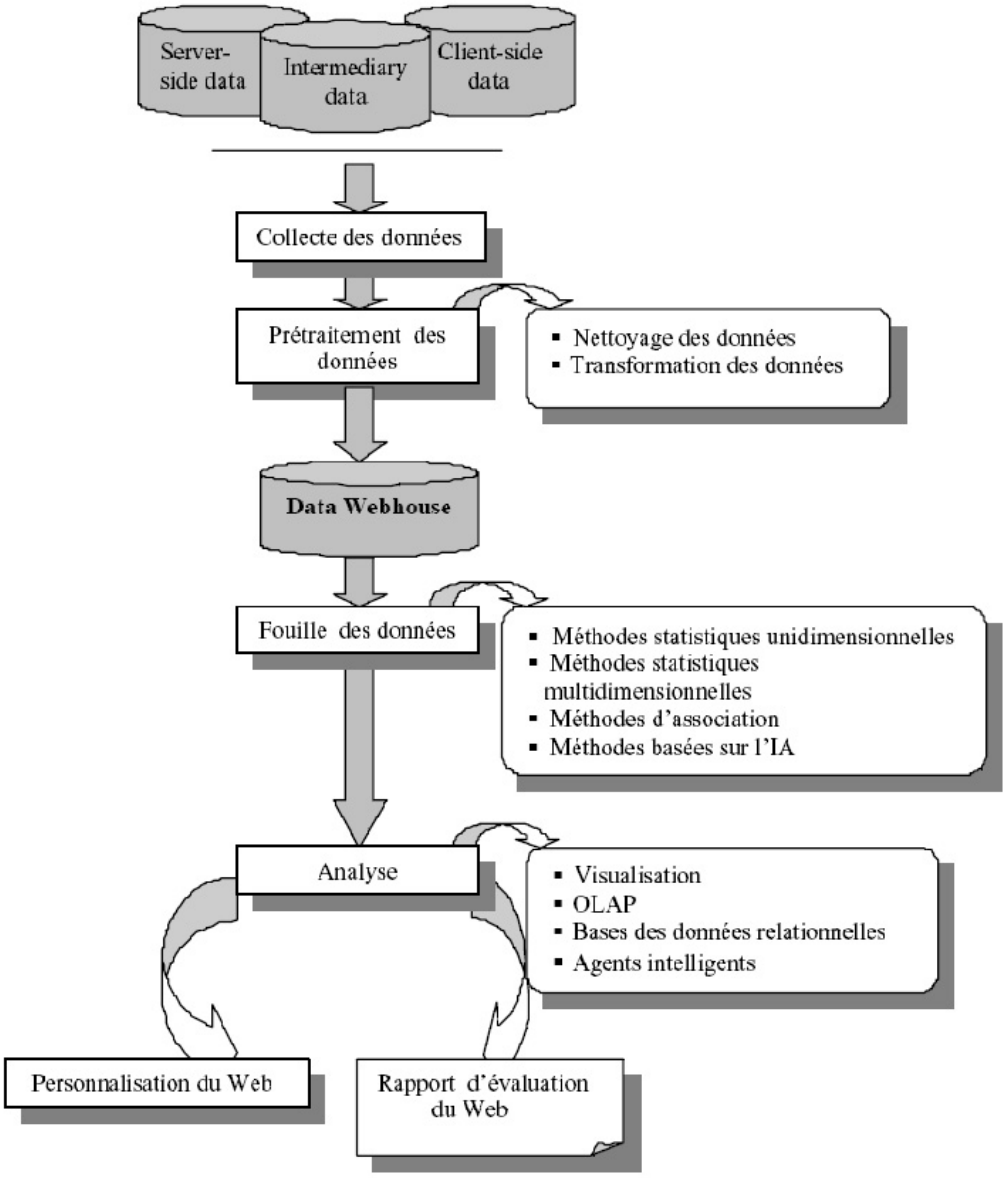


FIGURE 2.1 – Processus du Web Usage Mining

2.3.1.1 Données enregistrées au niveau du client

Les données sont collectées au niveau du poste client à travers des agents implémentés en Java ou en Java script. Ces agents sont incorporés dans les pages Web (sous forme d'applets java, par exemple) et utilisés pour une collecte directe des informations à partir du poste client (exemples d'informations : le temps

d'accès et d'abandon du site, l'historique de navigation). Une autre technique de collecte des données consiste à utiliser une version modifiée du navigateur [TG97]. Cette technique permet d'enregistrer les pages Web visitées par un utilisateur ainsi que le temps d'accès et le temps de réponse et les envoyer au serveur. La première méthode permet de collecter des données sur un utilisateur navigant sur un seul site Web. Par contre, un browser modifié permet la collecte des données sur un utilisateur navigant sur plusieurs sites Web. Le problème qui se pose dans le second cas est comment convaincre les internautes d'utiliser ce navigateur modifié dans leurs navigations sachant qu'il peut être considéré comme une menace de leur vie privée [SCDT00]. Les informations enregistrées au niveau du poste client sont plus fiables que les informations enregistrées au niveau du serveur puisqu'elles permettent de résoudre le problème du caching et l'identification des sessions [PPPS03].

2.3.1.2 Données enregistrées au niveau du Proxy

Le serveur Proxy joue le rôle d'intermédiaire entre des clients Web et des serveurs Web. C'est également un vaste espace disque servant au stockage des pages Web consultées par les utilisateurs (Web-cache server). En effet, pour toute requête émise sur une page, le Proxy, après consultation de son disque local, transmet la requête au serveur Web si le document n'est pas disponible à son niveau. Une fois l'information retournée par le serveur, le Proxy en effectue une copie locale sur son disque puis la transmet à l'initiateur de la requête. Le serveur Proxy garde la trace de toutes les communications établies dans des fichiers Logs semblables à ceux des serveursWeb. Ces traces peuvent révéler les requêtes HTTP émises par plusieurs clients vers plusieurs serveurs Web et servir ainsi de source de données pour caractériser le comportement de navigation d'un groupe anonyme d'utilisateurs partageant un même serveur Proxy [SCDT00] [FL05]. Cependant, les mêmes problèmes cités précédemment (problème du caching et d'allocation des adresses IP) sont présents au niveau du Proxy. Le tableau suivant présente les différents niveaux de collecte des données résultant de la navigation d'un ou de plusieurs utilisateurs sur un ou plusieurs sites.

2.3.1.3 Données enregistrées au niveau du serveur

A cours de sa navigation sur le site, l'utilisateur consulte des pages Web. La demande de ces pages déclenche des requêtes (affichage, téléchargement..) qui sont enregistrées en format texte et stockées de manière standardisée dans un fichier log, appelé *log web*. Ce fichier est maintenu par le serveur HTTP hébergeant le site. L'enregistrement des données dans les Logs du serveur (server-side Log files) permet d'identifier l'ensemble d'utilisateurs accédant au site Web. De plus, les Logs du serveur fournissent des données sur le contenu, des informations sur la

structure et des méta-informations sur les pages Web (taille du fichier, date de la dernière modification) [SCDT00] [FL05]. Cependant, les fichiers Logs des serveurs Web présentent quelques problèmes signalés dans le paragraphe suivant.

Présentation des fichiers Logs

Les fichiers logs se présentent sous forme de fichiers textuels (fig.2.2) organisés suivant un format particulier contenant généralement l'adresse IP du client (host), la date, la ressource envoyée au client (page), le code d'erreur (status), la page de provenance (referer) et l'identification du navigateur et du système d'exploitation (user agent).

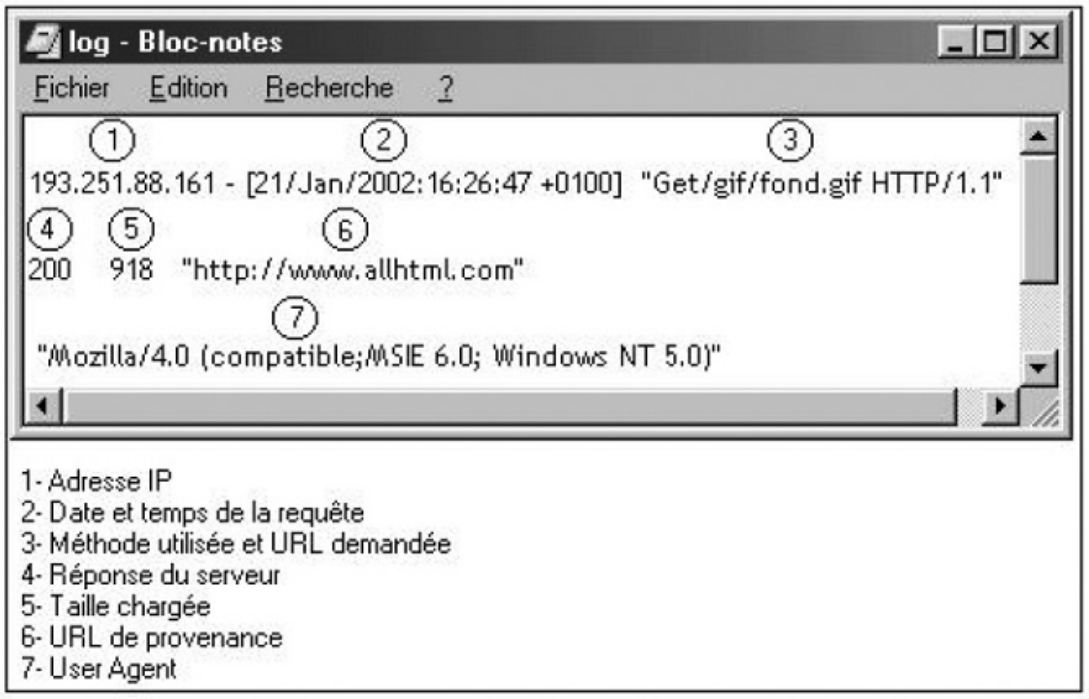


FIGURE 2.2 – Exemple de fichier Log Web

Plusieurs formats existent. Les formats les plus utilisés sont ECLF (Extended Common Log Format) préconisé par le W3C et le IIS Log File Format (Internet Information Services) qu'utilisent les serveurs web de Microsoft. Ces fichiers étant destinés, au départ, à des fins de maintenance, comportent beaucoup de bruits ou d'informations inutiles (requêtes effectuées par les robots, requêtes aux images, etc.). Un travail de prétraitement s'avère donc nécessaire pour identifier les demandes réelles des utilisateurs par rapport à d'autres données rajoutées par le serveur.

Un fichier log au format ECLF présente les informations suivantes :
(*ip*) (*name*) (*login*) (*date*) (*url*) (*status*) (*size*) (*referrer*) (*agent*)

- IP : adresse électronique de l'utilisateur ; cette adresse correspondant souvent au nom de domaine d'un serveur si l'utilisateur est connecté à internet via un fournisseur d'accès ou une entreprise
- Name/Login : supposent que l'utilisateur se soit lui-même identifié
- Date : date et heure précises de réception de la requête
- URL : adresse de la page visitée sur le site (www.<...>)
- Statut : code retour qui indique si l'action s'est bien déroulée (prend la valeur 200 en cas de réussite de la requête)
- Size : indique la taille du fichier retourné
- Referrer : la page précédemment visitée c.à.d la page de provenance (ou le moteur de recherche utilisé pour rejoindre la page Web suivi des mots clés demandés)
- Agent : le navigateur (Firefox, Internet Explorer, etc.) et le type de système d'exploitation de l'utilisateur (Windows, Linux, Mac OS, etc.)

Problèmes spécifiques aux données contenues dans les Logs

Bien que les données fournies par les fichiers Logs soient utiles, il importe de prendre en compte les limites inhérentes à ces données lors de leur analyse et de leur interprétation. Parmi les difficultés qui peuvent survenir :

- Les requêtes inutiles : chaque fois qu'il reçoit une requête, le serveur enregistre une ligne dans le fichier Log. Ainsi, pour charger une page, il y'aura autant de lignes dans le fichier que d'objets contenus sur cette page (les éléments graphiques). Un prétraitement est donc indispensable pour supprimer les requêtes inutiles.
- Le Web caching : afin de faciliter le trafic sur le Web, une copie de certaines pages est sauvegardée au niveau du navigateur local de l'utilisateur ou au niveau du serveur proxy afin de ne pas les télécharger chaque fois qu'un utilisateur les demande. Dans ce cas, une page peut être consultée plusieurs fois sans qu'il y'ait autant d'accès au serveur. Il en résulte que les requêtes correspondantes ne sont pas enregistrées dans le fichier Log.
- L'utilisation des robots : les annuaires du Web, connus sous le nom de moteurs de recherche, utilisent des robots qui parcourent tous les sites Web afin de mettre à jour leur index de recherche. Ce faisant, ils déclenchent des requêtes qui sont enregistrées dans tous les fichiers Logs des différents sites, faussant ainsi leurs statistiques.
- L'identification des sessions : toutes les requêtes provenant d'un utilisateur identifié constituent sa session. Le début de la session est défini par le fait que l'URL de provenance de l'utilisateur est extérieure au site [LTTV03].

Par contre, aucun signal n'indique la déconnexion du site et par suite la fin de la session.

- Le manque d'information : le fichier Log n'apporte rien sur le comportement de l'utilisateur entre deux requêtes : Que fait ce dernier ? Est-il vraiment en train de lire la page affichée ? De plus, le nombre de visites d'une page ne reflète pas nécessairement l'intérêt de celle-ci. En effet, un nombre élevé de visites peut simplement être attribué à l'organisation d'un site et au passage forcé d'un visiteur sur certaines.
- L'identification des utilisateurs : l'identification des utilisateurs à partir du fichier Log n'est pas une tâche simple. En effet, en employant le fichier Log, l'unique identifiant disponible est l'adresse IP et "l'agent" de l'utilisateur. Cet identifiant présente plusieurs limites [SCDT00] :
  - Adresse IP unique / Plusieurs sessions serveurs : la même adresse IP peut être attribuée à plusieurs utilisateurs accédant aux services du Web à travers un unique serveur Proxy.
  - Plusieurs adresses IP / Utilisateur unique : un utilisateur peut accéder au Web à partir de plusieurs machines.
  - Plusieurs agents / Utilisateur unique : un internaute qui utilise plus d'un navigateur, même si la machine est unique, est aperçu comme plusieurs utilisateurs.
  - Les firewalls : ces protections d'accès à un réseau masquent l'adresse IP des utilisateurs. Toute requête de connexion provenant d'un serveur doté d'une telle protection aura la même adresse et ce, quel que soit l'utilisateur. Il est donc impossible, dans ce cas, d'identifier et de distinguer les visiteurs provenant de ce réseau.

Compte tenu de l'ensemble de ces problèmes, le prétraitement des données provenant des fichiers Logs s'avère une phase incontournable dans le processus du WUM.

2.3.2 Prétraitement des données

Le prétraitement de fichiers logs a comme objectif la structuration et l'amélioration de la qualité des données contenues dans les fichiers pour les préparer à une analyse des usages. Cette étape est souvent la plus laborieuse et qui demande le plus de temps à cause de l'absence de structuration et la présence du bruit dans les données brutes d'usage. Les objets à identifier dans un processus de prétraitement de fichiers logs web sont les requêtes effectuées par des utilisateurs humains, les requêtes des robots Web ainsi que les sessions, les navigations et parfois les épisodes [Tan05].
Cette étape dans le processus du WUM, a été traitée dans de nombreux travaux dont les plus importants sont les travaux de Cooley [Coo00] qui propose une mé-

thode complète de traitement des fichiers logs basée sur les données contenues dans les fichiers logs et la carte du site, les travaux de Berendt et al. [BMSN02] qui introduisent dans leur méthodologie de prétraitement la durée des visites et les travaux récents de Tanasa et al. [Tan05] dont la méthodologie proposée réunit l'ensemble des méthodes et heuristiques classiques et propose une étape de prétraitement avancé. Dans tous ces travaux, le prétraitement des données se décompose en deux phases principales : une phase de nettoyage des données et une phase de transformation.

2.3.2.1 Nettoyage des données

L'étape du nettoyage consiste à filtrer les données inutiles à travers la suppression des requêtes ne faisant pas l'objet de l'analyse et celle provenant des robots Web. La suppression du premier type de requêtes dépend selon [Tan05] de l'intention de l'analyste. En effet, si son objectif est de trouver les failles de la structure du site Web ou d'offrir des liens dynamiques personnalisés aux visiteurs du site Web, la suppression des requêtes auxiliaires comme celles pour les images ou les fichiers multimédia est possible. Par contre, quand l'objectif est le " Web pre-fetching ", il ne faut pas supprimer ces requêtes puisque dans certains cas les images ne sont pas incluses dans les fichiers HTML mais accessibles à travers des liens, ainsi l'affichage de ces images indique une action de l'utilisateur. La suppression du second type de requêtes i.e. les entrées dans le fichier Log produites par les robots Web (WR) permet également de supprimer les sessions non intéressantes. En effet, les WRs suivent automatiquement tous les liens d'une page Web. Il en résulte que le nombre de demandes d'un WR dépasse en général le nombre de demandes d'un utilisateur normal. [Tan05] a utilisé trois heuristiques pour identifier les requêtes et les visites issues des WRs :

- Identifier les adresses IPs qui ont formulé une requête à la page "robots.txt".
- Utiliser des listes des "User agents" connus comme étant des WRs.
- Utiliser un seuil pour la vitesse de navigation " BS (Browsing Speed)", qui représente le rapport entre le nombre de pages consultées pendant une visite de l'utilisateur et la durée de la visite. Si BS est supérieure à deux pages par seconde et la visite dépasse 15 pages, alors la visite a été initiée par un WR.

2.3.2.2 Transformation des données

Cette phase regroupe plusieurs tâches telles que l'identification des utilisateurs et des sessions et l'identification des visites.

Identification des utilisateurs et des sessions

Plusieurs méthodes ont été proposées pour identifier les utilisateurs (tab. 2.1).

TABLE 2.1 – Principales méthodes d'identification des internautes

	Identification	**Durée**
Adresse IP	Groupe d'ordinateurs	Session
Identifiant de session	Individu	Session
Cookie	Ordinateur	Permanent/Session
Mot de passe	Individu	Permanent

- L'adresse IP : les adresses IP toujours disponibles et ne nécessitant aucun traitement préalable peuvent être utilisées pour identifier les internautes. Cependant, leur utilisation présente principalement deux limites. D'une part, les internautes utilisant un serveur Proxy sont identifiés par l'unique adresse IP de ce serveur. Ainsi, le site visité ne peut déceler s'il s'agit d'un ou de plusieurs visiteurs. D'autre part, l'attribution dynamique des adresses IPs ne permet une identification valable que pour une seule session ininterrompue *i.e.* si l'internaute interrompt sa visite en se déconnectant un bref instant, son adresse IP sera changée bien qu'il s'agit toujours du même utilisateur.
- Les cookies (Client Side Storage) : ces fichiers peuvent contenir des information telles que la date et l'heure de la visite, la page visitée, un code d'identification du client, etc. Chaque fois que l'utilisateur introduit une URL, le navigateur parcourt les cookies. Si l'un d'entre eux contient cette URL, la partie du cookie contenant les données associées est transférée conjointement à la requête afin de permettre au serveur d'identifier la provenance de cette requête. Cette méthode présente plusieurs avantages. En effet, les cookies permettent une identification s'étalant sur plusieurs sessions. Ils permettent également de stocker plus qu'un simple code d'identification et de collecter et d'enregistrer des informations directement exploitables par le serveur (comme le mot de passe) ; Cependant, l'identification par cookies présente des inconvénients. D'une part, les cookies identifient la machine, et non l'utilisateur ; D'autre part, ils nécessitent l'acceptation de l'utilisateur qui peut à tout moment désactiver leur chargement.
- Le mot de passe : pour qu'un serveur puisse identifier un visiteur de manière certaine, l'internaute doit s'identifier lui-même à l'aide d'un pseudonyme (Login) et un mot de passe (Password). Ainsi, le serveur est sûr de l'identité de son visiteur. Cette technique permet d'identifier les internautes de façon permanente et fiable mais elle requiert la participation de l'utilisateur et ne peut être réalisée à son insu. Le serveur devra donc attendre que son visiteur s'enregistre et ne pourra profiter des requêtes effectuées en dehors de l'identification. Pour remédier à cet inconvénient, les mots de passe et les pseudonymes sont souvent enregistrés dans un cookie. L'identification

établie lors d'une session ultérieure portera alors sur la machine et non plus sur l'utilisateur.

- L'identifiant de session : les identifiants de session permettent à un site entièrement dynamique d'identifier les internautes individuellement. Ils reposent sur la technologie PHP. Cette technique permet d'attacher un identifiant à chacun des liens hypertextes présents sur une page. Lors de la première requête émise, le serveur attribue arbitrairement à cette requête un identifiant de session, la réponse du serveur sera une page préparée dynamiquement. Le serveur peut ainsi insérer l'identifiant de session dans tous les liens hypertextes de cette page. Lorsque l'utilisateur clique sur l'un de ces liens, sa requête contiendra automatiquement l'identifiant qui lui a été attribué au départ. Cette technique est très fiable mais limite l'identification du visiteur à une seule session.

D'autres méthodes ont été proposées afin de résoudre le problème d'identification de l'utilisateur. Dans [CMS99], la méthode proposée combine l'utilisation de la topologie du site et des informations contenues dans le referrer. Si une requête de page provient de la même adresse IP que les requêtes précédentes sans qu'il y'ait d'hyperliens directs entre les pages demandées, alors l'utilisateur n'est plus le même. Cependant cette méthode n'identifie pas complètement l'utilisateur. [Sch01] emploie une technique différente pour identifier l'utilisateur. Cette technique consiste à inclure, pour chaque utilisateur, un identifiant unique généré par le serveur Web dans les URLs des pages Web du site. Cependant, cette technique nécessite l'intervention de l'internaute qui doit créer un signet, qui inclut l'identifiant comme une partie de l'URL dans l'une des pages afin d'identifier l'utilisateur s'il revient au site. Le tableau 2.2 proposé par [Gav02] présente une comparaison entre les principales techniques d'identification des internautes.

Ainsi, il s'avère que toutes les techniques proposées présentent des inconvénients dont le plus important est l'introduction dans le domaine privé de l'utilisateur. Cependant, dans [BMSN02], les auteurs rapportent que la combinaison de l'*adresse IP* et le *User Agent* constitue un bon identificateur de l'utilisateur dans 92% des cas.

Identification des visites ou des navigations

Une fois l'utilisateur identifié par l'une de méthodes décrites ci-dessus, il est possible de reconstituer sa session en regroupant les requêtes contenues dans les fichiers Log et émises par cet utilisateur. Selon [SFW99], les méthodes d'identification des sessions des utilisateurs peuvent être classifiées en méthodes basées sur le contexte (exemple : accès à des pages de types spécifiques) et méthodes basées sur le temps (exemple : limite seuil de temps de consultation d'une page). Les méthodes basées sur le temps sont les plus couramment utilisées. Elles consistent à considérer que l'ensemble des pages visitées par un utilisateur constitue une

TABLE 2.2 – Comparaison des méthodes d'identification des internautes

	Avantages	**Inconvénients**
Adresse IP	- Toujours disponible - Aucun traitement préalable	- Identifie un groupe d'ordinateurs - Problème d'attribution dynamique
Identifiant de session	- Grande fiabilité	- Limité aux sites Web dynamiques
Cookie	- Simplicité de mise en oeuvre - stocke plus d'un simple code d'identification	- Identifie la machine - Un seul cookie pour plusieurs utilisateurs - Désactivation ou destruction possible - Inadapté à la mobilité - Introduction dans la vie privée
Mot de passe	- Aucune approximation	- Nécessité de l'intervention humaine

visite unique si les pages sont consultées pendant un intervalle de temps ne dépassant pas un certain seuil temporel. Ce " temps de vue de pages " varie de 25,5 minutes [CP95] à 24 heures [YJGMD96]. Le temps de vue de pages couramment utilisé est de 30 minutes [CMS99]. Cependant, l'utilisateur peut passer plus de trente minutes à lire la même page ou quitter son poste pendant un moment et retourner pour consulter la même page. De plus, l'utilisateur du cache peut donner l'impression que la session est finie alors qu'il consulte les pages enregistrées par le cache. Selon les critères empiriques de Kimball [KM00], une visite est caractérisée par une série d'enregistrements séquentiellement ordonnés, ayant la même adresse IP et le même nom d'utilisateur, ne présentant pas de rupture de séquence de plus d'une certaine durée.

Identification des épisodes

L'objectif de l'identification des épisodes est de créer des classes de référence significatives pour chaque utilisateur. Selon [CMS97], les épisodes dépendent du comportement de navigation de l'utilisateur. En se basant sur cette hypothèse, les auteurs proposent de classifier les pages d'un site en pages auxiliaires contenant les hyperliens primaires aux autres pages Web et utilisées pour la navigation, pages de contenu contenant des informations intéressantes aux utilisateurs et pages considérées à la fois comme des pages auxiliaires et des pages de contenu. Elles contiennent les deux types d'information. Cette classification basée sur le contexte

dépend de l'utilisateur. En effet, une page de navigation (ou auxiliaire) pour un utilisateur peut être une page de contenu pour un autre. Suivant cette classification, il existe trois méthodes d'identification des épisodes : la référence-avant maximale (MF-Maximal Forward reference), le typage des pages et la longueur de la référence. Selon la méthode "Référence-avant maximale", proposée par [CY96], un épisode est défini par un ensemble de pages visitées par un utilisateur à partir de la première page enregistrée dans le fichier Log jusqu'à l'apparition de la première référence en arrière. Ainsi, cette méthode ne considère pas une deuxième fois les pages qui ont été traversées par l'utilisateur lors de sa visite, ce qui ne convient pas à certaines classes d'applications où il est important de prédire les types de référence en arrière. D'autre part, le Web caching empêche les références en arrière d'être enregistrées dans les fichiers Log. La méthode "typage de pages" [CMS99] est semblable à la méthode "longueur de référence". La différence entre les deux méthodes consiste dans l'algorithme de classification basé sur les données d'usage pour la méthode longueur de référence et sur le contenu de la page pour la méthode typage de page.

2.3.3 Fouille de données et analyse des résultats

Cette étape consiste à appliquer des techniques de fouille des données sur le fichier de sessions ou le fichier de navigations afin d'extraire des patrons d'utilisation du site.

2.3.3.1 Fouille des données

Bien qu'il existait plusieurs méthodes de classification utilisées dans la fouille de données, peu de ces méthodes sont appliquées aux données du Web vu la difficulté de les adapter aux particularités de ces données à savoir la taille des tableaux de sessions ou de pages. Ainsi, les méthodes les plus utilisés sont les règles d'association pour la découverte de motifs fréquents de navigation dans [MM06] et les motifs séquentiels de navigation dans [NER02]. Quand aux méthodes de classification, il est difficile, voire impossible, d'utiliser certaines d'entre elles à cause de la taille gigantesque des données du Web, surtout que la plupart des méthodes de classification retiennent toutes les données en mémoire. Les méthodes de classification les plus utilisées sont les cartes de Kohonen pour la classification des utilisateurs dans [SCDT00], [BT03], [ERCG$^+$04] et [ZEH07], CLIQUE [PE98], BIRCH dans [FSS00] et EM dans [CHM$^+$00], [ZO07] et [MJJ09].

Méthodes d'extraction de règles d'association

Comme le format et le contenu des fichiers logs se prêtent particulièrement bien aux méthodes de règles d'association, les travaux sur ce sujet ont été abondants.

Les principaux travaux utilisant les règles d'association en WUM appliquent majoritairement un algorithme de type Apriori [AR94] [Mic02] [Ram05] [MJM09]. D'autres algorithmes sont aussi utilisés tels que l'algorithme GSP [PPTK02] et l'algorithme FP-Growth [HPY00] [XZC+02].

Méthodes d'extraction de motifs séquentiels

La recherche de motifs séquentiels dérive de celle des règles d'association par l'introduction du temps qui sépare les requêtes entre elles et donne un aspect séquentiel aux informations contenues dans la base des données. Parmi les travaux qui sont intéressés à l'application des motifs séquentiels en WUM, les travaux de Spiliopoulou et al. [SFW99] [SP01] qui décrivent l'outil *Web Utilisation Miner* pour la découverte de motifs de visite "intéressants", les travaux de Masseglia et al. [MPC99b] [MPC99a] qui décrivent la plateforme *WebTool*. Citons aussi deux travaux effectués dans le cadre du projet Axis de l'Inria Sophia Antipolis qui visent à découvrir des motifs séquentiels avec un faible support dans des grands fichiers logs. Il s'agit de la méthode D&D (Diviser pour Découvrir) [MTT03] et la méthode C&D (Classer pour Découvrir) [TTM04][Tan05].

Méthodes de classification

De nombreuses stratégies ont été proposées dans la littérature afin de faciliter l'application de méthodes de classification sur les données issue des fichiers logs. Dans [MHLN02], les auteurs utilisent deux méthodes de classification qui ne prennent pas en compte l'ordre des pages dans les sessions. La première méthode de classification, appelée PACT (Profile Aggregation based on Clustering Transactions) classifie les sessions des utilisateurs. Les classes obtenues sont ensuite agrégées en profils. La seconde méthode classifie les pages en utilisant la technique ARHP ("Association Rule Hypergraph Partition") [HKKM97] qui permet de construire un graphe de pages les plus fréquentes à partir des successions de pages extraites avec un algorithme de type Apriori.
Dans [FSS00], les sessions des utilisateurs sont généralisées en utilisant une induction basée sur les attributs. Cette induction a comme objectif de réduire les dimensions des données. Lors d'une première étape, les pages sont organisées dans une hiérarchie construite seulement sur la syntaxe des URLs et non pas sur la sémantique des pages. Puis les sessions généralisées sont classées en utilisant l'algorithme BIRCH [ZRL96] de classification hiérarchique.
Dans [LTTV03], les auteurs ont appliqué deux analyses statistiques classiques ACP (Analyse en Composantes Principales) et ACM (Analyse des Correspondances Multiples) pour extraire des relations cachées entre les différents paramètres.
Dans [ZEH07], les auteurs appliquent les cartes de Kohonen pour la classification des sessions dans le cadre de personnalisation des ventes en ligne. Cette classi-

fication est réalisée à deux niveaux. Une classification hors ligne afin d'extraire des motifs de visites et des profils de visiteurs des sites commerciaux et une classification en temps réel afin d'examiner la session active et l'affecter à une classe particulière des utilisateurs ce qui permet de proposer au visiteur des produits appropriés.

2.3.3.2 Analyse, interprétation et validation des résultats

Cette dernière étape dans le processes du WUM est importante car elle permet de sélectionner les motifs intéressants et de les valdier. Certains travaux ont porté sur les outils de visualisation tels que ceux de Cadez et al. dans [CHM+00], auteurs du logiciel WebCANAVAS conçu pour la visualisation des classes des utilisateurs et les travaux de Labroche et al. [LLY07] qui proposent un nouvel outil d'analyse de l'usage et de visualisation des résultats. Cet outil repose sur l'application de l'algorithme Leader Ant [Lab06] [Lab07] pour la découverte de classes de sessions. Seules les classes les plus pertinentes sont représentées sur un graphe ce qui permet de pallier le problème des sites Web volumineux.
D'autres travaux ont abordé la notion d'intérêt (ou d'utilité) des motifs extraits [Tro00] dans le contexte des systèmes Web d'aide personnalisée et des systèmes d'aide à la conception de sites Web [KNY00].
Dans [KNY00], par exemple, les auteurs mesurent la notion d'intérêt par deux méthodes. La première s'appuie sur la distance en terme de liens entre deux pages (le nombre de liens à suivre pour arriver de la page A à la page B) et sur la co-occurrence des deux pages (la probabilité que les deux pages soient visitées ensemble). La représentation des deux mesures sur un même graphique permet aux auteurs de distinguer les pages fréquemment visitées ensemble mais qui sont "éloignées" sur le site. La seconde méthode s'appuie sur la pertinence conceptuelle entre deux pages, "inter-page conceptual relevance", c.à.d. la similarité entre les contenus des deux pages, et sur un ratio appelé "User traversal ratio" qui représente le rapport entre le nombre des utilisateurs ayant visité la page B puis la page A et le nombre des utilisateurs ayant visité le début du concept. Le coefficient de corrélation entre le "user traversal ratio" et "l'inter-page conceptual relevance" permet de trouver les paires de pages ayant une faible corrélation contrairement aux attentes du concepteur du site Web. La notion de mesure d'intérêt est également présente dans les travaux de Spiliopoulou et al. [SFW99] qui considèrent que l'intérêt d'un motif dépend de trois facteurs, à savoir le contenu des pages Web, la structure du site Web et les statistiques sur les accès aux pages.

2.4 Travaux existants et problèmes ouverts

Afin de mener à bien la tâche d'analyse du comportement des utilisateurs d'un site web, des nombreux travaux se sont concentrés sur la phase du prétraitement des fichiers logs puisque la qualité des résultats de l'analyse dépend fortement de la qualité des données en entrée. Diverses techniques ont été proposées dans la littérature du WUM, tels que celles présentées dans [CMS99], [Tan05], [ASL08], [AJL08] et [GMD08].
Afin de rendre un processus de Web Usage Mining plus efficace et pertinent, plusieurs auteurs proposent, au niveau de la phase du prétraitement des données, une généralisation ou une catégorisation des pages visitées par les internautes pour améliorer les résultats de l'application de certaines méthodes de fouille des données sur les données de l'usage. Dans [FSS00], les auteurs utilisent une généralisation syntaxique des URLs en remplaçant les rubriques syntaxiques de niveau supérieur à deux par leurs rubriques syntaxiques de niveau inférieur et ce avant d'appliquer une classification hiérarchique par BIRCH [ZRL96]. Cette hiérarchie est construite seulement sur la syntaxe des URLs et non pas sur la sémantique des pages. Par exemple, au lieu de
http ://tourisme.mairie-metz.fr/francais/manif/agenda.html
ils proposent d'utiliser
http ://tourisme.mairie-metz.fr/francais
ou bien
http ://tourisme.mairie-metz.fr/francais/manif.
Bien qu'elle soit automatique, cette généralisation syntaxique s'appuie trop sur l'organisation proposée par les concepteurs du site Web. Une mauvaise organisation génère implicitement une mauvaise classification. En plus, les performances de l'algorithme BIRCH se dégradent visiblement en augmentant la quantité des données ce qui limite la généralisation des pages à quelques niveaux. Dans [TT04], une généralisation basée sur des rubriques sémantiques est effectuée lors du prétraitement de logs Web. Ces rubriques (ou catégories) sont données *a priori* par un expert du domaine relatif au site Web considéré. Cette tâche est coûteuse en terme de temps aussi bien pour la définition que pour la mise à jour de telles catégories. Dans [TMT06], les auteurs proposent de regrouper les pages visitées dans différentes catégories lors d'un pré-traitement puis utiliser ces catégories, plutôt que les URLs, pour l'extraction des motifs séquentiels fréquents avec un support plus élevé. La catégorisation des URLs est effectuée à l'aide d'informations extraites à partir de l'accès à ces pages par les internautes. En effet, les auteurs considèrent que les mots-clés qui sont fréquemment employés dans un moteur de recherche pour accéder à une page, peuvent être utilisés pour caractériser cette page. Ces mots-clés sont accessibles dans le champs referer des requêtes contenues dans les fichiers Logs. Une matrice croisant les mots-clés est construite en utili-

sant l'indice de dissimilarité de Jaccard puis une classification de ces mots-clés grâce à la 2-3CAH [CBT04] est effectuée avant d'affecter les URLs aux différents clusters.
Dans [Mob99], les auteurs proposent la classification des pages visitées après la découverte des classes d'usage afin de faciliter la personnalisation du site en proposant aux utilisateurs des pages qui pourraient les intéresser. La même idée a été appliquée dans [HMS09] qui propose une technique de généralisation des sessions pour faciliter l'analyse de l'usage.
Ainsi, plusieurs travaux ont associé la catégorisation ou la classification des pages à l'analyse de l'usage. Cependant, la majorité de ces travaux catégorise les pages en se basant soit sur les URLs soit sur les mots-clés utilisés pour indexer la page. Le contenu textuel des pages n'intervient pas généralement dans de telles classifications. Il serait donc intéressant d'introduire le contenu des pages Web dans l'analyse de l'usage étant donné que le comportement des internautes sur un site dépend fortement du contenu proposé dans les pages. Ce contenu doit lui même être mis à jour et modifié selon les besoins des utilisateurs du site.

2.5 Synthèse

Dans ce chapitre, nous avons présenté la méthodologie généralement adoptée pour la découverte des modèles comportementaux sur les sites Web. Cette méthodologie repose essentiellement sur une phase de prétraitement afin de structurer les données contenues dans les fichiers Logs et les transférer dans un environnement plus facile à exploiter (comme par exemple dans une base des données) et une phase d'apprentissage ou de modélisation à l'aide des méthodes de data mining. Les méthodes couramment utilisées dans cette étape sont les méthodes de classification automatique, à savoir les réseaux de neurones et la classification hiérarchique, qui s'appliquent d'une façon dissymétrique sur les deux ensembles de données (les lignes ou les colonnes) et privilégie un des deux ensembles contrairement par exemple à l'analyse factorielle des correspondances qui permet d'obtenir simultanément des résultats sur les deux ensembles ; Il est alors intéressant de rechercher simultanément une partition des deux ensembles. Le chapitre suivant sera donc consacré à ce type d'approches en y présentant les différents algorithmes proposés dans la littérature pour la mise en oeuvre d'une classification simultanée des lignes et des colonnes d'un tableau de données.

Chapitre 3

Approches et algorithmes de recherche de biclasses

3.1 Introduction

Les méthodes de classification automatique appliquées à des tableaux mettant en jeu deux ensembles de données agissent de façon dissymétrique et privilégient un des deux ensembles en ne faisant porter la structure recherchée que sur un seul ensemble. L'application d'une classification sur chaque ensemble est possible mais la détermination des liens entre les deux partitions est difficile. La recherche de structures de classes symétriques, plus précisément, la recherche simultanée de partitions sur les deux ensembles a donné naissance à des méthodes de classification simultanée ou bi-partitionnement. On parle aussi de la classification croisée ou la classification par blocs. Ce type de classification est connu dans la littérature anglaise sous différents noms. Souvent on parle de "two-mode clustering", "two-side clustering", "two-way clustering", "direct clustering" [Har72], "biclustering" [Mir96] ou encore "co-clustering" [DMM03]. Ces approches diffèrent souvent dans les algorithmes employés, la nature des blocs recherchés qui peuvent être isolés ou imbriqués, le nombre de blocs identifiés dans les données et la nécessité de fixer le nombre de classes sur les lignes et les colonnes. Ce type d'approches a suscité beaucoup d'intérêt dans divers domaines, en particulier celui des biopuces où l'objectif est de caractériser des groupes de gènes par des groupes de conditions expérimentales. Cependant, les travaux de synthèse sur les algorithmes de bipartitionnement sont concentrés sur les algorithmes appliqués en bioinformatique tels que les travaux de [MO04] et [TSS04]. Par ailleurs, les algorithmes de classification directe (ou block clustering) proposés par Hartigan [Har75] et les travaux de Govaert [Gov83] sur la classification croisée trouvent aujourd'hui leur application en *web usage mining*, recherche d'information et *text mining*. D'autres algorithmes sont également proposés pour la classification simultanée tels que les

algorithmes basés sur le modèle de mélange [NG05] et ceux basés sur la théorie de l'information [DMM03] [Rob02]. La description de ces algorithmes et du principe de la classification simultanée constituent l'objet de ce chapitre.

3.2 Principe général de bipartitionnement

Soient X et Y les deux ensembles sur lesquels est défini le tableau de données Tab.3.1. On considère S l'ensemble de tous les couples de partitions de X et de Y. Une fonction $W : S \rightarrow R^+$, appelé *Critère*, mesurant la qualité d'un couple de partitions et dépendant du type de tableau de données est définie. Le problème de bi-partitionnement consiste à déterminer l'élément S optimisant ce critère, c'est à dire le couple de partitions de X et Y qui minimisera ou maximisera suivant le cas la fonction W.

Soit A une matrice des données à n lignes et m colonnes, définie par l'ensemble $\{X = x_1, \ldots, x_n\}$ des lignes et l'ensemble $\{Y = y_1, \ldots, y_m\}$ des colonnes.
a_{ij}, $1 \leq i \leq n$ et $1 \leq j \leq m$, sont les éléments de la matrice A.

TABLE 3.1 – Matrice des données

	y_1	...	y_i	...	y_m
x_1	a_{11}	...	a_{1j}	...	a_{1m}
...	...	...	...	...	...
x_i	a_{i1}	...	a_{ij}	...	a_{im}
...	...	...	...	...	...
x_n	a_{n1}	...	a_{nj}	...	a_{nm}

Une classe ligne $A_{IY} = (I, Y)$ est un sous-ensemble $I = \{i_1, ..., i_k\}$ de lignes ($I \subseteq X$ et $k \leq n$) ayant un comportement similaire sous l'ensemble Y des colonnes. Une classe ligne (I, Y) peut être définie comme une sous-matrice $k \times m$. De même, une classe colonne $A_{XJ} = (X, J)$ est un sous-ensemble de colonnes $J = \{j_1, ..., j_s\}$ ($J \subseteq Y$ et $s \leq m$) ayant un comportement similaire sous l'ensemble X des lignes. Cette classe colonne (X, J) peut être définie comme une sous-matrice $n \times s$. Une biclasse $A_{IJ} = (I, J)$ tels que $I = \{i_1, ..., i_k\}$ ($I \subseteq X$ et $k \leq n$) et $J = \{j_1, ..., j_s\}$ ($J \subseteq Y$ et $s \leq m$) est un sous ensemble de lignes ayant un comportement similaire sous un sous-ensemble de colonnes et vice-versa. Elle peut être définie comme une sous matrice $k \times s$. Les algorithmes de bipartitionnement ou de classification simultanée ont pour objectif d'identifier un ensemble de biclasses $B_k = (I_k, J_k)$, I_k est une classe définie sur X et J_k est une classe définie sur Y, tel que chaque biclasse B_k satisfait certains critères d'homogénéité. Ces critères varient d'un algorithme à un autre.

3.3 Partitionnement et bipartitionnement

Contrairement aux méthodes de partitionnement simple qui s'appliquent séparément sur les lignes et les colonnes, les méthodes de bipartitionnement s'appliquent simultanément sur les deux dimensions et produisent des blocs homogènes dans lesquels chaque individu est caractérisé par un sous-ensemble d'attributs et chaque attribut caractérise un sous-ensemble d'individus (tab. 3.2).

TABLE 3.2 – Comparaison entre partitionnement et bipartitionnement

Partitionnement	**Bipartitionnement**
- Appliqué aux lignes et aux colonnes **séparément**	- Appliqué aux lignes et aux colonnes **simultanément**
- Produit des classes (sous-groupes) sur les lignes **ou** des classes sur les colonnes	- Produit des biclasses (classes sur les lignes **et** sur les colonnes) i.e. des blocs homogènes
- Chaque objet est défini en utilisant **tous** les attributs et chaque attribut caractérise **tous** les objets. ⇒ On parle d'un Modèle **global**	- Chaque objet est défini en utilisant un **sous-ensemble** des attributs, chaque attribut caractérise un **sous-ensemble** d'objets. ⇒ On parle d'un Modèle **local**
- Les classes sont **exclusives**	- Les biclasses **ne sont pas exclusives**/ **exhaustives**

Les biclasses dans ce modèle local ne sont ni exclusives ni exhaustives. En d'autres termes, certains individus et/ou variables n'appartiennent à aucune biclasse ou appartiennent à plusieurs biclasses à la fois alors que dans le partitionnement simple chaque individu est affecté à une et une seule classe. L'avantage du bipartitionnement réside principalement dans la découverte des corrélations importantes entre des sous-ensembles de lignes et de colonnes. Parmi les premiers travaux témoignant de l'intérêt des méthodes de bipartitionnement, ceux de Hartigan dans [Har75], Govaert dans [Gov77] et [Gov83], Bock dans [Boc79], et Marchotorchino dans [Mar87]. Plus récemment plusieurs auteurs se sont intéressés à ce type d'approches, citons par exemple les travaux de Vichi dans [Vic00], Vichi et al. dans [VK01], Bock dans [Boc03], Dhillon et al. dans [Dhi01] et [DMM03], Van Mechelen et al. dans [MBB04], Madeira et Oliveira dans [MO04] et Nadif et Govaert dans [NG05].

3.4 Type et structure des biclasses

3.4.1 Type des biclasses

Les algorithmes de bipartitionnement permettent de découvrir quatre types de biclasses [MO04].

- Biclasses à valeurs constantes : une biclasse à valeurs constantes est une sous-matrice (I, J) tel que $\forall i \in I, \forall j \in J, a_{ij} = \mu$. Cependant, dans le cas réel, compte tenu du bruit dans les données, la valeur a_{ij} se présente généralement sous la forme $\eta_{ij} + \mu$, où η_{ij} est le bruit associé à la valeur réelle de a_{ij} (fig.3.1(A)).
- Biclasses à valeurs constantes sur les lignes ou les colonnes : une biclasse à valeurs constantes sur les lignes (fig.3.1(B)) est une sous-matrice (I, J) où toutes les valeurs a_{ij} de la biclasse sont obtenues en utilisant le modèle additif ou le modèle multiplicatif suivant.
  - Modèle additif : $a_{ij} = \mu + \alpha_i$
  - Modèle multiplicatif : $a_{ij} = \mu \times \alpha_i$

  μ est une valeur caractéristique de la biclasse et α_i est l'ajustement sur la ligne $i \in I$. Une biclasse à valeurs constantes sur les colonnes (fig.3.1(C)) est caractérisée par un ajustement β_j sur la colonne $j \in J$.

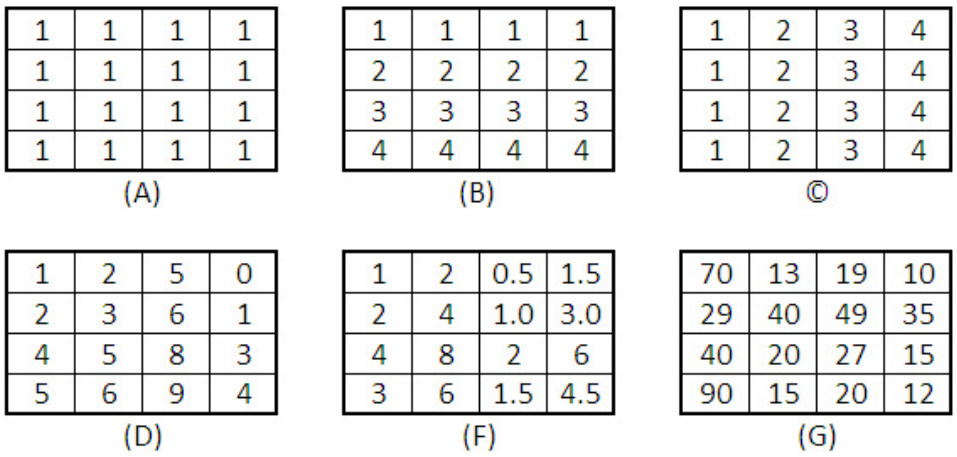


FIGURE 3.1 – Exemple des biclasses

- Biclasses à valeurs cohérentes : une biclasse à valeurs cohérentes est définie en utilisant le modèle additif (fig.3.1(D)) ou le modèle multiplicatif (fig.3.1(F)) .
  - Modèle additif : $a_{ij} = \mu + \alpha_i + \beta_j$
  - Modèle multiplicatif : $a_{ij} = \mu \times \alpha_i \times \beta_j$

où μ est une valeur caractéristique de la biclasse, α_i est l'ajustement sur la ligne $i \in I$ et β_j est l'ajustement sur la colonne $j \in J$.

- Biclasses à évolutions cohérentes : les approches visant à identifier dans les données des biclasses à évolutions cohérentes (fig.3.1(G)) considèrent que les éléments de la matrice sont des valeurs symboliques et essayent de découvrir des sous-ensembles de lignes et des sous-ensembles de colonnes dont l'évolution est cohérente par rapport à la valeur exacte de la matrice des données.

3.4.2 Structure des biclasses

Certains algorithmes de bi-partitionnement cherchent dans les données une seule biclasse, d'autres ont pour objectif d'identifier plusieurs biclasses dont le nombre est généralement fixé *a priori*. Les biclasses découvertes dans une matrice des données peuvent avoir l'une des structures suivantes :

- Biclasses exclusives sur les lignes et les colonnes : les algorithmes produisant cette structure, DCC (Double Conjugated Clustering) [BJK02] par exemple, supposent que chaque ligne et chaque colonne de la matrice des données appartient exclusivement à une seule biclasse (Fig. 3.2(b)).
- Structure d'échiquier sans chevauchement : certains algorithmes tels que l'algorithme spectral [KBJG03] considèrent que les lignes et les colonnes peuvent appartenir à plus d'une biclasse tout en présentant la structure d'un échiquier. Ils construisent ainsi K biclasses non exclusives mais sans chevauchement. Chaque ligne appartient exactement à K biclasses (Fig. 3.2(c)).
- Biclasses exclusives sur les lignes : dans ce genre de biclasses, chaque ligne ne peut appartenir qu'à une seule biclasse. Par contre, chaque colonne peut appartenir à une ou plusieurs biclasses (Fig. 3.2(d)).
- Biclasses exclusives sur les colonnes : contrairement à la structure précédente, chaque colonne ne peut appartenir qu'à une seule biclasse alors que les lignes peuvent appartenir à une ou plusieurs biclasses (Fig. 3.2(e)).
- Biclasses sans chevauchement présentant une structure arborescente (Fig. 3.2(f)).
- Biclasses non exclusives sans chevauchement : dans les structures précédentes, les biclasses sont exhaustives. En effet, chaque ligne et chaque colonne de la matrice des données appartient à au moins une biclasse. Dans les biclasses non exclusives sans chevauchement, chaque pair (ligne, colonne) appartient à une et une seule biclasse (Fig. 3.2(g)).
- Biclasses avec chevauchement et présentant une structure hiérarchique : dans cette structure, les biclasses sont soit disjointes soit contenues l'une dans l'autre (Fig. 3.2(h)).

– Biclasses positionnées arbitrairement avec chevauchement : cette structure est la plus générale. Elle permet le chevauchement des biclasses et qu'une biclasse soit contenue dans une autre. En plus, dans cette structure, les biclasses ne sont pas exhaustives puisque certaines lignes ou colonnes n'appartiennent à aucune biclasse (Fig. 3.2(i)).

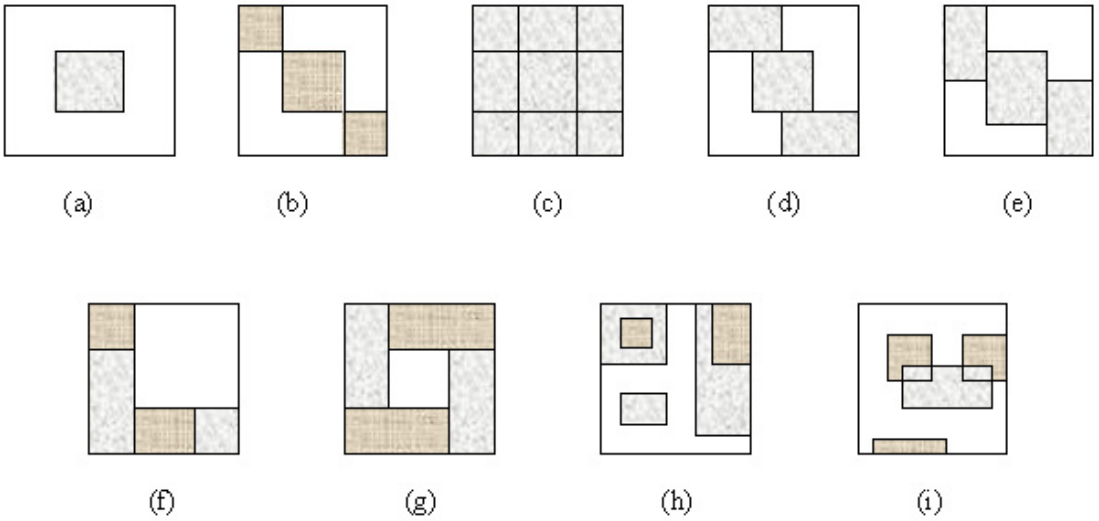


FIGURE 3.2 – Structure des biclasses

3.5 Approches de bipartitionnement

Dans [MO04], les auteurs proposent de classifier les approches utilisées pour la construction des biclasses en cinq catégories.

– IRCCC (Iterative Row and Column Clustering Combination) : le principe de cette approche est d'appliquer un algorithme de partitionnement simple (kmeans, CAH,...) sur les lignes et les colonnes de la matrice séparément puis de combiner les résultats pour construire des biclasses.
– DC (Divide and Conquer) : cette approche affecte initialement tous les éléments à la même biclasse et procède à un découpage itératif. Elle présente l'avantage d'être rapide à identifier les biclasses dans les données. Cependant, elle a comme inconvénient de ne pas pouvoir garder certaines bonnes biclasses avant leur découpage.
– GIS (Greedy Iterative Search) : l'approche GIS est basée sur l'idée de créer des biclasses en ajoutant ou supprimant des lignes ou des colonnes afin d'optimiser un certain critère. Malgré sa rapidité, cette approche présente

l'inconvénient de ne pas garder certaines bonnes biclasses en ajoutant ou supprimant des lignes ou des colonnes.
- EBE (Exhaustive Bicluster Enumeration) : l'approche EBE est basée sur l'idée que l'identification des meilleures biclasses nécessite une énumération exhaustive de toutes les biclasses possibles dans la matrice des données. L'avantage de cette approche est la certitude de trouver les meilleures biclasses mais son inconvénient majeur est sa complexité qu'il faut réduire en appliquant des restrictions sur la taille des biclasses.
- DPI (Distribution Parameter Identification) : l'approche DPI suppose la présence d'un modèle statistique dans les données et cherche à identifier la distribution des paramètres utilisés pour générer les données en minimisant certains critères à travers une approche itérative.

[Bes05] propose de classifier les approches de bipartitionnement en quatre grandes classes de méthodes heuristiques : les méthodes agglomératives, les méthodes divisives, les méthodes par permutation et les méthodes par approximation de paramètres. La première classe de méthodes commence avec une partition discrète des lignes et/ou des colonnes puis les éléments de cette partition sont affectés successivement aux différents groupes. Ces groupes forment finalement les bipartitions. Les méthodes divisives débutent avec un motif formé de toutes les lignes et toutes les colonnes. Ce motif est par la suite découpé successivement en bi-ensembles plus petits. Les méthodes par permutation déplacent des éléments entre groupes afin d'améliorer la qualité globale de la classification. Cette amélioration est bien souvent locale de sorte que la collection finale n'est souvent qu'un optimum local. La dernière classe de méthodes est assez différente. Elle cherche à calculer des paramètres tels que la moyenne, l'écart type, etc et s'applique souvent aux modèles probabilistes ou statistiques.

3.6 Algorithmes de bipartitionnement

Dans cette section, nous proposons de présenter une liste "exhaustive" des algorithmes de bipartitionnement classifiés selon le type du tableau des données sur lequel ils s'appliquent.

3.6.1 Algorithmes de bipartitionnement des tableaux de mesure

La majorité des algorithmes de bipartitionnement des tableaux de mesure sont développés pour l'analyse des biopuces en bioinformatique. Les autres algorithmes, tels que CROEUC, Block-EM et Block-CEM sont appliqués sur des données issues de domaines différents.

3.6.1.1 Algorithmes basés sur l'approche IRCCC

Ces algorithmes utilisent des algorithmes de classification simple appliqués séparément sur les lignes et les colonnes de la matrice des données.

Algorithme CTWC "Coupled Two-Way Clustering "

L'algorithme CTWC (Coupled two-way clustering) [GLD00] consiste à appliquer un algorithme de classification hiérarchique, le SPC "SuperParamagnetic Clustering (SPC)" sur les colonnes en utilisant toutes les lignes puis sur les lignes en utilisant toutes les colonnes. Toutes les sous-matrices (I, J) tel que I est une classe sur les lignes et J une classe sur les colonnes sont calculées. Seules les sous-matrices qui satisfont un certain critère comme la stabilité ou une taille minimale sont retenues. Ensuite le processus est réitéré : des classes de lignes et de colonnes sont extraites à partir de ces sous-matrices, etc.

Algorithme ITWC "Interrelated Two-Way Clustering"

L'algorithme ITWC [ZRZ01] fait appel à un algorithme de partitionnement simple, tels que k-means ou SOM, appliqué séparément sur les deux dimensions de la matrice des données pour obtenir des biclasses. La première étape de l'algorithme ITWC consiste à appliquer l'algorithme de partitionnement simple sur les lignes, puis en se basant sur les classes obtenues à l'étape 1, le même algorithme est appliqué sur les colonnes. Le nombre de classes recherchées sur les colonnes est généralement $k = 2$. L'étape suivante consiste à combiner les résultats de la première et la deuxième étape pour retrouver les biclasses. L'évaluation des biclasses est effectuée à l'aide de la validation croisée.

Algorithme DCC "Double Conjugated Clustering"

[BJK02] propose de partitionner l'ensemble des lignes et l'ensemble des colonnes à l'aide des cartes auto-organisatrices de Kohonen (SOM) et relier les deux partitions par l'intermédiaire d'une bijection associant à chaque noeud (i.e. vecteur représentant chaque classe) de l'un des deux espaces un noeud de l'autre espace appelé conjugué. Cette méthode présente l'avantage de convergence relativement rapide et aboutit à la construction de deux partitions, une dans l'espace des lignes et l'autre dans l'espace des colonnes. Chacune de ces partitions est le conjugué de l'autre.

Algorithme CROEUC de classification croisée

L'algorithme CROEUC (classifcation CROisée optimisant un critère basé sur la distance EUClidienne) est proposé par Govaert dans [Gov83]. Il consiste à chercher alternativement la meilleure partition P de X en fixant la partition Q de Y et inversement. La meilleure partition, celle qui optimise l'inertie intraclasse,

est obtenue en appliquant l'algorithme des centres mobiles au nuage associé aux individus après le regroupement des variables et vice versa. Le critère optimisé est l'information associée au tableau $A(P, Q)$.

3.6.1.2 Algorithmes basés sur l'approche DC

Dans l'approche DC (Divide and Conquer) les éléments à classifier ne sont ni les objets ni les variables mais les valeurs elles-mêmes du tableau de données. Ces algorithmes sont nommés par Hartigan [Har75] : algorithmes de *classification directe*.

Algorithme One-way splitting

Le "one way splitting" est un algorithme divisif qui propose un découpage en blocs homogènes des objets. Il se concentre principalement sur la partition des objets, en essayant de construire des classes de telle manière que les variables aient une variance intra-classe inférieure à un certain seuil. L'idée de base de l'algorithme est de n'utiliser que les variables ayant une variance supérieure au seuil dans une classe donnée pour découper cette classe.

Two-way splitting

Lorsque les données sont directement comparables d'un attribut à un autre, Hartigan [Har75] propose un algorithme divisif, Two-way splitting, qui choisit à chaque étape entre une division de l'ensemble des instances et une division de l'ensemble des attributs. Ce choix est basé sur la réduction au maximum de l'hétérogénéité du groupe d'instances ou de variables à diviser. Afin de respecter les contraintes hiérarchiques imposées pour cet algorithme, les divisions effectuées à une étape ne sont jamais remises en cause aux étapes suivantes. L'avantage de cet algorithme est qu'il ne nécessite pas de savoir à l'avance le nombre de blocs à obtenir.

3.6.1.3 Algorithmes basés sur l'approche GIS

Algorithme δ-biclusters

Le principe de l'algorithme δ-biclusters proposé par Cheng et Church [CC00] consiste à supprimer itérativement des lignes et des colonnes à partir de la matrice initiale jusqu'à ce que la mesure de distance soit inférieure à un certain seuil, puis ajouter des lignes et des colonnes itérativement sans entraîner une augmentation de cette mesure de distance. A chaque itération, une biclasse est générée puis remplacée dans la matrice initiale par des valeurs aléatoires. Une limite de cette approche est que le nombre de biclasses à rechercher doit être fixé par l'utilisateur tout comme le seuil δ utilisé pour la mesure de la qualité. En plus, la qualité des

biclasses diminue à chaque itération à cause des valeurs aléatoires ajoutées à chaque itération. Pour pallier ce problème, [YWWY03] proposent l'algorithme FLOC (FLexible Overlapped biClustering) permettant de prendre en compte les valeurs manquantes qui ne sont plus considérées dans le calcul de la moyenne des lignes, la moyenne des colonnes et la moyenne de la biclasse utilisée pour le calcul de la distance. Cet algorithme est réalisé en deux phases. Dans la première phase, k biclasses initiales sont générées en affectant chaque ligne ou colonne à chacune des biclasses avec une probabilité p. Dans la seconde phase, un processus itératif permet d'améliorer la qualité des biclasses.

Algorithme OPSM "Order-Preserving Sub-Matrix"

[BDCKY02] définissent une biclasse comme une sous-matrice préservatrice de l'ordre. Ils proposent l'algorithme OPSM dont l'objectif est de construire des larges biclasses. Une sous-matrice est préservatrice de l'ordre s'il existe une permutation des colonnes permettant d'avoir des valeurs strictement croissantes sur chaque ligne. [BDCKY02] définit un modèle complet comme étant un couple (J, π) où J est un ensemble de colonnes et π une permutation. Une ligne de la matrice suit ce modèle si ses valeurs ordonnées suivant la permutation π sont strictement croissantes. Soit la matrice des données $X \times Y$, le problème de détermination d'un OPSM de taille $k \times s$ est NP-complet. L'idée de l'algorithme est de commencer par identifier des modèles partiels. Ces modèles sont agrandis pour aboutir à des modèles complets.

Algorithme δ-patterns

L'algorithme δ-patterns proposé par [CST00] a pour objectif d'identifier des biclasses maximales à valeurs constantes sur les lignes. Ils définissent une ks-biclasse δ-valide comme un sous-ensemble I de lignes, de taille k et un sous-ensemble J de colonnes, de taille s tels que la différence entre la valeur maximale et la valeur minimale de chaque ligne est inférieure à un certain seuil. Une ks-biclasse δ-valide est dite maximale si elle ne peut pas être étendue en une k's-biclasse δ-valide en y ajoutant des lignes telle que $k' > k$ ou en une ks'-biclasse δ-valide en y ajoutant des colonnes telle que $s' > s$.

3.6.1.4 Algorithmes basés sur l'approche EBE

Parmi les algorithmes basés sur l'approche EBE, l'algorithme SAMBA "Statistical Algorithmic Method for Bicluster Analysis" proposé par [TSS02] est le plus connu. Il est basé sur la théorie des graphes et la modélisation probabiliste. Dans SAMBA, la matrice des données est représentée par un graphe G biparti pondéré où chaque noeud n_i correspond à une ligne et chaque noeud n_j correspond à une colonne. L'arête entre le noeud n_i et le noeud n_j a un poids a_{ij}

correspondant à l'élément de la matrice se trouvant à l'intersection de la ligne i et la colonne j. Une biclasse correspond au sous-graphe $H = (I, J, E)$ de G et représente un sous-ensemble I d'objets (gènes) dont la valeur change significativement sous un ensemble de variables (conditions) J. L'objectif de l'algorithme SAMBA est de chercher dans les données des biclasses maximales. L'application de l'algorithme SAMBA est effectuée en deux étapes. Dans la première étape, les données sont normalisées et représentées par un graphe biparti. Dans la seconde étape, l'algorithme identifie les K bi-cliques maximales. Dans une phase ultérieure, SAMBA apporte des améliorations locales aux biclasses par ajout ou suppression des sommets, et sélectionne les biclasses similaires ayant un nombre important de sommets en commun.

3.6.1.5 Algorithmes basés sur l'approche DPI

Algorithme de Lazzeroni et Owen

L'algorithme proposé par [LO00] pour la classification d'expression de gènes est connu sous le nom de "Plaid models". L'idée de base est de représenter la matrice des données comme une superposition des biclasses. La matrice des données est représentée par une fonction linéaire de variables correspondant aux biclasses $f_{ij} = \sum_{k=0}^{K} \theta_{ijk}\rho_{ik}\psi_{jk}$ Avec K est le nombre de biclasses,ρ_{ik} vaut 1 si la ligne i appartient à la biclasse k et 0 sinon. ψ_{jk} vaut 1 si la colonne j appartient à la biclasse k et 0 sinon. Les valeurs de θ_{ijk} représentent les modèles d'analyse de la variance (ANOVA) qui varient selon le type des biclasses recherchées. Le problème de bi-partitionnement consiste à rechercher les valeurs des paramètres de manière à minimiser la distance euclidienne entre les valeurs observées et celles modélisées.

Algorithme spectral

Dans [KBJG03], les auteurs utilisent une approche spectrale pour le bipartitionnement en supposant que la matrice des données comporte une structure d'échiquier après normalisation. L'algorithme est basé sur la recherche de vecteurs propres dans la matrice des données A. En effet, à partir d'une partition U des variables et de la matrice des données A, on estime une partition V des objets par $V = R^{-1}AU$ où R est une matrice de normalisation des lignes. De même, on peut estimer la partition sur les colonnes par $U = C^{-1}A^TV$ où C est une matrice de normalisation des colonnes. Ainsi, le vecteur de classification de colonnes recherché U est le vecteur propre de la matrice $C^{-1}A^TR^{-1}A$ et le vecteur de classification de lignes V est le vecteur propre de la matrice $R^{-1}AC^{-1}A^T$. Une décomposition en valeurs propres permet de résoudre ces deux problèmes. Kluger et al. proposent de normaliser conjointement les lignes et les colonnes de la matrice par un prétraitement itératif. Les vecteurs propres associés aux plus

grandes valeurs propres sont partitionnés selon différents nombres de classes et différentes valeurs seuils.

3.6.2 Algorithmes de bipartitionnement des tableaux binaires

Algorithmes Block-EM et Block-CEM

L'approche modèle de mélange suppose que les données proviennent d'un mélange de plusieurs distributions de probabilité, où chaque composant du mélange correspond à une classe. Le problème de bipartitionnement consiste alors à retrouver pour chaque objet sa population d'origine la plus probable en fonction du vecteurs d'observations qui le caractérise. Block-EM et Block-CEM proposés par Govaert et Nadif [GN03] [GN05b] sont des extensions des algorithmes EM et CEM qui recherchent une double partition des individus et des variables. L'algorithme Block-CEM maximise alternativement la log-vraisemblance classifiante conditionnelle à la partition en colonne et la log-vraisemblance classifiante conditionnelle à la partition en ligne. Ainsi, il utilise de façon alternée l'algorithme CEM sur les individus en bloquant la partition en colonne puis sur les variables en bloquant la partition en ligne. L'algorithme Block-EM consiste à maximiser alternativement l'espérance de la log-vraisemblance classifiante conditionnellement à la partition en colonne, puis conditionnellement à la partition en ligne. L'algorithme Block-EM est plus lent que l'algorithme Block-CEM mais il conduit à une estimation plus fiable des paramètres du modèle de mélange. Quand les deux algorithmes sont appliqués aux données binaires, la distribution de probabilités est la distribution de Bernouilli.

Algorithme CROBIN

L'algorithme CROBIN (classification **CRO**isée d'un tableau **BIN**aire) proposé par Govaert dans [Gov83] a pour objectif d'obtenir en réordonnant les lignes et les colonnes du tableau initial suivant les deux partitions P et Q, des blocs homogènes de 1 ou de 0. A chaque couple (k, l) de classes, une valeur binaire idéale (1 ou 0) est associée. Le tableau binaire obtenu est appelé noyau. Le meilleur noyau N associé à un couple de partitions est formé des éléments définis comme étant l'élément 1 ou 0 majoritaire dans le couple (P_k, Q_l). L'objectif de l'algorithme consiste à minimiser l'écart entre le tableau initial structuré suivant les deux partitions P et Q et le tableau idéal N. L'algorithme CROBIN est rapide et donne de bons résultats lorsque les blocs ont les mêmes proportions et des degrés d'homogénéité semblables. Son inconvénient est qu'il nécessite de fixer le nombre de classes sur les lignes et les colonnes.

3.6.3 Algorithmes de bipartitionnement des tableaux de contingence

Algorithme CROKI2

L'algorithme CROKI2 (classification **CRO**isée optimisant le **Khi2** du tableau de contingence) proposé par Govaert [Gov83] a pour objectif de trouver une partition P de X en K classes et une partition Q de Y en L classes telle que le Khi2 de contingence du nouveau tableau construit en regroupant les lignes et les colonnes suivant les partitions P et Q soit maximum. L'algorithme CROKI2 consiste à déterminer une série de couples de partitions (P_n, Q_n) optimisant le Khi2 du tableau de contingence en appliquant alternativement sur X et sur Y une variante de la méthode des nuées dynamiques.

Algorithme BSGP

L'algorithme BSGP (Bipartite Spectral Graph Partitioning) proposé par Dhillon dans [Dhi01] utilise la théorie spectrale des graphes pour la représentation des documents et des mots-clés. Les sommets du graphe sont les documents et les mots-clés et les arêtes sont les liaisons entre les mots-clés et les documents dans lesquels ils apparaissent. Le poids de chaque arête correspond au nombre d'occurrences du mot-clé dans le document. Le problème de bipartitionnement correspond alors au problème de partitionnement spectral d'un graphe biparti. L'algorithme BSGP fait appel à la décomposition en valeurs singulières et à l'algorithme de Kmeans pour l'identification des classes des documents et des mots-clés. Cet algorithme, comme tous les algorithmes basés sur le Kmeans, nécessite de fixer le nombre de classes dans les données.

Algorithmes basées sur la théorie de l'information

Deux variantes d'une même méthode de bipartitionnement ont été développées d'une manière indépendante par Dhillon et al. dans [DMM03] et Robardet dans [Rob02]. Cette méthode consiste à considérer les deux partitions cherchées comme deux variables aléatoires à valeurs discrètes et à concevoir la recherche d'une bipartition comme un problème de maximisation de l'association entre ces deux variables. Les deux variantes produisent une partition par un processus d'optimisation locale. L'algorithme IT (Information-Theoretic co-clustering) [DMM03] utilise la mesure de divergence entre la distribution de probabilités de Kullback et Leibler et propose de fixer *a priori* le nombre de classes de chacune des deux partitions puis optimise localement la fonction en estimant itérativement une partition en fonction de l'autre jusqu'à la convergence. L'algorithme de Robardet et al. ne nécessite pas de fixer à priori le nombre de classes des deux partitions. Il utilise un algorithme d'optimisation locale stochastique qui procède également par ajustement itératif d'une partition en fonction de l'autre.

Algorithme Cemcroki2

L'algorithme Cemcroki2 proposé par [NG05] est une extension de l'algorithme Croki2 basée sur un modèle de mélange croisé de distributions de Poisson. L'algorithme Cemcroki2 repose sur l'utilisation de l'algorithme CEM "Classification EM" pour la classification alternée des lignes et des colonnes au lieu de l'algorithme des nuées dynamiques utilisé dans Croki2. Son avantage par rapport à Croki2 est qu'il s'applique mieux aux données réelles où les classes ne sont pas bien séparées et les proportions des classes sont différentes. Des expérimentations effectuées par les auteurs sur des données réelles ont montré aussi qu'il fournit des meilleurs résultats par rapport algorithmes IT et BSGP proposés par Dhillon avec un nombre de documents mal classés nettement inférieur. Cependant, comme tous ces algorithmes, l'algorithme Cemcroki2 nécessite de fixer le nombre de classes sur les lignes et les colonnes. Dans [GN05a], les auteurs montrent que la maximisation de la vraisemblance classifiante est équivalente à la maximisation de l'information mutuelle utilisée par Dhillon dans l'algorithme IT et approximativement équivalente à la maximisation du critère de χ^2 utilisé par Govaert dans l'algorithme Croki2. L'algorithme Cemcroki2 est considéré comme une généralisation de ces deux algorithmes puisque la maximisation du critère de χ^2 ou de l'information mutuelle supposent implicitement que les données sont issues d'un mélange croisé de distributions de Poisson avec des proportions égales.

3.7 Les treillis de Galois

La théorie des treillis de Galois a été développée sous le nom d'Analyse Formelle de Concepts. Plusieurs algorithmes constructifs ont été mis au point depuis lors [SS01], et certaines applications concrètes sont apparues récemment, principalement en fouille de données dans [LS98] et [STB+02].

3.7.1 Treillis de Galois : Formalisme mathématique

- Définition d'un treillis : Un treillis est un ensemble ordonné dans lequel deux éléments quelconques ont une borne inférieure (BI, ou inf) et une borne supérieure (BS, ou sup). Un treillis complet est un treillis pour lequel tout sous-ensemble possède une borne inférieure et une borne supérieure.
- Définition d'un contexte : Un contexte K est un triplet $K = (O, A, f)$ pour lequel O est un ensemble d'objets, A un ensemble d'attributs et f une application binaire de OA dans $\{0, 1\}$.
- Définition des connexions de Galois : Étant donné un contexte $K = (O, A, f)$, on définit une application de $P(O)$ dans $P(A)$ et une application de $P(A)$

dans $P(O)$, notées identiquement par l'opérateur " ' " de la façon suivante :

$$\forall I \subset O, I' = \{a \in A / \forall o \in I, f(o,a) = 1\}$$

$$\forall J \subset A, J' = \{o \in O / \forall a \in J, f(o,a) = 1\}$$

Ces applications sont appelées *connexions de Galois* du contexte. A' est appelé *dual* de A et B' est appelé dual de B. Pratiquement, A' est l'ensemble des attributs communs à tous les objets de A, et B' est l'ensemble des objets qui possèdent en commun tous les attributs de B. Les propriétés essentielles d'une connexion de Galois sont :

$$I_1 \subset I_2 \Rightarrow I_2' \subset I_1'$$

$$I \subset I''$$

$$I' = I'''$$

- Définition d'un concept : Étant donné un contexte $K = (O, A, f)$, le couple $C(I, J)$ est un concept de K si et seulement si $I' = J$ et $J' = I$.
- Extension et intension : I est appelé l'extension du concept C et J est appelé son intension. On note $I = extent(C)$ et $J = intent(C)$.
- Treillis de Galois : Soit $L(K)$ l'ensemble de tous les concepts d'un contexte K. Le treillis complet $L(K)$ des concepts d'un contexte est appelé le treillis de Galois ou le treillis conceptuel.

3.7.2 Treillis de Galois et bipartitionnement

Un concept formel $C(I, J)$ dans le treillis de Galois peut être assimilé à une biclasse (I, J) telle que l'extension correspond à une classe sur les lignes et l'intension correspond à une classe sur les colonnes. L'ensemble des concepts dans un treillis sont des biclasses dont la valeur binaire idéale associée est égale à 1. Ainsi, la construction d'un treillis de Galois correspond à la recherche de bipartitions dans les algorithmes de bipartitionnement.

3.7.3 Algorithmes de construction des treillis de Galois

Il existe deux familles d'algorithmes de construction de treillis : les algorithmes incrémentaux et les algorithmes non incrémentaux. Les algorithmes incrémentaux ([GMA91], [CR96], [Nor78],...) construisent le treillis au fur et à mesure que les objets arrivent, alors que les algorithmes non incrémentaux ([Che69], [B.G84],

[Bor86],...) construisent le treillis une fois le contexte entièrement connu. Tous ces algorithmes construisent des treillis sans considérer une quelconque spécificité du contexte. Leur complexité est exponentielle, et plusieurs techniques ont été développées afin de réduire les temps de calcul, notamment par simplification du contexte en amont, comme [NN98] à l'aide d'une fonction d'entropie.

3.8 Synthèse

Les algorithmes de bipartitionnement présentés dans ce chapitre diffèrent selon l'approche sur laquelle ils reposent, le type de biclasses obtenues, le type de tableau des données et le domaine d'application (voir tab. 3.3). La majorité de ces algorithmes, à savoir CTWC, δ-biclusters, δ-patterns, ITWC, DCC, OPSM, SAMBA, FLOC et l'algorithme spectral, ont été développés pour l'analyse des biopuces en bioinformatique, en particulier pour identifier des groupes de gènes ayant le même comportement sous un groupe de conditions expérimentales. Les autres algorithmes, ont été développés pour des applications différentes. Les deux algorithmes proposés par Dhillon dans [Dhi01] et [DMM03] ont été appliqués à des tableaux de contingence croisant des documents à des mots-clés pour la catégorisation des documents.

Certains de ces algorithmes présentent l'avantage de pouvoir identifier les biclasses sans avoir besoin de fixer *a priori* le nombre de biclasses ou le nombre de classes sur les lignes et les colonnes. D'autres, par contre, nécessitent d'avoir une idée sur la structure des données. Outre ce critère, le choix de l'algorithme le plus approprié au contexte d'application repose sur d'autres critères tels que la dimension du tableau des données et le type du tableau des données (tableau de mesure, tableau des données binaires, tableau de contingence). Dans le cadre de cette thèse, notre objectif est découvrir des biclasses de pages et de mots-clés dans un tableau de contingence qui comptabilise le nombre d'occurrences de chaque mot-clé dans les pages d'un site Web. Ainsi, nous nous intéressons aux algorithmes de bipartitionnement des tableaux de contingence. D'après [NG05], l'algorithme CemCroki2 appliqué à des données réelles fournit des meilleurs résultats par rapport aux algorithmes IT et BSGP proposés par Dhillon. Notre choix est donc limité aux algorithmes Croki2 et CemCroki2. Comme le contenu d'un site Web est généralement composé de pages organisées sous forme de groupes de pages traitant de sous-thèmes différents, nous estimons que nos classes de pages et de descripteurs ne présentent pas de structure de mélange assez importante. Par suite, nous choisissons d'appliquer l'algorithme Croki2 à notre jeu des données réelles.

TABLE 3.3 – Caractéristiques des algorithmes de bipartitionnement

Algorithme	**Fixer le NB. de Biclasses**	**Type de biclasses**	**Tableau de données**
One-way splitting	Non	Val. constantes	Tab. de mesures
Two-way splitting	Oui	Val. constantes	Tab. de mesures
CROEUC	Oui	Val. cohérentes	Tab. de mesures
CROKI2	Oui	Val. cohérentes	Tab. de contingence
CROBIN	Oui	Val. cohérentes	Tab. de données binaires
CTWC	Non	Val. constantes sur les colonnes	Tab. de mesures
Plaid Models	Oui	Val. cohérentes	Tab. de mesures
δ-biclusters	Oui	Val. cohérentes	Tab. de mesures
δ-patterns	Non	Val. constantes sur les lignes	Tab. de mesures
ITWC	Non	Val. cohérentes	Tab. de mesures
DCC	Non	Val. constantes	Tab. de mesures
OPSM	Non	Evolution cohérente	Tab. de mesures
SAMBA	Non	Evolution cohérente	Tab. de mesures
FLOC	Oui	Val. cohérentes	Tab. de mesures
Spectral	Non	Val. cohérentes	Tab. de mesures
IT	Oui	Val. cohérentes	Tab. de contingence
BSGP	Oui	Val. cohérentes	Tab. de contingence
[Rob02]	Non	Val. cohérentes	Tab. de contingence
Block-EM	Oui	Val. cohérentes	Tab. de mesures Tab. de données binaires
Block-CEM	Oui	Val. cohérentes	Tab. de mesures Tab. de données binaires
Cemcroki2	Oui	Val. cohérentes	Tab. de contingence

Deuxième partie

Contributions

Chapitre 4

Algorithme Croki2

4.1 Introduction

L'algorithme Croki2 de bipartitionnement des tableaux de contingence repose sur l'utilisation de l'algorithme Kmeans pour la classification des lignes et des colonnes du tableau des données. Dans ce chapitre, nous proposons une nouvelle variante basée sur une optimisation globale que nous comparons à la version originale de l'algorithme à l'aide d'un ensemble de critères tels que la complexité, l'aptitude à déterminer la solution optimale et leur performance dans la génération de partitions avec des classes non vides. Pour ce faire, nous utilisons des jeux des données que nous générons artificiellement afin de tester les deux algorithmes. La méthodologie de génération des données artificielles est présentée dans la première section de ce chapitre. La section suivante présente en détail les deux variantes de l'algorithme Croki2. La dernière section est consacrée à la comparaison des deux algorithmes.

4.2 Méthodologie de génération des données artificielles

Dans cette section, nous présentons notre méthodologie de génération des jeux des données ayant une structure semblable à celle d'un tableau des données représentant à la fois le contenu textuel d'un site Web et une structure de classes sur les lignes et sur les colonnes.
Notre principale motivation est de mesurer l'efficacité des algorithmes proposées dans la section suivante à retrouver la structure initiale des données d'une part, et de tester dans le chapitre suivant les critères d'identification du bon nombre de classes sur les lignes et les colonnes d'autre part.
Soient $(x_1,...,x_i,...,x_n)$ et $(y_1,...,y_j,...,y_p)$ les ensembles artificiels de n individus et

p variables que nous voulons créer sur les lignes et les colonnes respectivement. Une variable sur les colonnes correspond dans notre cas à une page et un individu sur les lignes correspond à un descripteur ou mot-clé.
Ainsi, à chaque descripteur i sur les lignes un vecteur x_i de pages est associé et à chaque page sur les colonnes un vecteur y_j de descripteurs est attribué, avec $x_i = (x_i^1, ..., x_i^j, ..., x_i^p)$ et $y_j = (y_j^1, ..., y_j^i, ..., y_j^n)$.

Soient (p_1,...,p_k,...,p_K) le vecteur des profils initiaux des K classes *a priori* sur les lignes et (q_1,...,q_l,...,q_L) le vecteur des profils initiaux des L classes *a priori* sur les colonnes. La valeur x_i^j correspond au nombre d'occurrences du mot-clé i dans la page Web j.
La répartition des individus sur les classes *a priori* est effectuée selon les pourcentages indiqués par le vecteur α sur les lignes et le vecteur β sur les colonnes, avec α = (α_1,...,α_k,...α_K) et β=(β_1,...,β_l,...β_L), $\alpha_k \in [0,1]$, $\beta_l \in [0,1]$ et

$\sum_{k=1}^{K} \alpha_k = 1$ et $\sum_{l=1}^{L} \beta_l = 1$

4.2.1 Algorithme de génération des données artificielles

L'algorithme que nous proposons commence par générer à partir du tableau des profils initiaux de dimensions (K,L) un nouveau tableau de profils sur les colonnes de dimensions (K, p=NombreDeVariables).

Pour chaque classe l sur les colonnes
    Pour chaque variable j de classe l
        $q_j = q_l$

L'étape suivante consiste à générer les données artificielles sur les lignes à partir du nouveau tableau des profils. Les entrées de l'algorithme permettant cette génération sont :
- NbOccMin = Nombre Minimum d'Occurrences
- NbOccMax = Nombre Maximum d'Occurrences
- NbIndivLig = Nombre d'Individus sur les Lignes
- NbIndivCol = Nombre d'Individus sur les Colonnes

Un entier compris entre $NbOccMin$ et $NbOccMax$ est tiré au hasard et attribué à la variable $NbOcc_i$. Cet entier représente le nombre total d'occurrences du terme i dans toutes les pages du site. Ce nombre d'occurrences devrait être distribué sur les pages du site en respectant la distribution des occurrences imposée par les prototypes initiaux des classes.
C'est la raison de l'introduction de la fonction *cumul* sur la fréquence des occurrences dans les prototypes des classes.

1 Pour chaque classe k sur les lignes, k dans l'intervalle [1,K]
2 Pour chaque individu i de la classe ligne k
3 $NbOcc_i \leftarrow$ random($NbOccMin$, $NbOccMax$)
4 Pour j dans l'intervalle [1, $NbOcc_i$]
5 $y \leftarrow$ random(0,1)
6 Pour l dans l'intervalle [1,L]
7 Si y $\leq$ cumul (P_k^l)
8 $x_i^l = x_i^l + 1$
9 FinPour
10 FinPour
11FinPour
12 FinPour

4.2.2 Simulation d'un site Web

Les données artificielles que nous générons simulent le contenu textuel d'un site Web transformé en tableaux de contingence croisant les termes aux pages. Comme les données réelles sur lesquelles nous allons appliquer notre approche sont issues d'un site Web composé d'une centaine de pages web, nous fixons la dimension des tableaux générés à 200 lignes et 100 colonnes (Tab. 4.1) .

TABLE 4.1 – Jeux des données avec biclasses séparées

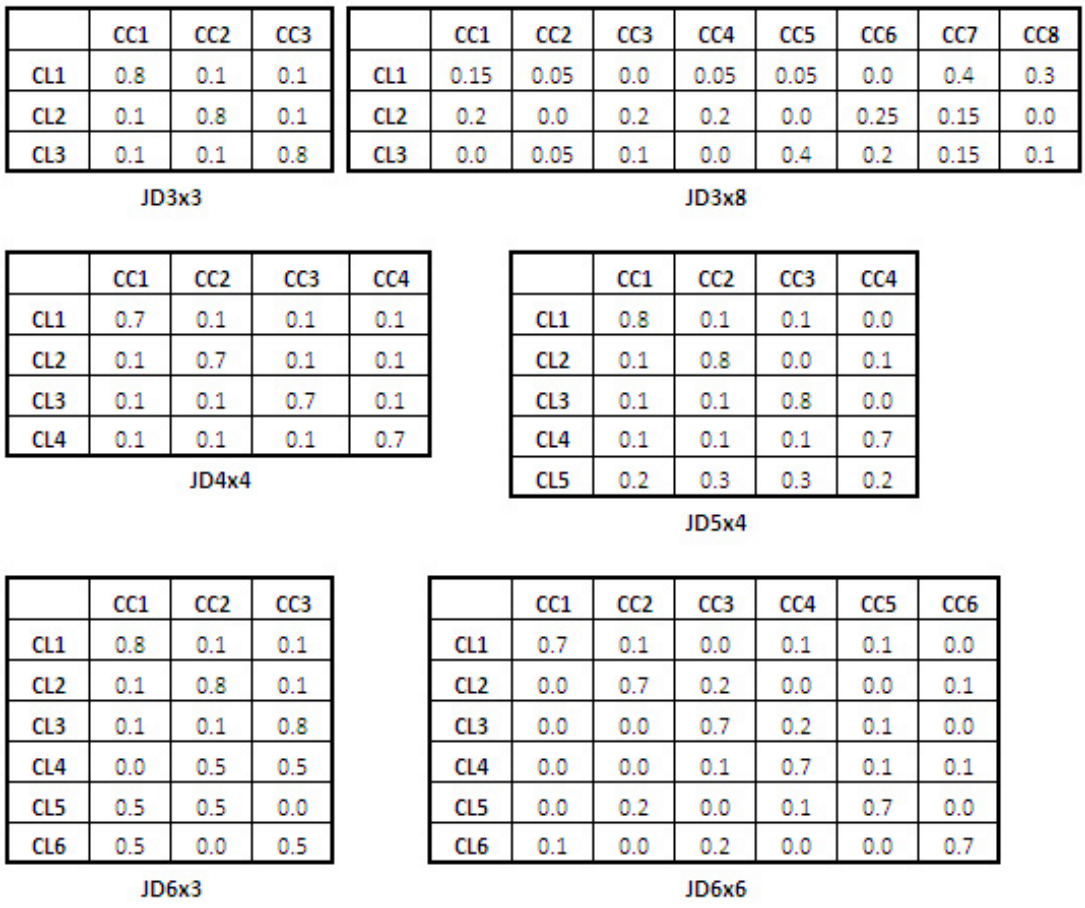
	Nombre de classes sur les lignes et les colonnes	Nombre minimum d'occurrences	Nombre maximum d'occurrences
JD3x3	3x3	34	150
JD4x4	4x4	25	150
JD5x4	5x4	25	150
JD6x3	6x3	34	150
JD3x8	3x8	13	150
JD6x6	6x6	17	150

Chaque ligne correspond à un terme et chaque colonne à une page. Le croisement d'une ligne et d'une colonne est le nombre d'occurrences de chaque terme (ligne) dans chaque page (colonne). Le nombre minimal d'occurrences est calculé en fonction du nombre de classes de pages et du nombre total de pages. En effet, un terme caractérisant une classe de pages doit apparaître au moins une fois dans chacune des pages de la classe. Ainsi, le nombre minimal d'occurrences est égal au nombre de pages dans la classe. Dans le cas où l'effectif des pages dans chaque

classe est le même, le nombre minimal d'occurrences serait le rapport entre le nombre total de pages M et le nombre de classes sur les colonnes L.

$$NbOccMin = M/L$$

Les sites Web sont généralement composés de pages dont le contenu est bien organisé. Chaque ensemble de pages du site traite d'un sujet particulier mais en liaison avec la thématique du site. Les classes de pages, comme les classes des termes les plus descriptifs du contenu des pages, sont généralement séparées.



	CC1	CC2	CC3
CL1	0.8	0.1	0.1
CL2	0.1	0.8	0.1
CL3	0.1	0.1	0.8

JD3x3

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8
CL1	0.15	0.05	0.0	0.05	0.05	0.0	0.4	0.3
CL2	0.2	0.0	0.2	0.2	0.0	0.25	0.15	0.0
CL3	0.0	0.05	0.1	0.0	0.4	0.2	0.15	0.1

JD3x8

	CC1	CC2	CC3	CC4
CL1	0.7	0.1	0.1	0.1
CL2	0.1	0.7	0.1	0.1
CL3	0.1	0.1	0.7	0.1
CL4	0.1	0.1	0.1	0.7

JD4x4

	CC1	CC2	CC3	CC4
CL1	0.8	0.1	0.1	0.0
CL2	0.1	0.8	0.0	0.1
CL3	0.1	0.1	0.8	0.0
CL4	0.1	0.1	0.1	0.7
CL5	0.2	0.3	0.3	0.2

JD5x4

	CC1	CC2	CC3
CL1	0.8	0.1	0.1
CL2	0.1	0.8	0.1
CL3	0.1	0.1	0.8
CL4	0.0	0.5	0.5
CL5	0.5	0.5	0.0
CL6	0.5	0.0	0.5

JD6x3

	CC1	CC2	CC3	CC4	CC5	CC6
CL1	0.7	0.1	0.0	0.1	0.1	0.0
CL2	0.0	0.7	0.2	0.0	0.0	0.1
CL3	0.0	0.0	0.7	0.2	0.1	0.0
CL4	0.0	0.0	0.1	0.7	0.1	0.1
CL5	0.0	0.2	0.0	0.1	0.7	0.0
CL6	0.1	0.0	0.2	0.0	0.0	0.7

JD6x6

FIGURE 4.1 – Prototypes initiaux des jeux des données

Les tableaux de la figure (fig. 4.1) présentent les prototypes initiaux (noyaux) utilisés pour la génération des classes bien séparées sur les lignes et les colonnes. Une projection des classes-lignes et des classes-colonnes du jeu des données JD6x3 sur le premier plan factoriel d'une ACP est présentée sur la figure 4.2.

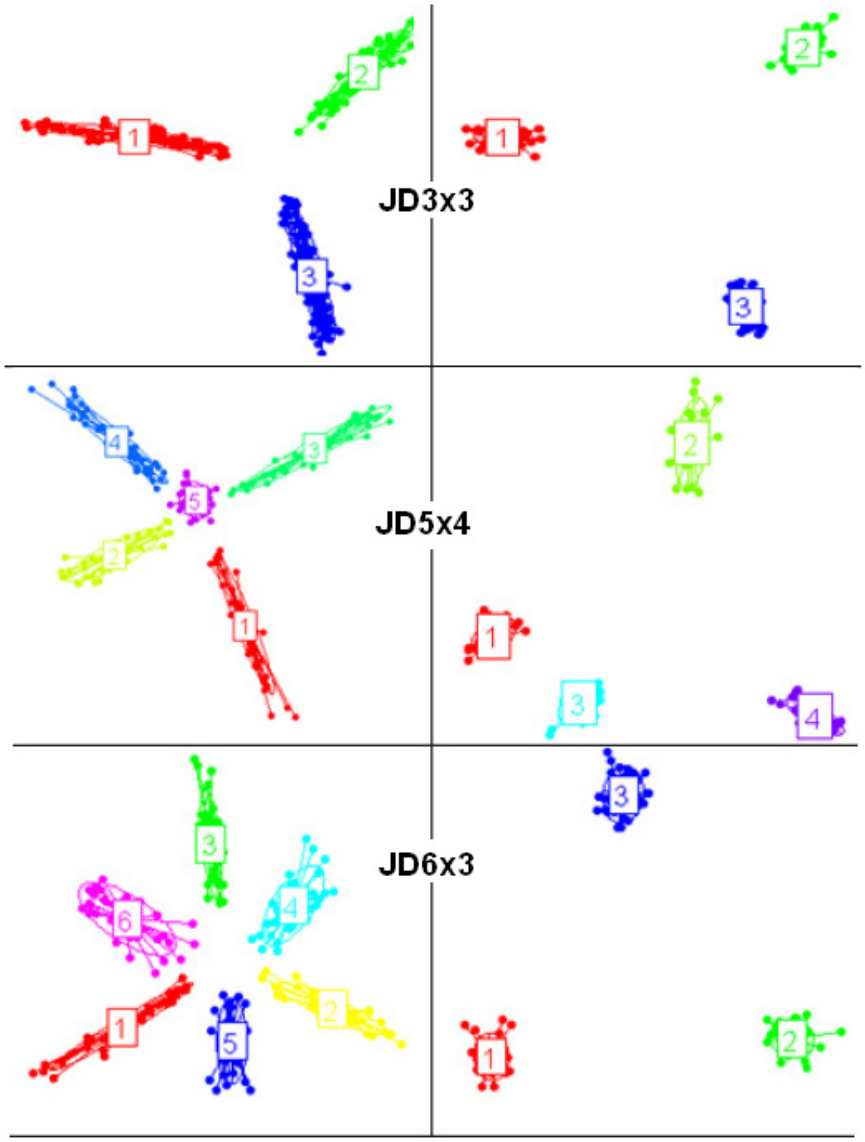


FIGURE 4.2 – Projection des biclasses des jeux de données JD3x3, JD5x4 et JD6x3 sur le premier plan factoriel d'une ACP. À gauche c'est la projection des classes-lignes et à droite c'est la projection des classes-colonnes.

4.3 Algorithme Croki2

Reprenons les notations du chapitre précédent. Soit A un tableau de contingence à N lignes et M colonnes, défini par l'ensemble $\{X = x_1, \ldots, x_N\}$ des lignes et l'ensemble $\{Y = y_1, \ldots, y_M\}$ des colonnes.
a_{ij}, $1 \leq i \leq N$ et $1 \leq j \leq M$, sont les éléments du tableau A.

TABLE 4.2 – Tableau de contingence initial

	y_1	...	y_i	...	y_M
x_1	a_{11}	...	a_{1j}	...	a_{1M}
...	...	...	...	...	...
x_i	a_{i1}	...	a_{ij}	...	a_{iM}
...	...	...	...	...	...
x_N	a_{N1}	...	a_{Nj}	...	a_{NM}

Le χ^2 du tableau de contingence initial s'écrit :

$$\chi^2(X,Y) = S \sum_{i \in X} \sum_{j \in Y} \frac{(f_{ij} - f_{i.} f_{.j})^2}{f_{i.} f_{.j}}$$

avec

$$S = \sum_{i \in X} \sum_{j \in Y} a_{ij}$$

$$f_{ij} = \frac{a_{ij}}{S}, f_{i.} = \sum_{j \in Y} f_{ij}, f_{.j} = \sum_{i \in X} f_{ij}$$

4.3.1 Principe général de l'algorithme Croki2

L'algorithme Croki2 consiste à trouver une partition $P = (P_1, ..., P_K)$ de X en K classes et une partition $Q = (Q_1, ..., Q_L)$ de Y en L classes telles que le χ^2 de contingence du nouveau tableau de données $T(P, Q)$ soit maximum. À partir d'un couple initial (P^0, Q^0), une suite de couples de partitions (P^n,Q^n) optimisant le χ^2 de contingence est construite.

Le critère général à optimiser est :

$$\chi^2(P,Q) = \sum_{k=1}^{K} \sum_{l=1}^{L} \frac{(f_{kl} - f_{k.} f_{.l})^2}{f_{k.} f_{.l}}$$

avec

$$f_{kl} = \sum_{i \in P_k} \sum_{j \in Q_l} f_{ij}$$

$$f_{k.} = \sum_{l=1,L} f_{kl} = \sum_{i \in P_k} f_{i.}$$

$$f_{.l} = \sum_{k=1,K} f_{kl} = \sum_{j \in Q_l} f_{.j}$$

$$S = \sum_{i=1}^{N} \sum_{j=1}^{M} a_{ij}$$

TABLE 4.3 – Nouveau tableau de contingence construit en regroupant les lignes et les colonnes suivant les partitions P et Q.

	Q_1	...	Q_L
P_1	...	...	...
...	...	g_{kl}	...
P_K	...	...	...

L'élément général du tableau est $g_{kl} = \sum_{i \in P_k} \sum_{j \in Q_l} a_{ij}$.

4.3.2 Déroulement de l'algorithme Croki2

L'algorithme Croki2 se déroule comme suit :

Pour chaque tirage aléatoire des partitions initiales (P^0, Q^0)
1 Démarrer de la position initiale $(P^{(0)}, Q^{(0)}, G^{(0)})$
2 Calculer $(P^{(n+1)}, Q^{(n+1)}, G^{(n+1)})$ à partir $(P^{(n)}, Q^{(n)}, G^{(n)})$
3 Calculer $(P^{(n+1)}, Q^{(n)}, G^{(n+\frac{1}{2})})$ à partir de $(P^{(n)}, Q^{(n)}, G^{(n)})$
4 Calculer $(P^{(n+1)}, Q^{(n+1)}, G^{(n+1)})$ à partir de $(P^{(n+1)}, Q^{(n)}, G^{(n+\frac{1}{2})})$
5 Recommencer l'étape 2 jusqu'à la convergence de l'algorithme

La recherche des deux partitions (P, Q) sur les lignes et les colonnes peut se faire selon plusieurs stratégies. La première stratégie, proposée par [Gov83] consiste à alterner l'optimisation sur les lignes puis sur les colonnes en utilisant l'algorithme des nuées dynamiques jusqu'à la convergence. Une autre stratégie annoncée dans la page 72 de [Gov83] consiste en une étape de représentation suivie de deux étapes d'affectation répétées plusieurs fois jusqu'à la convergence. Dans le cadre de cette thèse, nous proposons d'effectuer une optimisation globale sur les lignes et les colonnes en effectuant une étape de représentation suivie d'une étape d'affectation sur les lignes puis sur les colonnes plusieurs fois jusqu'à la convergence. L'avantage de l'optimisation globale par rapport à l'optimisation alternée est qu'elle accorde la même importance aux deux ensembles des données puisqu'elle procède par simple affectation. Par contre, l'optimisation alternée privilégie le premier ensemble découpé par rapport au second puisque le traitement sur le deuxième ensemble ne débute qu'après optimisation du partitionnement

du premier. Dans les sections 4.3.4, 4.3.5 et 4.4 nous présentons en détail l'algorithme Croki2 avec optimisation alternée et l'algorithme Croki2 avec optimisation globale et nous comparons leurs résultats sur des données simulées.

4.3.3 Étapes de représentation et d'affectation

Les étapes de représentation et d'affectation sont définies comme suit :

- Étape d'affectation des lignes : consiste à calculer pour chaque objet i de X l'indice $k*$ de la classe d'affectation qui vérifie

$$k^* = argmin_{k=1...K} d_{\chi^2}(u_i, G_k)$$

- Étape de représentation des lignes : consiste à calculer pour chaque classe k le prototype $G_k = (g_{k1}, ..., g_{kl}, ..., g_{kL})$

$$g_{kl} = \sum_{i \in P_k} a_{il} = \sum_{i \in P_k} \sum_{j \in Q_l} a_{ij}$$

- Étape d'affectation des colonnes : consiste à calculer pour chaque objet j de Y l'indice l^* de la classe d'affectation qui vérifie

$$l^* = argmin_{l=1...L} d_{\chi^2}(v_j, G_l)$$

- Étape de représentation des colonnes : consiste à calculer pour chaque classe l le prototype $G_l = (g_{1l}, ..., g_{kl}, ...g_{Kl})$

$$g_{kl} = \sum_{j \in Q_l} a_{kj} = \sum_{j \in Q_l} \sum_{i \in P_k} a_{ij}$$

Sachant que l'algorithme Croki2 débute avec deux partitions initiales tirées au hasard sur les lignes et les colonnes $({}^{}P^0, Q^0)$, les résultats obtenus, comme pour toutes les méthodes convergeant vers un optimum local, dépendent de ces partitions initiales. Il est donc nécessaire d'exécuter plusieurs fois l'algorithme afin d'assurer l'indépendance du résultat final des partitions initiales.

4.3.3.1 Paramètres de l'algorithme

Les paramètres en entrée de l'algorithme sont le nombre de classes sur les lignes, K, le nombre de classes sur les colonnes, L, et le nombre d'exécutions ou tirages, n. Le problème d'identification du nombre de classes sur les lignes et les colonnes est traité dans le chapitre suivant.
Nous nous concentrons dans cette section sur l'estimation du nombre de tirages nécessaires pour aboutir à la solution optimale. En effet, en effectuant n tirages,

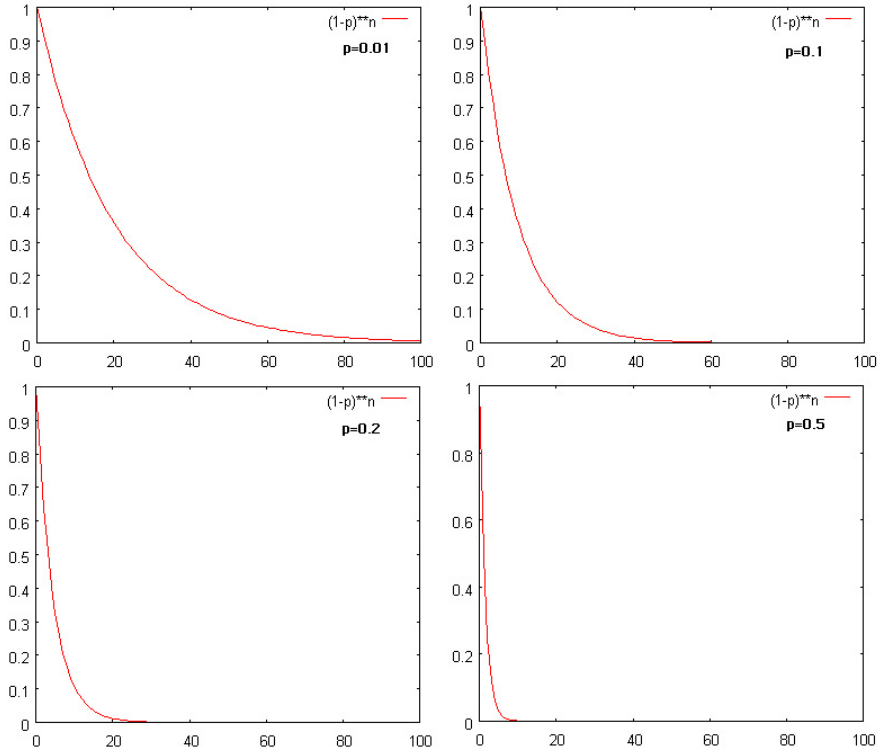


FIGURE 4.3 – Probabilité d'échec à trouver la solution optimale en fonction du nombre d'exécutions.

n résultats sont obtenus. En se basant sur le critère de maximisation du χ^2, la solution optimale correspond au couple de partitions qui fournit la plus grande valeur du critère.

Soient p la probabilité d'obtenir la solution optimale et n le nombre de tirages. L'obtention de la solution optimale au bout de n exécutions suit une loi binomiale de paramètres n et p. Soit X la variable aléatoire correspondant au nombre de succès obtenus à l'issue des n tirages indépendants.
La variable aléatoire suit une loi de probabilité définie par :

$$p(k) = P(X = k) = C_n^k p^k (1-p)^{n-k}$$

C_n^k est le nombre de combinaisons de k éléments parmi n. k prend la valeur 0 si la solution optimale n'est pas obtenue au bout de n tirages,

$$p(0) = P(X = 0) = (1-p)^n$$

$p(0)$ décroit avec n. Ainsi, au bout d'un certain nombre d'exécutions, on est sûr d'aboutir à la solution optimale, celle qui optimise le critère de χ^2 et qui ne

change pas en modifiant le tirage de départ. La figure 4.3 représente la fonction $(1-p)^n$ pour différentes valeurs de p.

Expérimentations sur le nombre de tirages

Dans le but d'estimer le nombre minimum d'exécutions nécessaires pour avoir des résultats indépendants de l'état initial, nous proposons de comparer la solution optimale obtenue pour $n = 10$, $n = 20$, $n = 50$ et $n = 100$ pour le jeu des données JD3x3 en faisant varier le nombre de classes sur les lignes et les colonnes (Tab. 4.4) puis pour tous les jeux des données sachant que le nombre de classes sur les lignes et les colonnes correspond au nombre correct (Tab. 4.5).

TABLE 4.4 – χ^2 optimal en fonction du nombre d'exécutions pour le jeu des données JD3x3.

	10	20	50	60	100
(2,2)	**9010.667**	**9010.667**	**9010.667**	**9010.667**	**9010.667**
(2,3)	*9027.720*	*9027.720*	**9027.799**	**9027.799**	**9027.799**
(2,4)	**9033.147**	**9033.147**	**9033.147**	**9033.147**	**9033.147**
(2,5)	*9034.707*	*9034.707*	*9034.714*	**9034.717**	**9034.717**
(3,2)	*9039.613*	*9039.613*	*9039.613*	**9039.614**	**9039.614**
(3,3)	**17876.650**	**17876.650**	**17876.650**	**17876.650**	**17876.650**
(3,4)	**17898.200**	**17898.200**	**17898.200**	**17898.200**	**17898.200**
(3,5)	*17913.950*	**17916.980**	**17916.980**	**17916.980**	**17916.980**
(4,2)	**9046.359**	**9046.359**	**9046.359**	**9046.359**	**9046.359**
(4,3)	*17904.470*	**17915.940**	**17915.940**	**17915.940**	**17915.940**
(4,4)	**18102.220**	**18102.220**	**18102.220**	**18102.220**	**18102.220**
(4,5)	**18120.350**	**18120.350**	**18120.350**	**18120.350**	**18120.350**
(5,2)	*9049.947*	*9049.947*	**9084.129**	**9084.129**	**9084.129**
(5,3)	**17943.840**	**17943.840**	**17943.840**	**17943.840**	**17943.840**
(5,4)	**18119.950**	**18119.950**	**18119.950**	**18119.950**	**18119.950**
(5,5)	**18152.940**	**18152.940**	**18152.940**	**18152.940**	**18152.940**

D'après le tableau 4.4, pour la majorité des couples il suffit de tirer 10 fois pour avoir la solution optimale. Cependant, pour certains couples, tels que (2,5) et (3,2) il faut au moins 60 exécutions pour aboutir à la solution optimale. Dans le cas particulier où le nombre de classes en entrée est le nombre correct sur les lignes et les colonnes, au bout de 10 exécutions nous aboutissons à la solution optimale pour tous les jeux des données (Tab. 4.5).
Dans le cas des données réelles, il est préférable d'augmenter le nombre d'exécutions afin d'augmenter les chances d'aboutir à la bonne solution. Dans notre cas, nous fixons n à 100 pour tous nos jeux des données.

TABLE 4.5 – χ^2 optimal en fonction du nombre d'exécutions pour chaque jeu des données.

n	3x3 K=3, L=3	4x4 K=4, L=4	5x4 K=5, L=4	6x3 K=6, L=3	6x6 K=6, L=6	3x8 K=3, L=8
10	17876.65	18524.62	20889.55	13576.02	35727.76	10911.86
20	17876.65	18524.62	20889.55	13576.02	35727.76	10911.86
50	17876.65	18524.62	20889.55	13576.02	35727.76	10911.86
100	17876.65	18524.62	20889.55	13576.02	35727.76	10911.86

4.3.3.2 Evaluation des partitions générées par Croki2

L'objectif de cette section est de présenter les critères d'évaluation des partitions générées par l'algorithme Croki2. Ces critères proposés par [Gov83] nous permettent de choisir les biclasses les plus significatives dans notre analyse des données réelles dans le chapitre suivant.

- **Pertinence de la biclasse :**
  La pertinence PB de la biclasse est mesurée à travers la part de l'inertie conservée par la biclasse, notée W_{kl}, dans l'inertie totale W.

$$PB = W_{kl}/W$$

avec

$$W_{kl} = f_{k.}f_{.l}(\frac{f_{kl}}{f_{k.}f_{.l}} - 1)^2$$

et

$$W = \sum_{k,l} W_{kl}$$

- **Homogénéité de la biclasse :**
  L'homogénéité, HB, de la biclasse est mesurée par la part d'inertie W_{kl}, conservée par la biclasse par rapport à l'inertie initiale T_{kl} des points de la classe. Cette valeur indique la qualité de représentation d'une biclasse. La valeur obtenue comprise entre 0 et 1 est d'autant plus grande que la biclasse est homogène.

$$HB = (W_{kl}/T_{kl})$$

avec

$$W_{kl} = f_{k.}f_{.l}(\frac{f_{kl}}{f_{k.}f_{.l}} - 1)^2$$

et

$$T_{kl} = \sum_{i \in P_k} \sum_{j \in Q_l} f_{i.}f_{.j}(\frac{f_{ij}}{f_{i.}f_{.j}} - 1)^2$$

4.3.4 Croki2 avec optimisation alternée

La méthode utilisée dans l'algorithme Croki2 [Gov83] est l'optimisation alternée de la partition en lignes et la partition en colonnes. Pour obtenir ces partitions, on utilise une variante des nuées dynamiques, en prenant la métrique du χ^2 et le centre de gravité comme noyau. L'algorithme de Croki2 avec optimisation alternée (Croki2 OA) se déroule suivant les deux étapes suivantes :

- **Optimisation sur les lignes**
  On bloque la partition en colonnes Q et on ne travaille que sur la partition en lignes P. Soit le tableau de contingence $A(X, Q)$ défini par :

$$A_1(x, l) = \sum_{j \in Q_l} a_{ij}$$

  où $Q = (Q_1, ..., Q_L)$. En considérant que les objets à classer sont les éléments de X et les variables les classes de Q, une suite de partitions de X en L classes est construite telle que les χ^2 de contingence associés soient croissants jusqu'à la convergence. Le critère à optimiser avec l'algorithme des nuées dynamiques simple est :

$$\Delta(P/Q, G) = \sum_{k=1}^{K} \sum_{i \in P_k} d_{\chi^2}(u_i, G_k)$$

- **Optimisation sur les colonnes**
  Contrairement à l'étape précédente, on fixe la partition en lignes P et on applique l'algorithme des nuées dynamiques sur la partition en colonnes Y. Le tableau de contingence est cette fois $A(P, Y)$ défini par :

$$A_1(k, y) = \sum_{i \in P_k} a_{ij}$$

  où $P = (P_1, ..., P_K)$. Une suite de partitions de Y en L classes fait croître le χ^2 jusqu'à la convergence. Cette fois, on cherche à optimiser le critère suivant :

$$\Delta(Q/P, G) = \sum_{l=1}^{L} \sum_{j \in Q_l} d_{\chi^2}(v_j, G_l)$$

4.3.4.1 Problème d'optimisation

L'algorithme Croki2 recherche alternativement une partition de X et une partition de Y en appliquant à chaque fois l'algorithme des nuées dynamiques sur une partition en fixant l'autre et vice versa. Sur le tableau de contingence $A(X, Q)$,

plusieurs itérations sont nécessaires pour atteindre la convergence de l'algorithme des nuées dynamiques. Une suite de partitions de X, $H^0, ..., H^r$, en K classes est construite telle que les χ^2 de contingence associés soient croissants jusqu'à la convergence.

$$\chi^2(H^0, Q^n) < \chi^2(H^1, Q^n) < ... < \chi^2(H^r, Q^n) = \chi^2(H^{r+1}, Q^n)$$

$$\chi^2(H^r, Q^n) = \Delta(H^r/Q^n, G) = \Delta(Q^n/H^r, G)$$

où

$$G(k, l) = \sum_{j \in Q_l^n} \sum_{i \in H_k^r} a_{ij}$$

r est l'étape à partir de laquelle la suite devient stationnaire. La partition P^{n+1} sera la partition obtenue à la convergence : $P^{n+1} = H^r$. De même pour le tableau de contingence $A(P, Y)$, plusieurs itérations sont nécessaires pour atteindre la convergence de l'algorithme des nuées dynamiques. À un niveau supérieur, une autre boucle d'itérations sert à alterner l'optimisation sur les lignes et l'optimisation sur les colonnes jusqu'à ce que l'ensemble des deux partitions soit stable. Ainsi, pour chaque tirage, l'algorithme Croki2 OA est itératif à deux niveaux.

4.3.4.2 Algorithme Croki2 OA

```
Pour chaque tirage aléatoire des partitions initiales
    Démarrer d'une position initiale (P^0, Q^0, G^0)
        Pour iter < iterMax
            Optimisation sur les lignes
                Etape de représentation
                Etape d'affectation par les nuées dynamiques
            Optimisation sur les colonnes
                Etape de représentation
                Etape d'affectation par les nuées dynamiques
        FinPour
FinPour
```

4.3.5 Croki2 avec optimisation globale

L'algorithme Croki2 avec optimisation globale (Croki2 OG) consiste à combiner les deux étapes d'optimisation utilisées dans la version originale de Croki2 en une seule étape. L'algorithme des nuées dynamiques est remplacé par une simple

affectation de l'individu à la classe la plus proche. Une boucle d'itérations permet d'alterner les étapes d'affectation et de représentation sur les lignes et les colonnes plusieurs fois jusqu'à la convergence. La convergence est atteinte lorsqu'aucun individu sur les lignes et sur les colonnes ne change de classe d'appartenance.

4.3.5.1 Algorithme Croki2 OG

L'algorithme suivant explicite le déroulement de l'algorithme Croki2 avec optimisation globale.

```
1 Pour chaque tirage aléatoire des partitions initiales
2       Démarrer d'une position initiale (P^0, Q^0, G^0)
3               Optimisation globale
5                       Etape d'affectation des lignes
6                       Etape de représentation des lignes
8                       Etape d'affectation des colonnes
9                       Etape de représentation des colonnes
10 FinPour
```

4.3.5.2 Peut-on améliorer la solution de Croki2 OA par Croki2 OG, et vice-versa ?

Disposant des deux algorithmes Croki2 OA et Croki2 OG, la question qu'on se pose : est-il possible d'améliorer une solution obtenue par Croki2 OA par l'algorithme Croki2 OG, et inversement ? Soit Q une partition de Y en L classes, G le prototype correspondant à cette partition. Soit la fonction f définie telle que $P = f(Q, G)$ est la partition de X en K classes minimisant

$$\Delta(P/Q, G) = \sum_{k=1}^{K} \sum_{i \in P_k} d_{\chi^2}(u_i, G_k)$$

À la convergence de l'algorithme Croki2 OA, chaque individu i est affecté à la classe la plus proche. Soit k_i la classe à laquelle appartient i. Par suite,

$$\Delta(P^{(r)}/Q, G) = \sum_{k=1}^{K} \sum_{i \in P_k} d_{\chi^2}(u_i, G_{k_i})$$

À partir de la partition P^r, on veut calculer P^{r+1} en utilisant l'algorithme Croki2 OG.

$$\Delta(P^{(r+1)}/Q, G) = \sum_{k=1}^{K} \sum_{i \in P_k} min_k d_{\chi^2}(u_i, G_k)$$

Comme l'algorithme Croki2 OG procède par simple affectation de l'individu i à la classe k^* la plus proche selon la distance de χ^2,

$$min_k d_{\chi^2}(u_i, G_k) = d_{\chi^2}(u_i, G_{k_i})$$

d'où,

$$\Delta(P^{(r+1)}/Q, G) = \Delta(P^{(r)}/Q, G)$$

La solution obtenue par Croki2 OA ne peut pas être améliorée par Croki2 OG car les deux étapes d'affectation de ces deux algorithmes sont identiques.

4.4 Comparaison des deux algorithmes

Nous avons démontré dans le paragraphe précédent que l'ensemble des solutions de Croki2 OA et Croki2 OG est le même. Cependant, pour une initialisation fixée, la solution obtenue par Croki2 OA est très souvent différentes de celle obtenue par Croki2 OG. Pour cette raison, nous proposons dans cette section de comparer les deux variantes de l'algorithme Croki2. Cette comparaison repose sur plusieurs critères, à savoir le χ^2 optimal à la convergence, la complexité des deux algorithmes, les partitions obtenues par les deux algorithmes à la convergence, la capacité des deux algorithmes à identifier les bonnes partitions sur les deux ensembles des données et à trouver des bipartitions avec des classes non vides. Nous appliquons l'algorithme Croki2 avec optimisation alternée et l'algorithme Croki2 avec optimisation globale sur les différents jeux des données simulées en faisant varier le nombre de classes sur les lignes et les colonnes. Afin de comparer les deux algorithmes, nous proposons d'utiliser des indices de validation externe tels que la F-mesure et l'indice de Rand corrigé.

4.4.1 Indices de validation externe

La validation externe permet de comparer les résultats d'une classification automatique à une information connue *a priori* sur la structure des données. Elle consiste généralement à comparer deux partitions obtenues sur un même ensemble d'individus. Cette comparaison est basée sur la recherche des correspondances entre les classes formant les deux partitions. Les deux indices les plus répandus sont l'indice de Rand [Ran71] et l'indice de Rand corrigé [HA85] utilisés pour une analyse globale basée sur la comparaison de deux partitions et la F-mesure [Rij79] permettant d'associer à chaque classe *a priori* une classe dans l'ensemble de classes produites par l'algorithme de classification.

Soit le tableau de contingence suivant (Tab.4.6) croisant les deux partitions P et Q résultantes de deux classifications sur un même ensemble d'individus

de dimension n. La partition P est considérée comme la partition de référence. Chaque cellule du tableau correspond au nombre d'individus affectés à la fois à la classe C_i^P de la partition P et la classe C_j^Q de la partition Q.

TABLE 4.6 – Tableau de contingence entre deux partitions

	C_1^Q	...	C_j^Q	...	C_q^Q	
C_1^P	n_{11}	...	n_{1j}	...	n_{1q}	$n_{1.}$
...						
C_i^P	n_{i1}	...	n_{ij}	...	n_{iq}	$n_{i.}$
...						
C_p^P	n_{p1}	...	n_{pj}	...	n_{pq}	$n_{p.}$
	$n_{.1}$	...	$n_{.j}$	...	$n_{.q}$	$n_{..} = n$

4.4.1.1 F-mesure

La F-mesure [Rij79] est la moyenne harmonique entre *Rappel* et *Précision* définis comme suit.
À partir du tableau de contingence Tab.4.6, le *Rappel* est défini comme le rapport entre le nombre d'individus en commun entre deux clusters de différentes partitions et le total des individus dans le cluster de la partition P. La *Précision* est définie par le rapport entre le nombre d'individus en commun entre deux clusters de différentes partitions et le total des individus dans le cluster de la partition Q.

$$Rappel(C_i^P, C_j^Q) = \frac{n_{ij}}{\sum_{j=1}^{q} n_{ij}} = \frac{n_{ij}}{n_{i.}}$$

$$Precision(C_i^P, C_j^Q) = \frac{n_{ij}}{\sum_{j=1}^{p} n_{ij}} = \frac{n_{ij}}{n_{.j}}$$

$$F - mesure(C_i^P, C_j^Q) = \frac{2Precision(C_i, C_j)Rappel(C_i, C_j)}{Precision(C_i, C_j) + Rappel(C_i, C_j)}$$

La F-mesure cherche le cluster A' dans la partition Q qui correspond le plus au cluster A de la partition P. Dans le cas où $Rappel = Precision$, $A = A'$. Plus les valeurs de *Rappel* et *Precision* sont élevées, plus les deux clusters sont semblables. Au niveau de la partition, la F-mesure fournit une évaluation numérique de la similarité entre deux partitions P et Q issues d'un même ensemble d'individus. Elle est calculée à partir des moyennes des valeurs maximales sur les colonnes du tableau de contingence 4.6.

$$F(P, Q) = \frac{1}{n} \sum_{i=1}^{p} n_{i.} max_{j=1...q} F - mesure(C_i^P, C_j^Q)$$

La F-mesure prend ses valeurs dans l'intervalle [0,1]. Plus la valeur de la F-mesure est élevée plus les deux partitions sont similaires. Cependant, la F-mesure n'est pas un indice symétrique. En effet, $F(P,Q) \neq F(Q,P)$.

4.4.1.2 Indice de Rand

L'indice de Rand mesure la concordance entre les deux partitions P et Q. Il est calculé par la formule suivante :

$$Rand(P,Q) = \frac{a+d}{a+b+c+d}$$

avec

- a est le nombre de paires d'individus affectés à une même classe de P et une même classe de Q.

$$a = \sum_{i=1}^{p}\sum_{j=1}^{q}\binom{n_{ij}}{2}$$

$\binom{n_{ij}}{2}$ représente le nombre de paires d'individus dans $C_i^P \cap C_j^Q$

$$\binom{n_{ij}}{2} = \frac{n_{ij}(n_{ij}-1)}{2}$$

- b représente le nombre de paires d'individus classés ensemble selon P et dans deux classes différentes selon Q.

$$b = \sum_{i=1}^{p}\binom{n_{i.}}{2} - \sum_{i=1}^{p}\sum_{j=1}^{q}\binom{n_{ij}}{2}$$

$\binom{n_{i.}}{2}$ représente le nombre de paires d'individus de la classe C_i^P

$$\binom{n_{i.}}{2} = \frac{n_{i.}(n_{i.}-1)}{2}$$

- c représente le nombre de paires d'individus classés ensemble selon Q et dans deux classes différentes selon P.

$$c = \sum_{j=1}^{q}\binom{n_{.j}}{2} - \sum_{i=1}^{p}\sum_{j=1}^{q}\binom{n_{ij}}{2}$$

$\binom{n_{.j}}{2}$ représente le nombre de paires d'individus de la classe C_j^Q

$$\binom{n_{.j}}{2} = \frac{n_{.j}(n_{.j}-1)}{2}$$

– d représente le nombre de paires d'individus classés dans deux classes différentes selon P et selon Q.

$$d = \binom{n}{2} + \sum_{i=1}^{p}\sum_{j=1}^{q}\binom{n_{ij}}{2} - \sum_{i=1}^{p}\binom{n_{i.}}{2} - \sum_{j=1}^{q}\binom{n_{.j}}{2}$$

$\binom{n}{2}$ représente le nombre total de paires d'individus dans l'ensemble des données.

$$\binom{n}{2} = \frac{n(n-1)}{2}$$

$a+d$ représente le nombre d'accords et $b+c$ représente le nombre de désaccords entre les deux partitions. L'indice de Rand prend ses valeurs dans l'intervalle [0,1]. Il est égal à 1 si les deux partitions sont identiques. Pour deux partitions aléatoires, la valeur espérée de l'indice de Rand n'est pas nulle. L'indice de Rand ajusté proposé par [HA85] correspond à l'indice de Rand auquel on apporte une correction pour le manque de valeur constante de Rand quand les partitions sont sélectionnées au hasard. Ainsi, l'indice de Rand ajusté a pour forme générale :

$$RC = \frac{(IndicedeRand - IndiceEspere)}{(IndiceMaximum - IndiceEspere)}$$

Cet indice qui peut être au plus égal à 1, prend donc la valeur 0 quand l'indice de Rand = indice espéré. L'indice de Rand brut est souvent plus élevé que celui corrigé.

$$RC(P_1,P_2) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{k}\binom{n_{ij}}{2} - \binom{n}{2}^{-1}\sum_{i=1}^{m}\binom{n_{i.}}{2}\sum_{j=1}^{k}\binom{n_{.j}}{2}}{\frac{1}{2}[\sum_{i=1}^{m}\binom{n_{i.}}{2} + \sum_{j=1}^{k}\binom{n_{.j}}{2}] - \binom{n}{2}^{-1}\sum_{i=1}^{m}\binom{n_{i.}}{2}\sum_{j=1}^{k}\binom{n_{.j}}{2}}$$

L'espérance de l'indice corrigé est nulle lorsque les accords entre les deux partitions sont dus au hasard [CLP01] [MC86] [YS04]. Cependant, cet indice corrigé peut prendre des valeurs négatives lorsque les partitions sont peu liées. Les valeurs proches de 1 correspondent à des partitions très semblables.

4.4.2 Comparaison du χ^2

Dans l'algorithme Croki2, un bon bipartitionnement est obtenu pour des valeurs élevées de χ^2 reflétant une corrélation maximale entre la partition-ligne et la partition-colonne.
Notre idée est de comparer les valeurs de χ^2 obtenues à la convergence par les deux algorithmes Croki2 OA (Croki2 avec Optimisation Alternée) et Croki2 OG (Croki2 avec Optimisation Globale) sachant que les deux algorithmes sont initialisés de la même manière et que la même stratégie de tirage aléatoire est appliquée dans les deux.
Nous proposons alors de calculer la différence entre les valeurs de χ^2 obtenues par Croki2 OA et celles obtenues par Croki2 OG.

$$Diff = \frac{2 \times |\chi^2_{Croki2OA} - \chi^2_{Croki2OG}|}{(\chi^2_{Croki2OA} + \chi^2_{Croki2OG})}$$

Les tableaux 4.7 et 4.8 présentent les valeurs de χ^2 obtenues en appliquant respectivement les algorithmes Croki2 OA et Croki2 OG sur les jeux des données JD4x4 et JD6x3. La ligne supérieure de chaque case concerne le Croki2 OA. La ligne inférieure concerne le Croki2 OG.

TABLE 4.7 – Valeus de χ^2 résultant de l'application de Croki2 OA et Croki2 OG sur JD4x4.

χ^2	**2**	**3**	**4**	**5**	**6**	**7**
2	6399.867	6423.867	6428.962	6431.248	*6432.274*	*6432.779*
	6399.867	6423.867	6428.962	6431.248	*6432.415*	*6432.816*
3	6443.349	12523.73	12540.15	12556.73	*12554.19*	*12399.67*
	6443.349	12523.73	12540.15	12556.21	*12565.88*	*12570.78*
4	*6485.393*	*12562.58*	18524.62	18541.99	*18556.23*	*18549.76*
	6456.744	*12566.13*	18524.62	18541.55	*18551.79*	*18561.15*
5	*6498.821*	*12604.97*	18567.38	*18736.56*	*18771.35*	*18707.02*
	6461.974	*12604.97*	18567.38	*18731.61*	*18760.36*	*18783.97*
6	*6504.051*	*12617.69*	18608.72	*18744.97*	*18858.87*	*18914.35*
	6465.539	*12629.08*	18608.72	*18752.47*	*18814.23*	*18774.45*
7	*6507.491*	*12641.45*	*18649.94*	*18816.61*	*18913.88*	*18936.82*
	6507.517	*12661.48*	*18649.83*	*18779.11*	*18870.63*	*18868.88*

La représentation graphique de la différence entre les valeurs de χ^2 (fig.4.4 et fig. 4.5) montre que cette différence n'est pas significative lorsque le nombre de classes sur les lignes ou les colonnes est faible. Cette différence est nulle lorsque

TABLE 4.8 – Valeurs de χ^2 résultant de l'application de Croki2 OA et Croki2 OG sur JD6x3

χ^2	2	3	4	5	6
2	6055.661	6095.293	6106.738	6114.162	*6119.194*
	6055.661	6095.293	6106.738	6112.396	*6114.754*
3	6862.216	10186.25	10218.9	10241.32	*10261.02*
	6862.216	10186.25	10218.9	10241.32	*10261.15*
4	6960.637	11386.11	11436.89	*11481.15*	*11503.92*
	6960.637	11386.11	11436.89	*11453.64*	*11503.74*
5	6989.64	12501.52	12567.18	*12616.78*	*12631.38*
	6989.64	12501.52	12567.18	*12618.18*	*12645.5*
6	7002.237	13576.02	13641.71	*13692.72*	*13733.23*
	7002.237	13576.02	13641.71	*13692.16*	*13730.89*
7	*7020.455*	13608.32	*13733.77*	*13794.98*	*13837.09*
	7008.095	13608.32	*13712.37*	*13780.24*	*13801.62*
8	*7024.368*	*13636.89*	*13743.85*	*13799.83*	*13824.03*
	7014.964	*13635.34*	*13774.21*	*13811.79*	*13831.61*
9	*7027.276*	*13649.19*	*13775.31*	*13861.21*	*13960.71*
	7017.556	*13653.77*	*13757.55*	*13848.14*	*13907.91*
10	*7008.122*	*13663.37*	*13806.56*	*13961.08*	*14065.31*
	7019.59	*13655.08*	*13806.8*	*13871.34*	*13848.32*

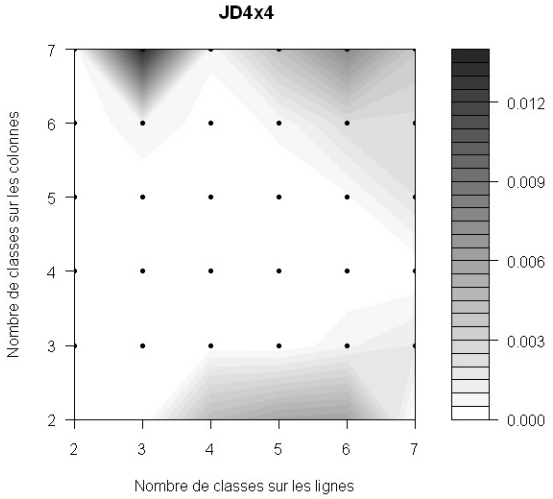


FIGURE 4.4 – Représentation de la différence entre les valeurs de χ^2 obtenues par les deux algorithmes appliqués à JD4x4.

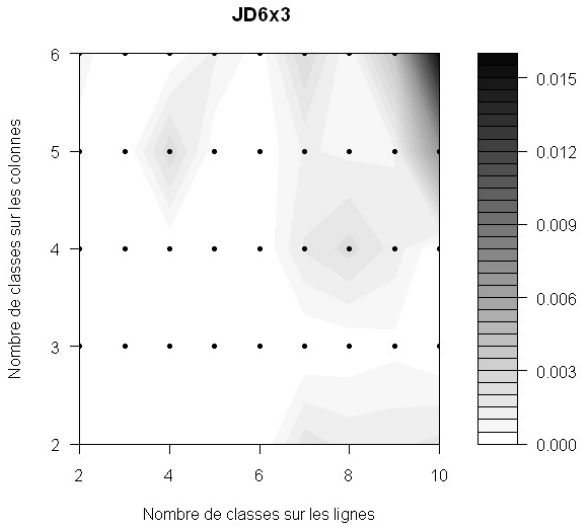


FIGURE 4.5 – Représentation de la différence entre les valeurs de χ^2 obtenues par les deux algorithmes appliqués à JD6x3.

le nombre de classes sur les lignes et les colonnes correspond au bon nombre de classes ((4,4) dans le cas de JD4x4 et (6,3) dans le cas de JD6x3).

4.4.3 Comparaison des bipartitions

Pour comparer les partitions obtenues par les deux algorithmes, nous appliquons pour chaque couple de classes, l'indice de Rand corrigé et la F-mesure sur les partitions-lignes obtenues par les deux algorithmes puis sur les partitions-colonnes.

D'après les graphiques des figures 4.6 et 4.7, nous remarquons que plus on s'éloigne du bon nombre de classes, l'indice corrigé de Rand s'éloigne de 1 donc la différence entre les partitions augmente. En effet, lorsque le nombre de classes sur les lignes et/ou les colonnes augmente, les chances d'affecter deux individus à la même classe par les deux algorithmes diminuent. Par ailleurs, nous notons que, pour tous les jeux des données, les partitions retrouvées par les deux algorithmes coïncident lorsqu'on leur donne en entrée le bon nombre de classes sur les lignes et les colonnes. En d'autres termes, lorsque le nombre de classes en entrée est correct les deux algorithmes affectent les individus, aussi bien sur les lignes que sur les colonnes, aux mêmes classes (l'indice de Rand corrigé est égal à 1).

La F-mesure calculée sur les partitions-lignes et les partitions-colonnes obtenues par les deux algorithmes pour chaque couple de classe confirme les résultats obtenues par l'indice de Rand corrigé (fig.4.8 et fig.4.9). La représentation gra-

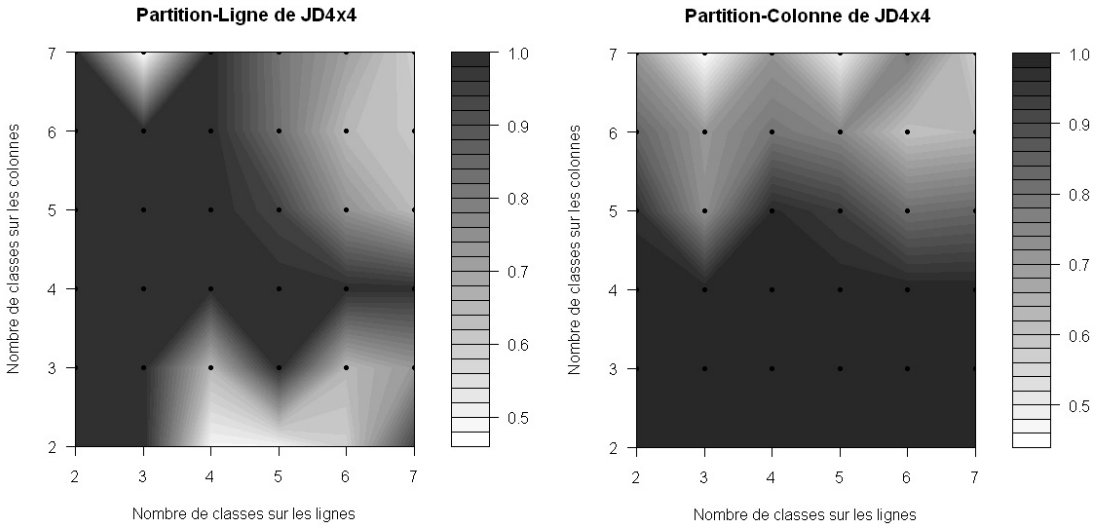


FIGURE 4.6 – Représentation de l'indice de Rand corrigé appliqué aux partitions-Lignes et partitions-Colonnes du jeu des données JD4x4 en fonction du nombre de classes sur les lignes et les colonnes.

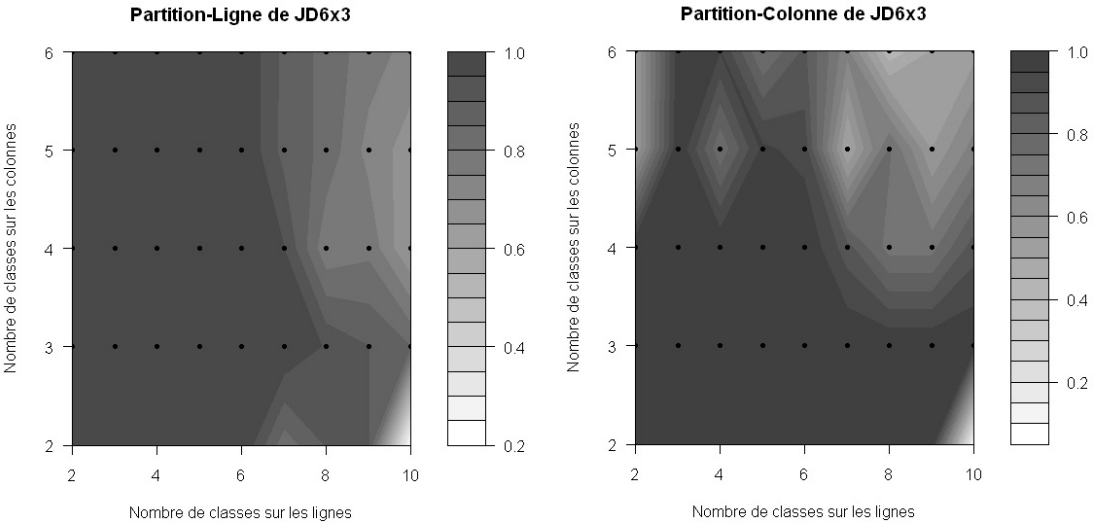


FIGURE 4.7 – Représentation de l'indice de Rand corrigé appliqué aux partitions-Lignes et partitions-Colonnes du jeu des données JD6x3 en fonction du nombre de classes sur les lignes et les colonnes.

phique de l'indice de Rand corrigé et la F-mesure appliqués aux partitions-lignes et partitions-colonnes des jeux des données JD6x3 et JD4x4 (fig.4.10) montre que les deux indices coïncident pour la majorité des couples de classes (k, l), en

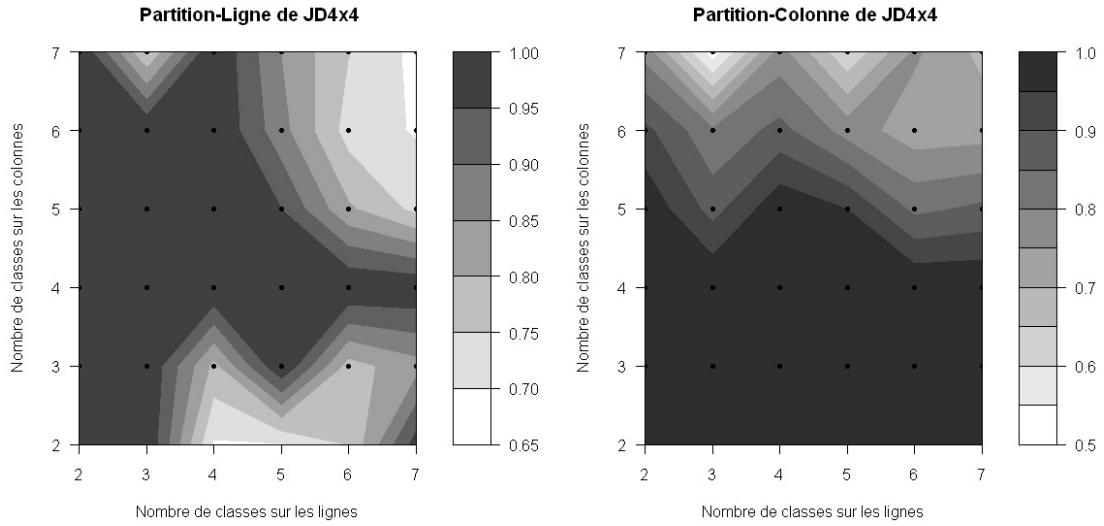


FIGURE 4.8 – Représentation de la F-mesure appliquée aux partitions-Lignes et partitions-Colonnes du jeu des données JD4x4 en fonction du nombre de classes sur les lignes et les colonnes.

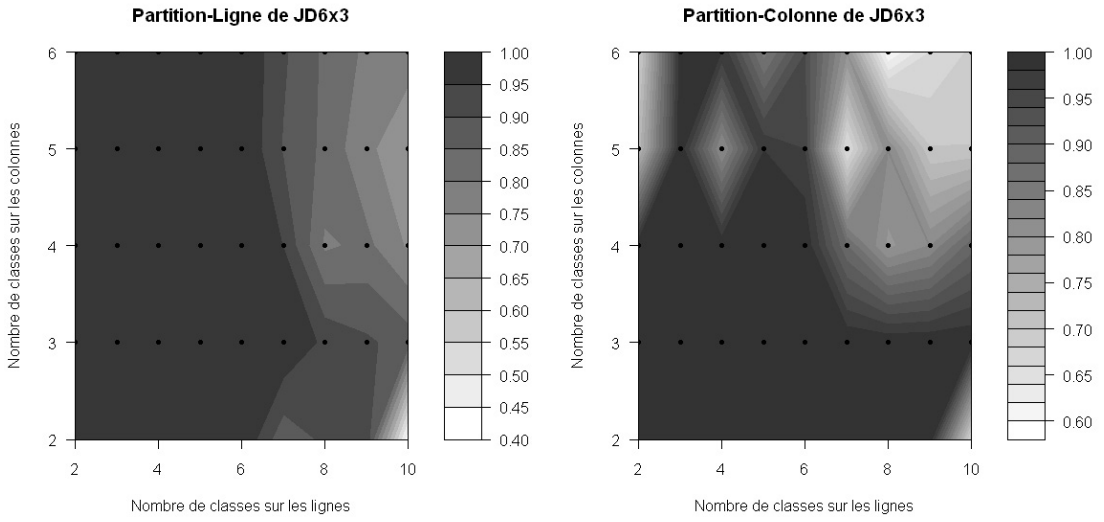


FIGURE 4.9 – Représentation de la F-mesure appliquée aux partitions-Lignes et partitions-Colonnes du jeu des données JD6x3 en fonction du nombre de classes sur les lignes et les colonnes.

particulier lorsque ce dernier correspond au bon nombre de classes sur les lignes et les colonnes. Dans le cas où ils ont des valeurs différentes, la valeur de l'indice corrigé de Rand est légèrement inférieure à celle de la F-mesure. Il s'avère alors

que l'indice corrigé de Rand est plus sensible aux différences entre les deux partitions. Ceci s'explique aussi par le fait que l'indice corrigé de Rand effectue une analyse globale des deux partitions comparées, alors que la F-mesure réalise une analyse plus fine classe par classe.

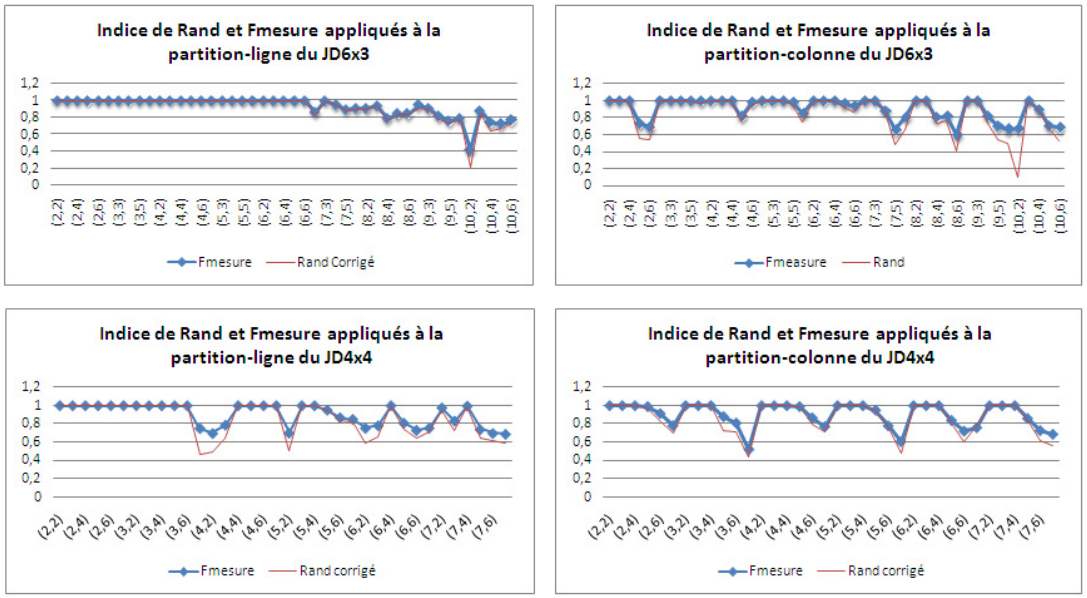


FIGURE 4.10 – Représentation de la F-mesure et l'indice corrigé de Rand appliqués aux partitions-Lignes et partitions-Colonnes des jeux des données JD6x3 et JD4x4. Les valeurs de l'indice corrigé de Rand sont égales ou légèrement inférieures à celles de la F-mesure.

4.4.4 Comparaison des bipartitions avec des classes non vides

Dans le bipartitionnement d'un tableau de données, nous imposons à l'algorithme de construire des bipartitions avec des classes non vides. Chaque fois que l'algorithme trouve une classe vide dans l'une des partitions, le programme arrête le bipartitionnement et recommence avec un nouveau tirage aléatoire des partitions initiales. Un nombre élevé de tirages aléatoires favorise l'identification des bipartitions avec des classes non vides.
Les résultats expérimentaux montrent que plus le nombre de classes sur les lignes ou les colonnes augmente plus le nombre d'échecs à trouver des bipartitions avec des classes non vides augmente.
Dans certains cas, sur les 100 tirages, aucun tirage ne permet de trouver une bipartition avec des classes non vides (taux d'échec = 100%). Nous représentons

par des courbes les taux d'échecs des deux algorithmes dans la figure 4.13).

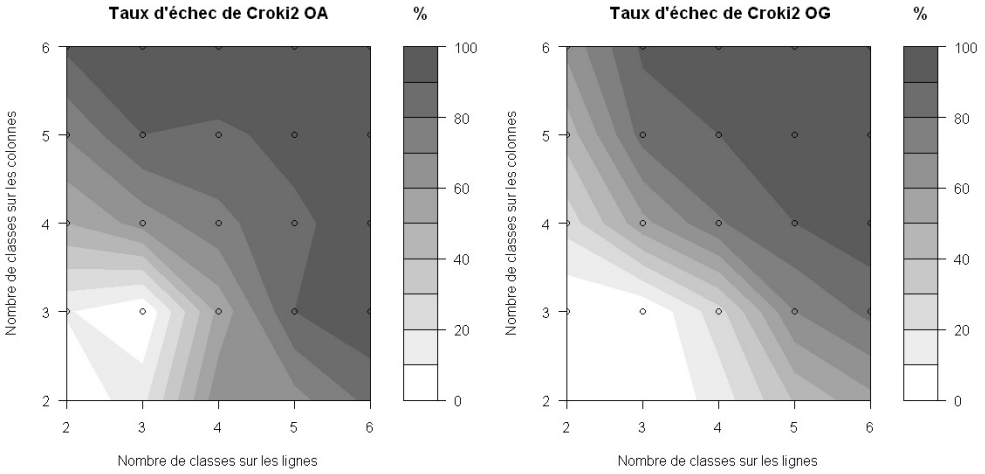


FIGURE 4.11 – Comparaison des taux d'échec des deux algorithmes à trouver des bipartitions avec des classes non vides sur JD3x3. On remarque que lorsque le nombre de classes demandées est grand sur l'une des dimensions, 6 par exemple, par rapport au nombre correct (3 dans ce cas) l'algorithme Croki2 OA présente souvent un taux d'échec élevé qui peut atteindre 100% pour certains couples de classes.

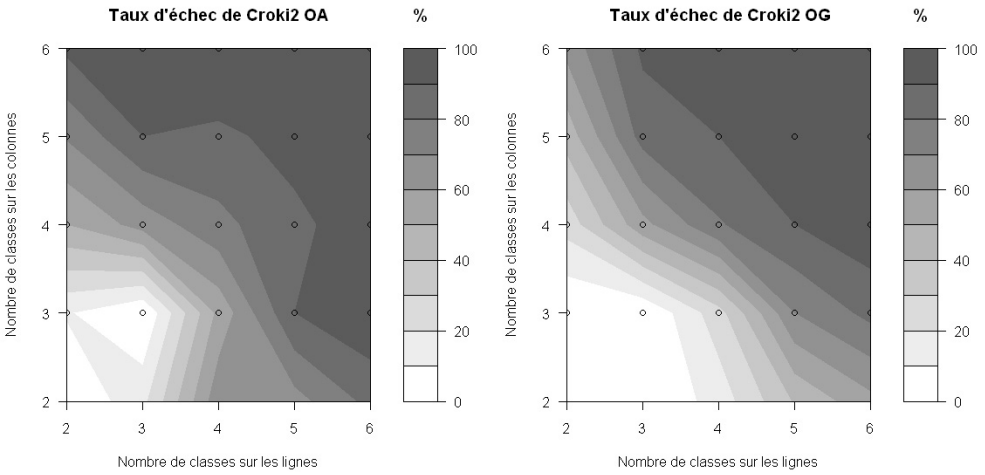


FIGURE 4.12 – Comparaison des taux d'échec des deux algorithmes à trouver des bipartitions avec des classes non vides sur JD5x4. Lorsque le nombre de classes demandées sur les colonnes dépasse 6, nous remarquons que l'algorithme Croki2 OA échoue presque souvent à trouver des partitions avec des classes non vides, contrairement à l'algorithme Croki2 OG.

Les figures 4.11 et 4.12 présentent plus explicitement les variations du nombre

de tirages aboutissant à des classes vides en fonction du nombre de classes sur les lignes et les colonnes pour les jeux des données JD3x3 et JD5x4. En examinant les graphiques, nous remarquons que l'algorithme Croki2 OG répond mieux à cette contrainte. En d'autres termes, sur les 100 tirages, le nombre d'échecs de l'algorithme Croki2 OG à trouver des bipartitions avec des classes non vides est inférieur au nombre d'échecs de l'algorithme Croki2 OA.

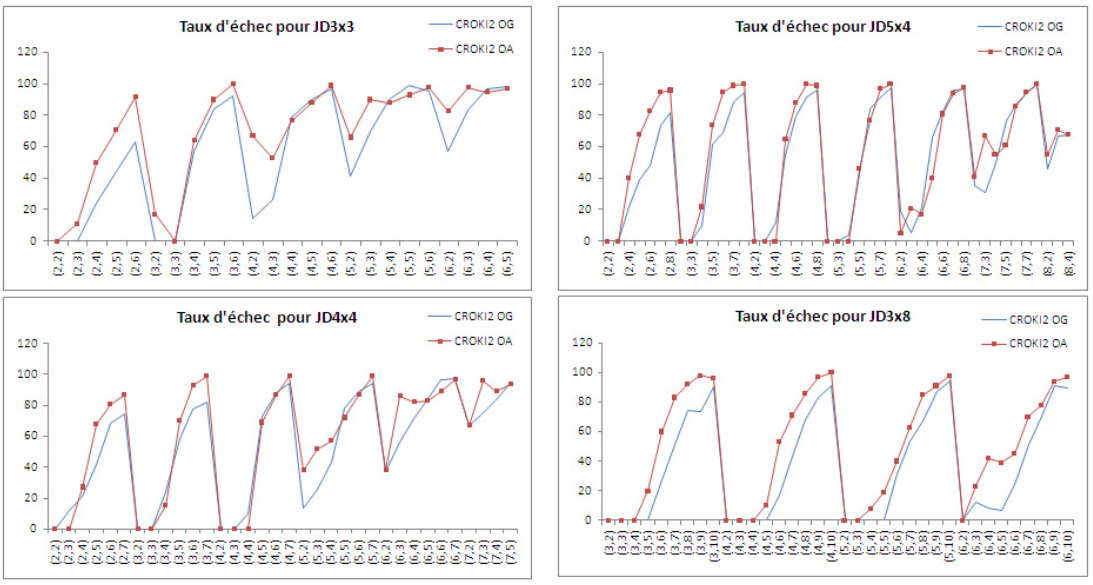


FIGURE 4.13 – Comparaison des taux d'échec des deux algorithmes à trouver des bipartitions avec des classes non vides sur les jeux des données JD3x3, JD4x4, JD5x4 et JD3x8. Nous remarquons que la courbe rouge représentant le taux d'échec de l'algorithme Croki2 OA est souvent au dessus de la courbe bleue représentant le taux d'échec de l'algorithme Croki2 OG.

4.4.5 Performance dans l'identification de la structure initiale des données

Afin d'évaluer la performance de chacun des algorithmes dans l'identification de la structure initiale des données, nous proposons d'utiliser l'indice corrigé de Rand.

Pour ce faire, nous appliquons les deux algorithmes (Croki2 OA et Croki2 OG) sur tous les jeux des données générées en faisant varier le nombre de classes sur les lignes et les colonnes. Pour chaque algorithme et pour chaque couple de classes nous récupérons la partition ligne et la partition colonne correspondante. Chaque partition est comparée à la partition initiale (artificielle) à travers l'indice de Rand corrigé. Une valeur maximale de l'indice (proche de 1) indique que

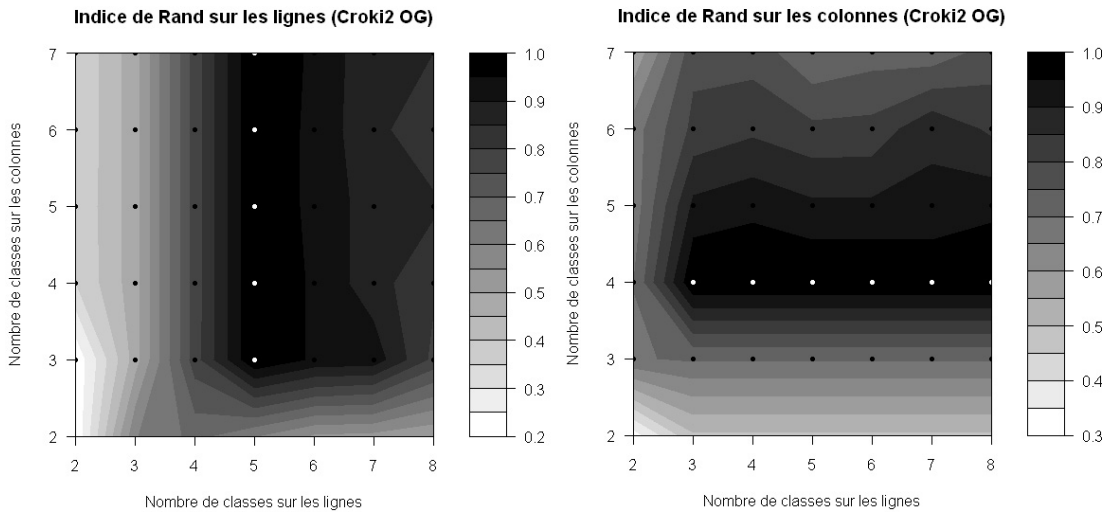


FIGURE 4.14 – Comparaison des partitions résultant de l'application de Croki2 OG sur les lignes et les colonnes de JD5x4 avec les partitions artificielles par l'indice de Rand corrigé.

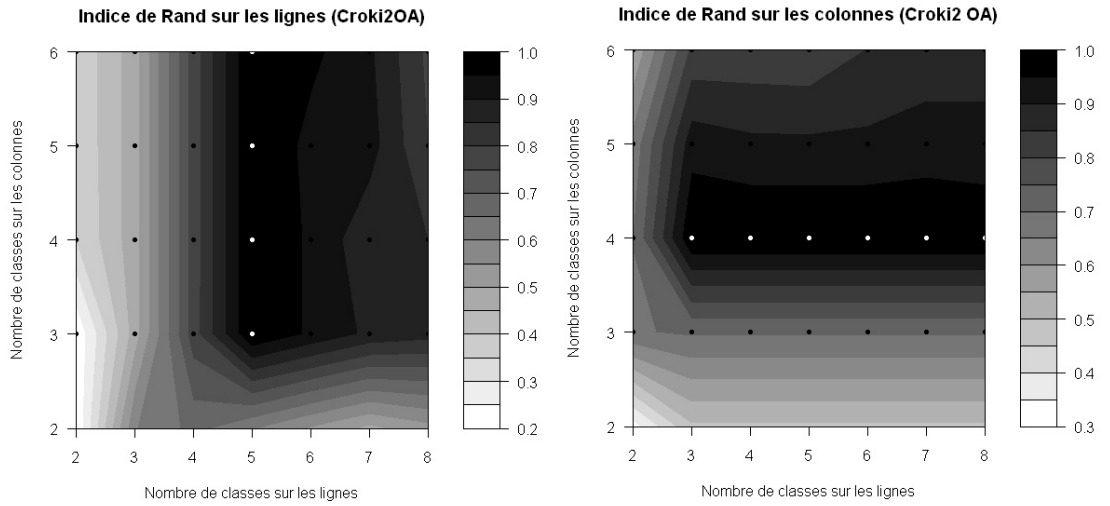


FIGURE 4.15 – Comparaison des partitions résultant de l'application de Croki2 OA sur les lignes et les colonnes de JD5x4 avec les partitions artificielles par l'indice de Rand corrigé.

les deux partitions sont égales. En d'autres termes, l'algorithme affecte les individus aux bonnes classes. Soit l'exemple du jeu des données JD5x4. L'indice de Rand corrigé appliqué à la partition-colonne prend la valeur 1 pour les couples $\{(3,4),(4,4),(5,4),(6,4)\}$ (fig.4.14 et fig.4.15). Ainsi, au niveau des colonnes, l'algorithme retrouve la bonne partition composée de 4 classes et affecte les va-

riables aux bonnes classes quelque soit le nombre de classes sur les lignes. De même, au niveau des lignes, l'indice de Rand corrigé prend la valeur 1 pour les couples $\{(5,3),(5,4),(5,5),(5,6),(5,7),(5,8)\}$. L'algorithme affecte les individus aux bonnes classes lorsque le nombre de classes sur les lignes est égal à 5 indépendamment du nombre de classes sur les colonnes. Ceci montre bien que l'algorithme est apte à identifier le bon nombre de classes dans les données.

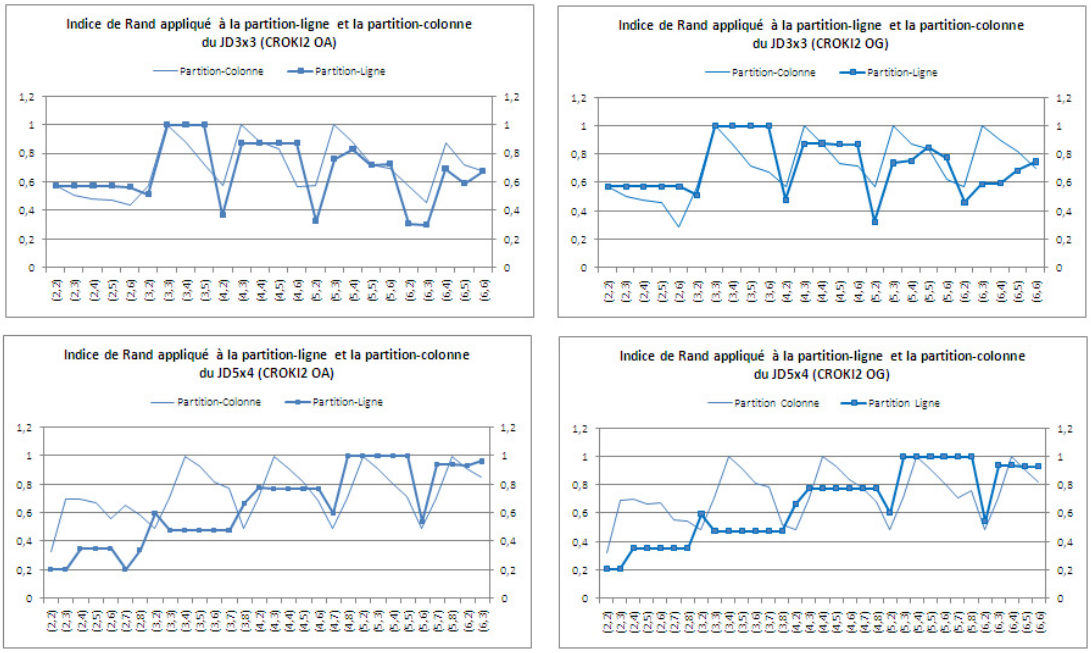


FIGURE 4.16 – Indice de Rand corrigé appliqué à la partition-ligne et la partition-colonne obtenues par les algorithmes Croki2 OA et Croki2 OG sur les jeux des données 5x4 et 3x3.

Le couple de classes qui maximise à la fois l'indice de Rand corrigé appliqué à la partition-ligne et celui appliqué à la partition-colonne est le (5,4) qui correspond au bon couple de classes. Pour tous les jeux des données, les deux indices de Rand corrigé obtenus sur la partition-ligne et la partition-colonne sont égaux à 1 lorsque le couple de classes coincide avec le couple choisi dès le départ pour la construction des jeux des données, par exemple (3,3) pour JD3x3 et (5,4) pour JD5x4 (fig.4.16). Ainsi, lorsque le nombre de classes sur les lignes et les colonnes correspond au bon nombre de classes, les deux algorithmes, Croki2 OA et Croki2 OG, sont capables d'affecter les individus aux bonnes classes.

4.4.6 Comparaison de la complexité des deux algorithmes

L'algorithme Croki2 OA est itératif à deux niveaux. Le premier niveau d'itérations assure la convergence de l'algorithme des nuées dynamiques. Soient, pour un couple de classes (K, L) donné, $niter11$ et $niter12$ le nombre d'itérations nécessaires pour la convergence de l'algorithme des nuées dynamiques sur les lignes et sur les colonnes respectivement, $niter2$ le nombre d'itérations nécessaires pour alterner l'optimisation sur les lignes et l'optimisation sur les colonnes.

K	L	Tirage	niter	Khi2	complexité OG
6	3	1	6	13576.02	32400
6	3	2	10	12400.36	54000
6	3	3	6	13576.02	32400
6	3	4	4	12501.52	21600

K	L	Tirage	niter11	niter12	niter2	Khi2	complexité OA
6	3	1	14	3	1		
6	3	1	3	1	2	12387.86	68400
6	3	2	10	3	1		
6	3	2	4	1	2	12492.66	57600
6	3	3	10	2	1		
6	3	3	5	1	2	12528.51	59400
6	3	4	13	8	1		
6	3	4	5	2	2	11086.80	82800

FIGURE 4.17 – Exemple des données permettant le calcul de la complexité de Croki2 OA et Croki2 OG sur JD6x3.

Comme l'opération la plus coûteuse dans l'algorithme Croki2 est le calcul des distances, nous estimons le nombre total de calcul de distances dans l'algorithme. Soit le tableau des données de dimensions (N, M). Pour un tirage donné, et pour chaque itération au niveau supérieur (allant de 1 à niter2), il est nécessaire d'effectuer $niter11 \times N \times K$ calculs de distances dans R^L et $niter12 \times M \times L$ calculs de distances dans R^K. Or niter11 et niter12 varient d'une itération à une autre dans la boucle supérieure. Ainsi, le nombre total d'opérations pour Croki2 OA est donné par la formule suivante pour un tirage donné :

$$Croki2OA : \sum_{i=1}^{niter2} \times (niter11_i \times N \times K \times L + niter12_i \times M \times L \times K)$$

$$Croki2OA : K \times L \times \sum_{i=1}^{niter2} (niter11_i \times N + niter12_i \times M)$$

L'algorithme Croki2 OG présente un seul niveau d'itérations. Soit *niter* le nombre d'itérations nécessaires pour la convergence de l'algorithme pour un couple de classes (K, L) donné. Le nombre total d'opérations dans ce cas est :

$$Croki2OG : niter \times (N \times K \times L + M \times L \times K) = niter \times K \times L \times (N + M)$$

Pour comparer expérimentalement la complexité des deux algorithmes, nous les appliquons sur les jeux des données artificielles. Pour l'algorithme Croki2OA, généralement au bout de 2 itérations au niveau supérieur, i.e. $niter2 = 2$, l'algorithme converge. Par contre, les valeurs de niter11 et niter12 sont souvent différentes.

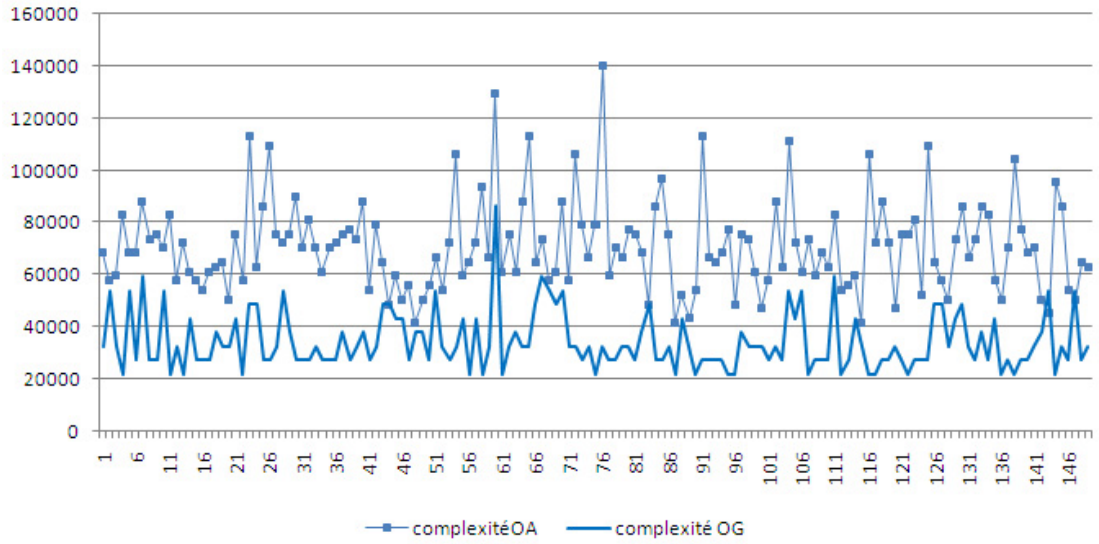


FIGURE 4.18 – Comparaison de la complexité des deux algorithmes sur JD6x3 pour $K = 6$ et $L = 3$. Le jeu des données JD6x3 est composé de $N = 200$ lignes et $M = 100$ colonnes, comme tous les jeux des données artificielles que nous avons générées. L'axe des abscisses représente le nombre de tirage.

Dans l'exemple de la figure.4.18, nous fixons le nombre de classes sur les lignes à $K = 6$ et le nombre de classes sur les colonnes à $L = 3$ et nous nous concentrons sur les quatre premiers tirages aléatoires des centres initiaux. Dans tous ces tirages, la complexité de l'algorithme Croki2 OG est nettement inférieure à celle de l'algorithme Croki2 OA. Par exemple, dans le premier tirage, la complexité de Croki2 OG est égale à 32400 alors que celles de l'algorithme Croki2 OA est égale à 68400. Dans le deuxième tirage, la complexité de Croki2 OG est égale à 54000 alors que celles de l'algorithme Croki2 OA est égale à 57600. Le calcul de

la complexité des deux algorithmes sur les différents jeux des données artificielles montre que la complexité de l'algorithme Croki2 OA dépasse celle de l'algorithme Croki2 OG. Par suite, l'algorithme Croki2 OG permet de gagner en temps de calcul, en particulier lorsque le nombre de classes fixé sur les lignes et les colonnes est grand.

4.5 Synthèse

Les expérimentations effectuées dans ce chapitre sur les données artificielles ont servi à étudier les deux algorithmes Croki2 OA et Croki2 OG. Les résultats expérimentaux ont démontré que les deux algorithmes sont équivalents quand à la solution proposée pour tout problème de classification croisée. Une solution obtenue par l'un ne peut être améliorée par l'autre. L'avantage de cette caractéristique consiste à pouvoir mettre en oeuvre les deux algorithmes parallèlement tout en réduisant le nombre de tirages aléatoires. A la convergence, toute solution commune entre les deux peut être considérée comme la bonne solution. Cette technique permet de réduire le temps de calcul. D'autre part, la comparaison entre les deux algorithmes a montré que l'algorithme Croki2 OG répond mieux à la contrainte des partitions sans classes vides et permet également de gagner en temps de calcul.
Dans le chapitre suivant, nous nous concentrons sur le problème de détermination du nombre de classes dans les deux algorithmes.

Chapitre 5

Choix du nombre de classes dans l'algorithme Croki2

5.1 Introduction

Une des premières critiques que présentent certains algorithmes de bipartitionnement, comme les méthodes classiques de partitionnement, est la connaissance requise du nombre de classes dans la population. L'algorithme Croki2 qui fait appel à l'algorithme des nuées dynamiques pour la classification alternée des lignes et des colonnes nécessite de fixer le nombre de classes *a priori*. L'utilisateur se trouve en présence de plusieurs solutions possibles sans moyen permettant de favoriser une solution. Le problème devient plus complexe dans le cas de bipartitionnement où il faut déterminer le nombre de classes sur les lignes et sur les colonnes. Ainsi, bien que le problème d'identification du bon nombre de classes soit consideré comme étant "un faux problème" puisque dans le cas de données réelles ce nombre de classes est rarement connu, il nous semble qu'il est nécessaire d'en avoir une idée afin d'automatiser la tâche de bipartitionnement. Notre but n'est pas donc de retrouver le nombre exact de classes sur les lignes et les colonnes, mais d'avoir la bipartition la plus profitable, en ayant un nombre ni trop grand ni trop petit de classes, aisément interprétables.
Différentes approches ont été proposées dans la littérature pour pallier ce problème dans le cas de méthodes de partitionnement simple. En effet, il est possible d'imposer des contraintes en ce qui concerne les classes : nombre maximum d'objets par classe par exemple ou d'effectuer plusieurs classifications avec des valeurs différentes de k (nombre de classes) et déterminer la partition qui minimise ou maximise les critères de classification. Cependant, aucun travail n'a été consacré à la résolution du problème de détermination du nombre de classes dans le cadre de la classification simultanée.
Nous présentons dans la suite certains des critères proposés pour les méthodes

de partitionnement simple que nous adaptons au cas de bipartitionnement. Afin d'identifier les critères les plus adéquats par rapport aux données issues du Web Content Mining, nous proposons de les tester sur des données artificielles (tab. 5.1) qui simulent le contenu d'un site Web et qui sont générées suivant une méthodologie présentée dans la section 4.2 du chapitre précédent.

TABLE 5.1 – Jeux des données artificielles

	Nombre de classes sur les lignes	Nombre de classes sur les colonnes
JD3x3	3	3
JD4x4	4	4
JD5x4	5	4
JD6x3	6	3
JD3x8	3	8
JD6x6	6	6

5.2 Détermination du nombre de classes dans les méthodes de classification automatique

Dans le cadre de la classification automatique, les auteurs définissent trois types de critères de validation selon que l'on dispose ou pas d'information a priori sur les données : critère externe, critère interne et critère relatif. Les deux premiers sont basés sur des méthodes statistiques et nécessitent un temps de calcul élevé [HBV01].

- Le critère externe mesure le degré avec lequel la structure découverte sur les données confirment des informations connues *a priori* sur ces données [JD88].
- Le critère interne mesure l'écart entre la structure engendrée par un algorithme de classification et les données, en tenant compte du biais introduit par l'utilisation de l'algorithme pour obtenir la structure de classification.
- Le critère relatif permet de comparer deux structures de classification. Il décide quelle structure est meilleure dans le sens plus stable ou mieux appropriée pour les données et ce en utilisant un ensemble d'indices.

Ces critères de validation sont proposés pour les méthodes de partitionnement simple. Dans le cadre de bipartitionnement, aucun critère, n'a été proposé pour résoudre le problème de détermination du nombre de classes sur les lignes et les colonnes.
Notre idée dans ce chapitre est d'étendre l'utilisation de certains de ces critères

aux méthodes de bipartitionnement. Nous présentons dans les paragraphes suivants les indices de validation les plus utilisés et qui sont basés sur des critères relatifs ou internes. Les indices basés sur des critères externes, à savoir l'indice de Rand et la F-mesure, sont déjà présentés dans la section précédente.

5.2.1 Indices de validation basés sur des critères relatifs

Les indices de validité sont appliqués sur les différentes partitions obtenues pour différentes valeurs de k. Une méthodologie d'évaluation est requise pour déterminer le nombre optimal de clusters $k*$. C'est ce qu'on appellera indice de validité des clusters (cluster validity index).
Le processus pour le calcul de l'indice de validation des clusters est résumé par les étapes suivantes :

- Etape 1 : Initialiser les paramètres de l'algorithme excepté le nombre de clusters k.
- Etape 2 : Appliquer l'algorithme pour différentes valeurs de k avec $k = 2, 3, , cmax$. ($cmax$ est fixé par l'utilisateur).
- Etape 3 : Calculer l'indice de validité pour chaque partition obtenue à l'étape 2.
- Etape 4 : Choisir le nombre optimal k^* de clusters.

Plusieurs indices de validité de clusters sont proposés dans la littérature. Nous décrivons dans ce qui suit les indices les plus répandus.

5.2.1.1 Indice Dunn de validité

L'indice de Dunn proposé par [Dun74] a pour objectif d'identifier la partition dont les classes sont homogènes et bien séparées. L'indice D est défini par la formule suivante :

$$D = min_{1 \leq i \prec j \leq K} \frac{d(C_i, C_j)}{max_{1 \leq k \leq K} d'(C_k)}$$

avec :

- K est le nombre de classes de la partition
- $d(C_i, C_j)$ est la distance entre les classes C_i et C_j (distance interclasse)
- $d'(C_k)$ est la distance intraclasse de la classe C_k.

Une bonne partition maximise la distance interclasse et minimise la distance intraclasse. Ainsi, le nombre de classes qui maximise D est le bon nombre de classes.

5.2.1.2 Indice de Davies-Bouldin

L'indice de Davies-Bouldin [DB79] est fonction des distances intra-classes et des distances inter-classes. Une valeur faible de l'indice DB indique que les classes sont compactes et séparables et par suite la classification est bonne.

$$DB = \frac{1}{K} \sum_{k} max_{k \neq k'} \frac{S_n(c_k) + S_n(c_{k'})}{S(c_k, c_{k'})}$$

avec

- K est le nombre de classes
- S_n est la moyenne des distances entre les objets de la classe C_k et le centre de la classe c_k.
- $S(c_k, c_{k'})$) est distance entre les centres des classes c_k et $c_{k'}$

Dans le cas des tableaux de contingence, la distance utilisée est celle de χ^2.

5.2.1.3 Indice Silhouette de Rousseeuw

L'indice Silhouette [Rou87] est calculé pour chaque objet de la population à travers la formule suivante :

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

- $a(i)$ est la moyenne des distances entre l'objet i et les objets affectés à la même classe
- $b(i)$ est la moyenne des distances entre l'objet i et les objets affectés à la classe la plus proche

Nous définissons la classe la plus proche comme étant celle dont le centre est le plus proche de l'objet en question. D'après la formule, la valeur $S(i)$ est comprise entre -1 et 1.

- Si $S(i)$ est proche de 1 alors l'objet i est affecté à la classe la plus appropriée.
- Si $S(i)$ est proche de 0 alors l'objet i semble être loin des deux classes et peut être affecté à une autre classe plus proche.
- Si $S(i)$ est proche de -1 alors l'objet i est mal classifié.

La moyenne de tous les indices Silhouette calculés pour tous les objets de la population donne une indication sur la qualité de la classification. En effet, la meilleure partition est celle dont l'indice Silhouette global S est maximum.

$$S = \frac{1}{n} \sum_{i=1}^{n} S(i)$$

Ainsi, la valeur de k qui maximise S est choisie comme le bon nombre de classes.

5.2.1.4 Indice de Hubert et Levin ou C-index

L'indice de Hubert et Levin [HL76], connu sous le nom de C-index, est calculé par la formule suivante :

$$HL(K) = \frac{SD_{intra} - min(SD)}{max(SD) - min(SD)}$$

avec :

- SD_{intra} est la somme des distances intra-classe (au nombre de N_{intra}
- $min(SD)$ est la somme des N_{intra} plus petites distances
- $max(SD)$ est la somme des N_{intra} plus grandes distances

La valeur de k qui minimise $HL(k)$ est choisie comme le bon nombre de classes. Une bonne partition est indiquée par une petite valeur de HL, valeur toujours comprise entre 0 et 1.

5.2.1.5 Indice de Baker et Hubert

L'indice de Baker et Hubert [BH75] est une adaptation de la statistique Gamma de Goodman et Kruskal [GK54]. La formule de calcul de cet indice est la suivante :

$$BH(k) = \frac{S^+ - S^-}{S^+ + S^-}$$

avec S^+ est le nombre de quadruplets cohérents, tandis que S^- est le nombre de quadruplets incohérents.
Pour calculer S^+ et S^- il faut examiner tous les quadruplets d'objets, et les distances u induites par la partition. Si deux objets i et j appartiennent à la même classe de la partition alors $u(i,j) = 0$ sinon $u(i,j) = 1$.
Un quadruplet est dit :

- cohérent si $u(i,j) < u(k,l)$ et $d(i,j) < d(k,l)$
- incohérent si [$u(i,j) < u(k,l)$ et $d(i,j) > d(k,l)$] ou [$u(i,j) > u(k,l)$ et $d(i,j) < d(k,l)$]

Les valeurs de cet indice sont dans l'intervalle [-1,1]. La valeur de k qui maximise BH correspond au bon nombre de classes.

5.2.1.6 La pseudo-statistique de Calinski et Harabasz

La pseudo-statistique de Calinski et Harabasz [CH74] est une fonction de la distance intra-classe et la distance inter-classe. Elle est calculée par la formule suivante :

$$CH(k) = \frac{Trace(B)/(k-1)}{Trace(W)/(n-k)}$$

où B est la matrice des variances-covariances inter-groupes et W est la matrice des variances-covariances intra-groupes.
Il s'agit, en fait, du quotient de la somme des variances inter-classes par la somme des variances intra-classes. Une bonne partition est indiquée par une valeur élevée de $CH(k)$.
La valeur de k qui maximise $CH(k)$ est choisie comme le bon nombre de classes.

5.2.1.7 Indice de Krzanowski et Lai

Cet indice proposé par [KL88] est défini par l'équation suivante :

$$KL(k) = \left| \frac{DIFF(k)}{DIFF(k+1)} \right|$$

où $DIFF(k) = (k-1)^{2/p} W_{k-1} - k^{2/p} W_k$ et p est le nombre de variables.
La valeur de k qui maximise $KL(k)$ est choisie comme le bon nombre de classes.

5.2.2 Indices de validation basés sur des critères internes

Outre le χ^2 de contingence, nous proposons d'appliquer d'autres indices utilisés pour l'analyse conjointe de deux variables qualitatives tels que l'indice de Tschuprow et l'indice de Cramer. En effet, la partition ligne et la partition colonne générées par l'algorithme Croki2 peuvent être considérées comme deux variables qualitatives dont on cherche à maximiser la corrélation.
Afin d'identifier le bon nombre de classes k^* sur les lignes et l^* sur les colonnes, nous proposons de faire varier k et l et mesurer à chaque fois la corrélation entre les deux partitions résultantes. Un bon bipartitionnement est celui qui maximise cette corrélation.

5.2.2.1 Indice de Cramer

L'indice de Cramer est calculé par la formule suivante.

$$C = \sqrt{\frac{\chi^2}{n(d-1)}}$$

avec $d = inf(k, l)$ et n est la somme sur les lignes et les colonnes du tableau.
On vérifie $0 \leq T \leq C \leq 1$.
En appliquant cet indice sur tous les jeux des données, nous remarquons que la meilleure valeur est obtenue généralement pour le couple (K^*, L^*) telle que $K^* = max(K)$ et/ou $L^* = max(L)$. En d'autres termes, l'indice de Cramer croit avec l'augmentation du nombre de classes sur les lignes ou les colonnes.

À titre d'exemple, pour le jeu des données JD4x4, la valeur maximale de l'indice est obtenue pour le couple (7,7) (Tab.5.2). En le testant sur tous les autres jeux des données, nous pouvons conclure que l'indice de Cramer ne permet pas d'identifier le bon couple de classes.

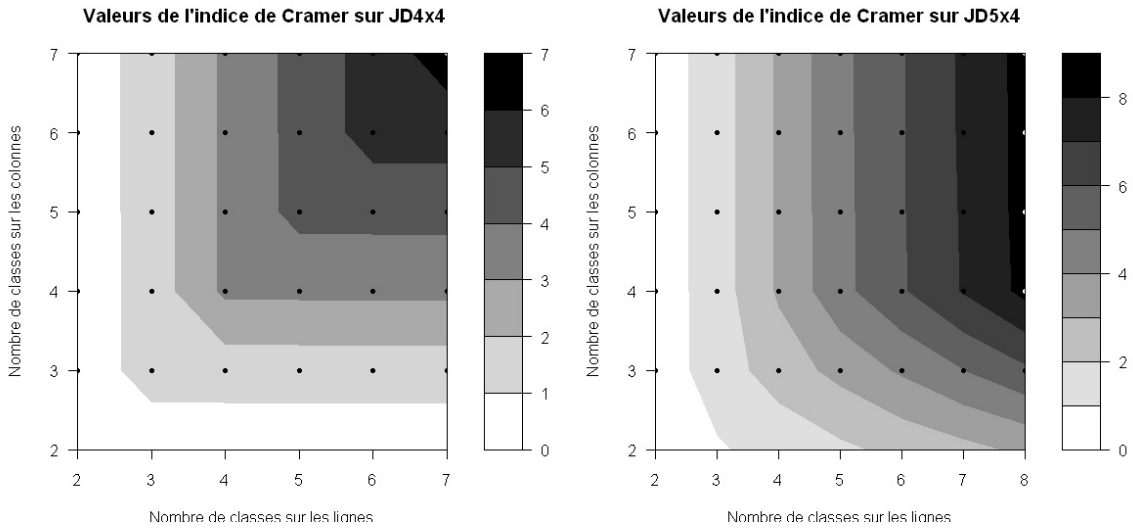


FIGURE 5.1 – Représentation de l'indice de Cramer en fonction du nombre de classes sur JD4x4 et JD5x4. Les valeurs maximales de l'indice sont obtenues pour le couple (7,7) sur JD4x4 et le couple (8,7) sur JD5x4.

TABLE 5.2 – Valeurs de l'indice Cramer calculé sur les partitions du JD4x4

	2	3	4	5	6	7
2	0.368	0.37	0.370	0.370	0.370	0.370
3	0.371	1.442	1.444	1.446	1.447	1.447
4	0.372	1.447	3.199	3.202	3.204	3.206
5	0.372	1.451	3.207	4.313	4.32	4.325
6	0.372	1.454	3.214	4.318	5.415	5.404
7	0.375	1.458	3.221	4.324	5.432	**6.517**

5.2.2.2 Indice de Tschuprow

L'indice de Tschuprow constitue une pénalisation du χ^2 dont les valeurs croissent avec le nombre de classes sur les lignes et les colonnes.

$$T = \sqrt{\frac{\chi^2}{n\sqrt{(k-1)(l-1)}}}$$

avec n est la taille de l'échantillon. $0 \leq T \leq 1$

En testant cet indice sur le jeu des données artificielles JD4x4, nous remarquons qu'il favorise le couple (2,2) puisque le χ^2 est divisé par $1 = \sqrt{(2-1) \times (2-1)}$ (voir Tab.5.3). De même pour tous les jeux des données, le couple de classes qui maximise l'indice de Tschuprow est toujours le couple (2,2).

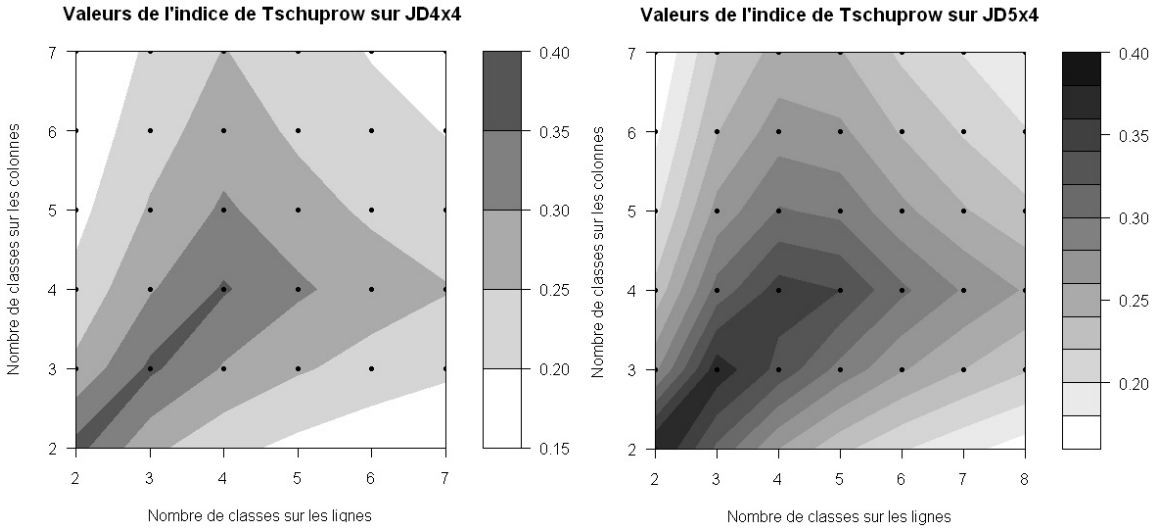


FIGURE 5.2 – Représentation de l'indice de Tschuprow en fonction du nombre de classes sur JD4x4 et JD5x4. Les valeurs maximales de l'indice sont obtenues toujours pour le couple (2,2).

TABLE 5.3 – Valeurs de l'indice de Tschuprow calculé sur les partitions du JD4x4

	2	3	4	5	6	7
2	**0.368**	0.261	0.214	0.185	0.166	0.151
3	0.262	0.36	0.295	0.256	0.229	0.209
4	0.215	0.295	0.355	0.308	0.276	0.252
5	0.186	0.257	0.309	0.27	0.241	0.221
6	0.166	0.23	0.277	0.241	0.217	0.197
7	0.153	0.21	0.253	0.221	0.198	0.181

5.2.2.3 Le Chi2 pénalisé

Dans le cadre probabiliste, les méthodes de classification se basent sur la maximisation de la vraisemblance ou la vraisemblance complétée. Or celles-ci augmentent avec le nombre de classes. Plus il y a de classes, plus la partition est ajustée aux données, et donc plus la vraisemblance est grande. Les critères d'information utilisés dans ce cas pour le choix du modèle ou du nombre de classes pénalisent le maximum de vraisemblance par le nombre de paramètres à estimer dans le modèle. Leur formulation générale [Jol03] est

$$C(s) = -2(L_{max}(s) + \gamma_c \times \nu(s))$$

où

- $Lmax(s)$ est le maximum du critère de vraisemblance du modèle à s classes,
- γ_c est un coefficient de pénalisation de la complexité spécifique à chaque critère,
- $\nu(s)$ est le nombre de paramètres libres dans le modèle à s classes.

Parmi les critères de pénalisation du maximum de la vraisemblance, citons le critère d'information Akaike (AIC) [Aka73], le critère AIC modifié [Boz81] [Boz83], appelé $AIC3$, et le Bayesian Information Criterion BIC [Sch78] qui sont largement diffusés et appliqués.
A l'instar des critères d'information qui pénalisent la vraisemblance, l'indice de Tschuprow constitue une pénalisation du χ^2 par le produit du nombre de classes sur les lignes et les colonnes. En effet,

$$Log(T) = \frac{1}{2}(Log\chi^2 - Log(n \times \sqrt{(k-1)(l-1)}))$$

Cependant, en le testant sur les données artificielles, nous remarquons que le couple (2,2) correspond souvent à la meilleure valeur de l'indice. Nous proposons alors de remplacer le $(k-1) \times (l-1)$ par $k \times l$. Par suite,

$$\chi^2_p = \sqrt{\frac{\chi^2}{n\sqrt{k \times l}}}$$

Le χ^2 pénalisé par la racine du produit du nombre de classes sur les lignes et le nombre de classes sur les colonnes permet d'avoir un résultat meilleur que celui obtenu par l'indice de Tschuprow. En effet, pour les jeux de données JD3x3, JD5x4 et JD4x4 (fig. 5.3), il identifie le bon nombre de classes dans les données. Par contre, pour les jeux des données JD6x3, JD6x6 et JD3x8, l'indice n'arrive pas à détecter le bon nombre de classes (fig. 5.4).

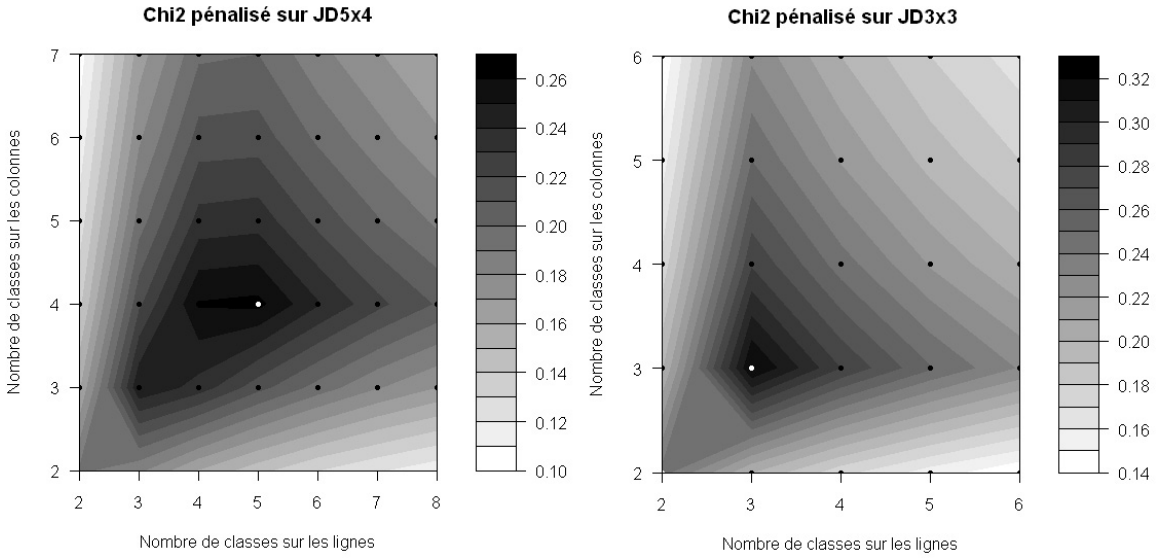


FIGURE 5.3 – Représentation du χ^2 pénalisé en fonction du nombre de classes sur JD3x3 et JD5x4. Les valeurs maximales de l'indice sont obtenues pour le couple (3,3) sur JD3x3 et le couple (5,4) sur JD5x4.

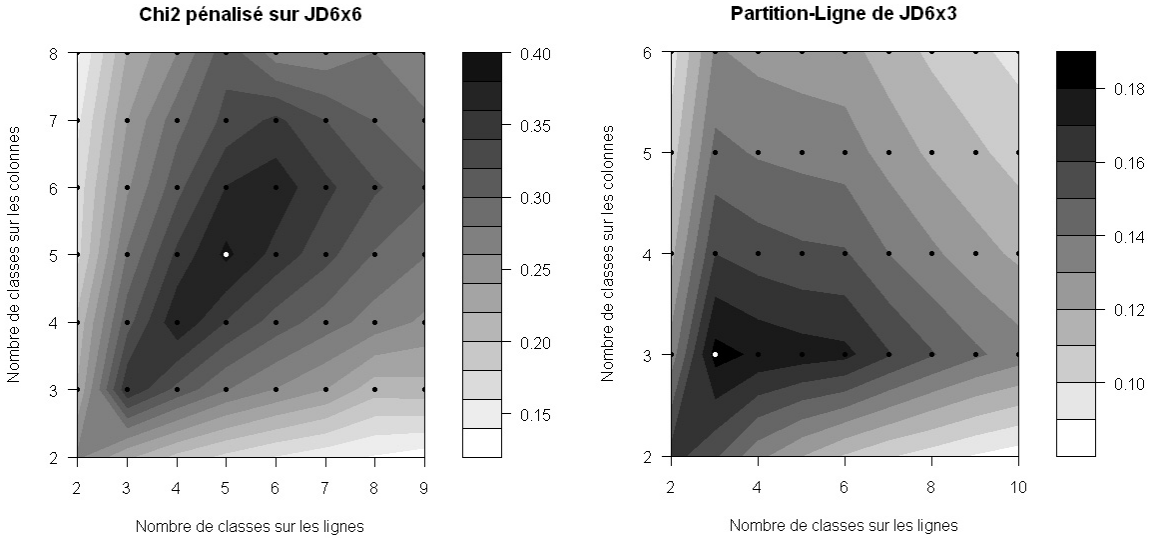


FIGURE 5.4 – Représentation du Chi2 pénalisé en fonction du nombre de classes pour JD6x6 et JD6x3. Le χ^2 pénalisé propose (5,5) comme meilleur couple de classes pour JD6x6 et (3,3) pour JD6x3. Dans ces deux cas, cet indice n'a pas réussi à identifier les bons couples de classes.

TABLE 5.4 – Valeurs de l'indice de Chi2 pénalisé calculé sur les partitions du JD4x4

	2	3	4	5	6	7
2	0.184	0.151	0.131	0.117	0.107	0.099
3	0.151	0.24	0.208	0.187	0.171	0.158
4	0.131	0.209	**0.267**	0.239	0.218	0.202
5	0.118	0.187	0.239	0.216	0.197	0.183
6	0.107	0.171	0.219	0.197	0.181	0.167
7	0.1	0.159	0.203	0.183	0.168	0.155

5.2.3 Détermination du nombre de classes par la coupure du dendrogramme

Dans les méthodes de classification hiérarchique, le dendrogramme indique l'ordre dans lequel les agrégations (ou divisions) successives ont été opérées ainsi que la valeur de l'indice à chaque niveau d'agrégation (ou division). La stratégie proposée pour la détermination du bon nombre de classes dans ces méthodes est la coupure du dendrogramme [LMP00] utilisée dans le logiciel SPAD [1].
En appliquant l'algorithme de Classification Ascendante Hiérarchique (CAH) avec le critère d'agrégation de Ward, les deux classes à rassembler, ainsi que l'inertie intra-classe sont connues à chaque niveau d'agrégation. La problématique consiste à découper le dendrogramme de manière à avoir le meilleur couple (inertie intra-classe, nombre de classes) puisque la diminution du nombre de classes dans les données fait augmenter l'inertie intra-classe. La coupure du dendrogramme au niveau d'un saut important de l'indice permet d'avoir une partition de bonne qualité où les individus regroupés au dessous de la coupure sont proches et ceux regroupés au dessus de la coupure sont éloignés. Ainsi, la stratégie de recherche de bon nombre de classes dans cette méthode consiste à identifier un *coude* sur le graphique représentant les valeurs du critère optimisé, à savoir l'inertie intra-classe. Ce coude est repéré à l'aide des dérivées premières et secondes définies dans le paragraphe 5.2.3.1 des valeurs des n-1 gains d'inertie intra-classe obtenus à chaque itération. Lorsque plusieurs solutions acceptables se présentent, en particulier dans le cas des données réelles où les individus se positionnent sur les frontières des classes, [Sil09] propose deux stratégies de choix du nombre de classes.
La première stratégie consiste à parcourir de droite à gauche le graphique des différences secondes et choisir la première valeur significative (le premier pic). La seconde consiste à parcourir le graphique dans l'autre sens, c'est à dire de gauche

1. http ://www.spadsoft.com

à droite, et choisir aussi la première valeur significative. La différence entre les deux stratégies réside dans le fait que la première stratégie privilégie la solution ayant le plus grand nombre de classes alors que la seconde privilégie la solution ayant le plus petit nombre de classes.
Des expérimentations sur des données générées artificiellement montrent que cette méthode est efficace dans l'identification de la structure initiale des données.

5.2.3.1 Définition de la dérivée première et seconde dans le cas discret

Soit une fonction f de la variable réelle x. On pose h une quantité réelle supposée "petite". Les dérivées premières et secondes sont définies comme suit.

– La dérivée première discrète de f en un point x est définie par l'équation :

$$Df(x) = \frac{f(x+h) - f(x)}{h}$$

– La dérivée seconde discrète de f en un point x est définie par l'équation :

$$D^2 f(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

Lorsque h tend vers 0, on retrouve la dérivée usuelle, définie par la valeur limite du rapport ci-dessus.

5.3 Choix du nombre de classes dans la cadre de bipartitionnement

Plusieurs algorithmes de bipartitionnement nécessitent d'avoir une idée *a priori* sur le nombre de classes sur les lignes et les colonnes. Pour ce faire, nous proposons d'adapter les méthodes proposées à la classification automatique (section 5.2) au cas de la classification simultanée.
Nous testons l'efficacité de ces méthodes sur les mêmes jeux des données générées artificiellement présentés dans le chapitre précédent à la section 4.2.2 autour d'un nombre de classes connus sur les lignes et les colonnes.

5.3.1 Adaptation des indices de validation aux méthodes de bipartitionnement

Comme l'algorithme Croki2 repose sur l'utilisation d'une variante des nuées dynamiques, il nécessite de fixer le nombre de classes sur les lignes et les colonnes. Dans le cas où K et L ne sont pas fixés, le meilleur couple de partitions correspond

aux partitions triviales où chaque élément de l'ensemble X et de l'ensemble Y forment à eux seuls une classe. Dans ce cas,

$$\chi^2(P,Q) = \chi^2(X,Y)$$

Afin d'étendre l'utilisation des indices de validation aux méthodes de bipartitionnement, nous proposons d'appliquer l'indice séparément à la partition obtenue sur les lignes et la partition obtenue sur les colonnes. Ainsi, pour chaque couple de classes nous obtenons deux valeurs de l'indice : une valeur obtenue sur la partition-ligne et une valeur obtenue sur la partition-colonne.
Considérons deux ensembles de données. Le premier, JD3x3, est généré autour de 3 classes artificielles sur les lignes et 3 classes artificielles sur les colonnes. Le second, JD5x4, est généré autour de 5 classes sur les lignes et 4 classes sur les colonnes.
Nous représentons les valeurs de l'indice Baker et Hubert (BH) calculé sur la partition-ligne et la partition-colonne de JD3x3 (fig.5.5). Les points en surbrillance sur les cartes représentent les meilleurs valeurs de l'indice sur les lignes (resp. sur les colonnes).

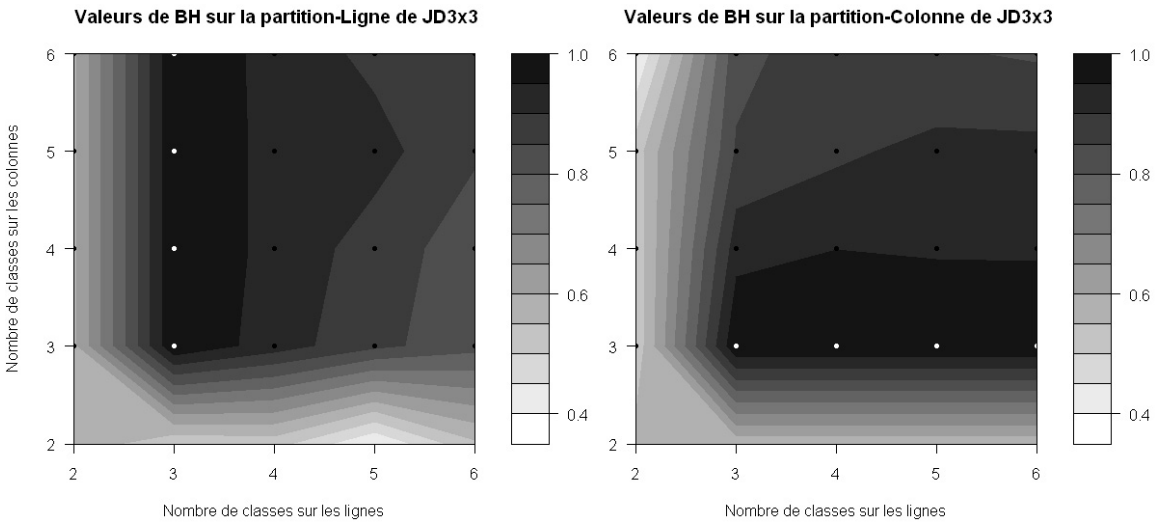


FIGURE 5.5 – Valeurs de l'indice BH (Baker and Hubert) calculé sur la partition-ligne et la partition-colonne du Jeu des données JD3x3

Si la meilleure valeur de l'indice calculé sur les lignes et la meilleure valeur de l'indice calculé sur les colonnes sont obtenues pour le même couple de classes (k^*, l^*) alors nous retenons ce couple comme meilleure solution.
À titre d'exemple, pour le jeu des données JD3x3 la valeur maximale de l'indice

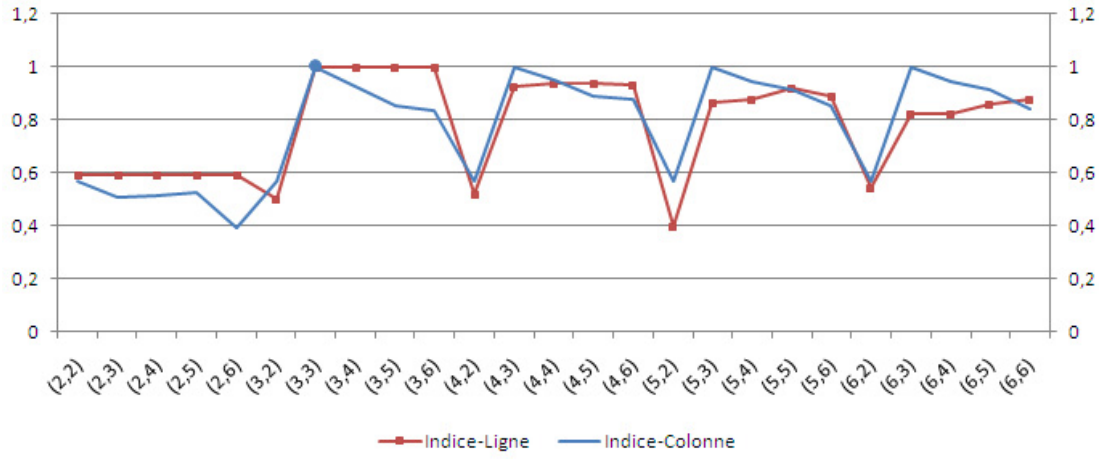


FIGURE 5.6 – Représentation des valeurs de l'indice *BH* (Baker and Hubert) calculé sur la partition-ligne et la partition-colonne du Jeu des données JD3x3.

BH calculé sur les lignes est obtenue pour les couples (3,3),(3,4),(3,5) et (3,6). La valeur maximale de l'indice calculé sur les colonnes est obtenue pour les couples (3,3), (4,3), (5,3) et (6,3) (figure 5.6). Le couple qui maximise à la fois les deux valeurs de l'indice, correspond à la bonne solution. Ainsi, la meilleure bipartition sur JD3x3 est obtenue pour 3 classes sur les lignes et 3 classes sur les colonnes. Graphiquement, la représentation du nuage des points dont l'ordonnée est l'indice-colonne et l'abscisse est l'indice-ligne permet de détecter un point unique correspondant au maximum des deux indices, c'est le point correspondant au couple (3,3) (voir figure 5.7).

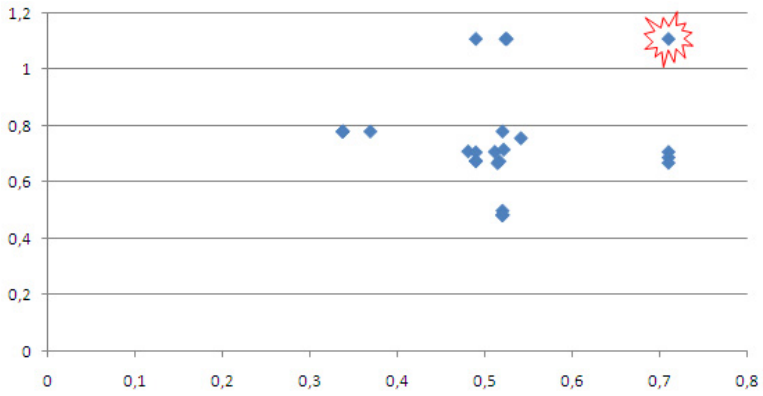


FIGURE 5.7 – Nuage de points représentant l'indice *BH* (Baker and Hubert) appliqué à JD3x3. L'axe des ordonnées représente l'indice-colonne et l'axe des abscisses représente l'indice-ligne.

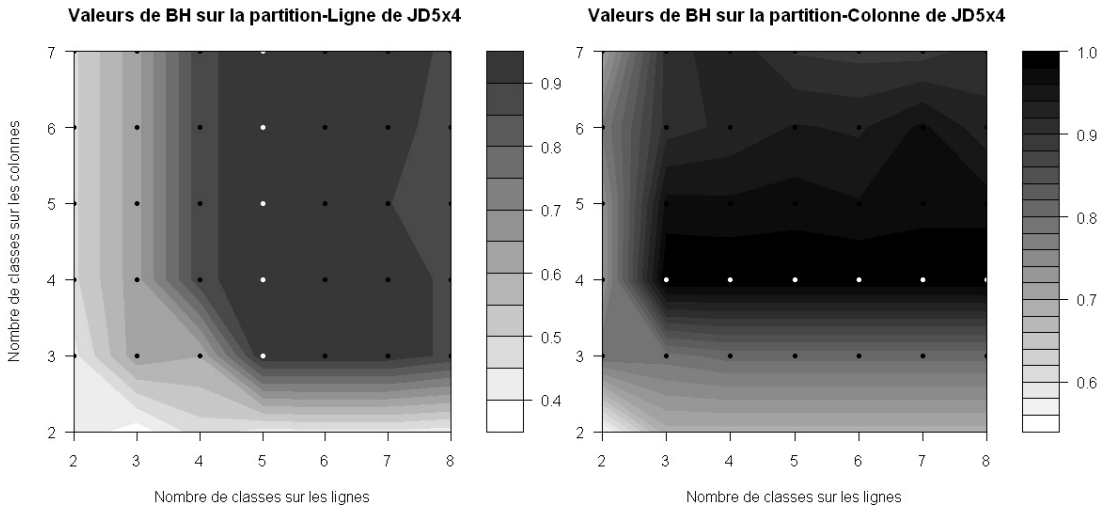


FIGURE 5.8 – Valeurs de l'indice BH (Baker and Hubert) calculé sur la partition-ligne et la partition-colonne du Jeu des données JD5x4.

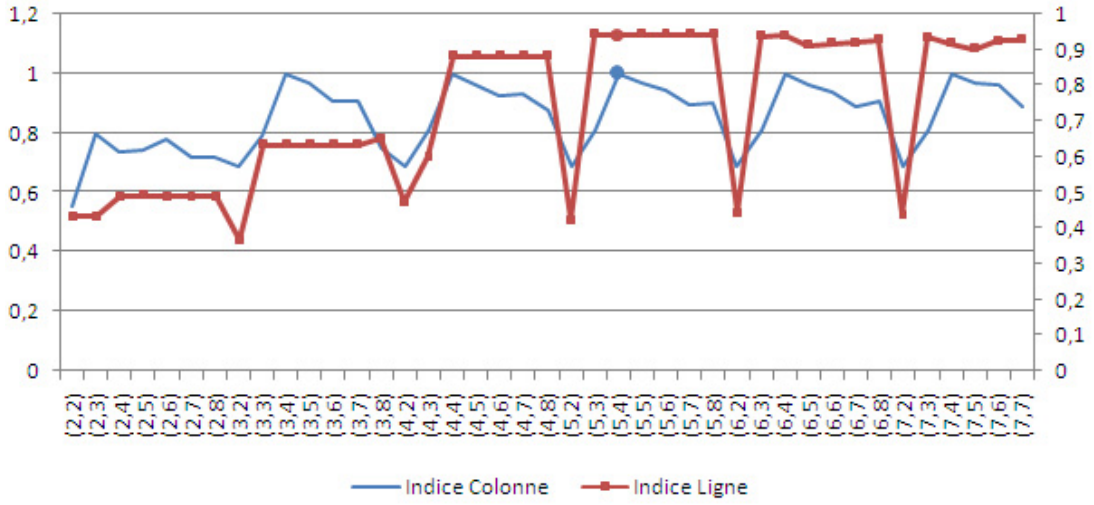


FIGURE 5.9 – Représentation des valeurs de l'indice BH (Baker and Hubert) calculé sur la partition-ligne et la partition-colonne du Jeu des données JD5x4.

En appliquant le même indice sur le jeu des données JD5x4 (fig.5.8 et fig. 5.9), l'indice-ligne prend sa valeur maximale pour les couples

$$\{(5,3),(5,4),(5,5),(5,6),(5,7),(5,8)\}$$

L'indice-colonne prend sa valeur maximale pour les couples

$$\{(3,4),(4,4),(5,4),(6,4),(7,4),(8,4)\}$$

Nous retenons alors le couple (5,4) qui est en commun et qui maximise les deux indices à la fois.

Pour certains jeux des données, en particulier les données réelles, la meilleure valeur de l'indice-ligne et la meilleure valeur de l'indice-colonne sont obtenues pour deux couples de classes différents (aucun couple en commun). Dans ce cas, la solution dépend de l'importance qu'on accorde à la partition ligne et la partition colonne. En d'autres termes, si on privilégie la partition ligne, on choisira le couple de classes qui maximise l'indice-ligne bien qu'il ne correspondait pas à la meilleure valeur de l'indice-colonne et inversement.
Dans l'exemple de la figure 5.10, la meilleure valeur de l'indice-ligne est obtenue pour les couples

$$\{(3,3),(3,4),(3,5),(3,6),(3,7),(3,8),(3,9)\}$$

alors que la meilleure valeur de l'indice-colonne est obtenue pour les couples

$$\{(2,2),(2,3),(4,2),(5,2),(6,2)\}$$

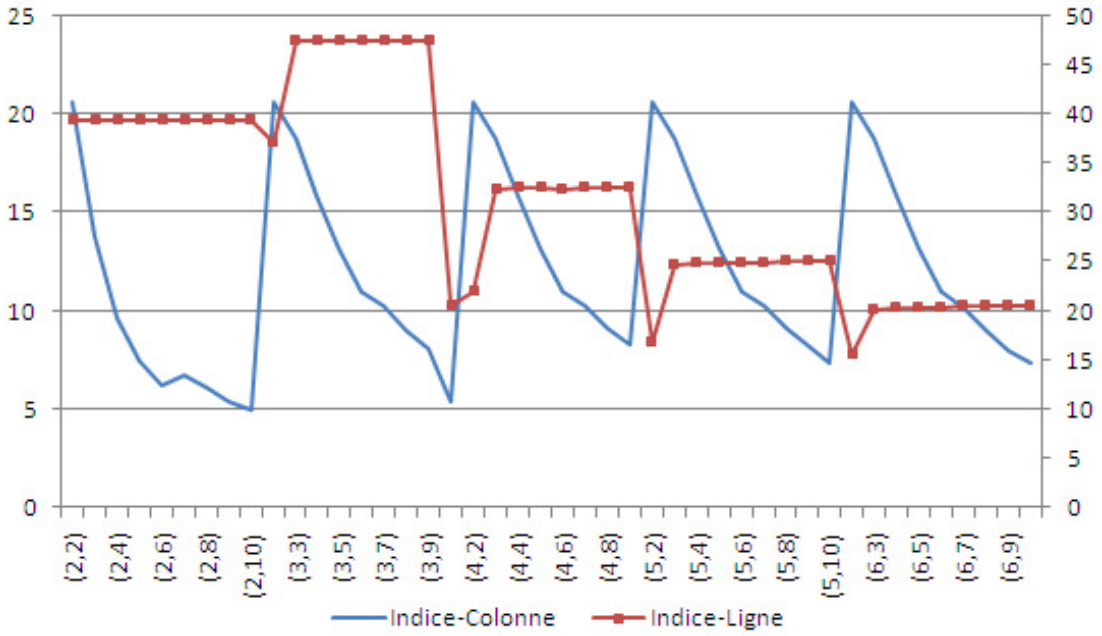


FIGURE 5.10 – Valeurs de l'indice CH (Calinski Harabasz) calculé sur la partition-ligne et la partition-colonne du jeu des données JD3x8. Les deux indices optent pour des couples de classes différents, aucun couple de classes ne maximise les deux indices à la fois.

En représentant le nuage de points de l'indice-colonne en fonction de l'indice-ligne, on repère sur la figure 5.11 plusieurs points candidats. Les points rouges

représentent les meilleurs couples sur la partition-colonne alors que les points verts correspondent aux meilleurs couples sur la partition ligne. Ainsi, plusieurs solutions sont possibles selon que l'on privilégie la partition ligne ou la partition colonne.

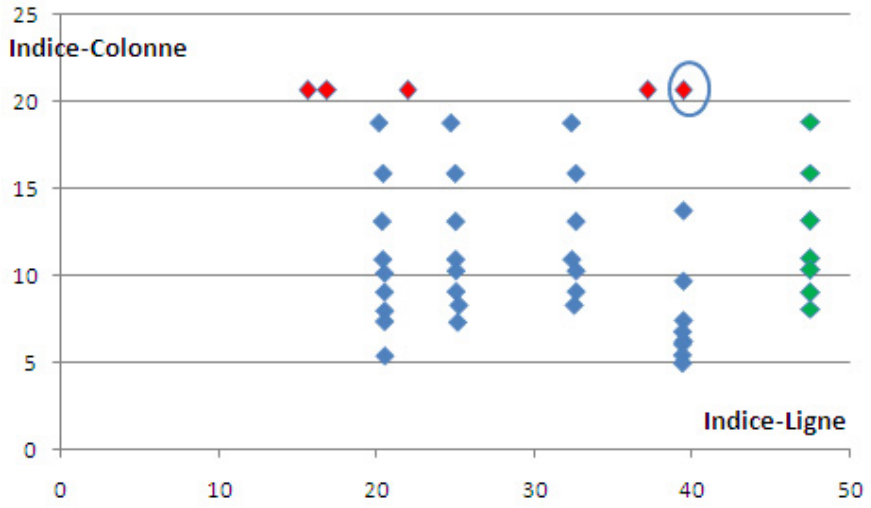


FIGURE 5.11 – Nuage de points représentant l'indice-Colonne CH (Calinski Harabasz) en fonction de l'indice-Ligne CH du jeu des données JD3x8 : plusieurs points candidats.

Dans le cas où on est en présence de plusieurs couples de classes candidats, nous proposons de calculer un indice pondéré dans lequel on attribue un poids à l'indice-ligne et un poids à l'indice-colonne.

$$IndiceGlobal = \alpha \times IndiceLigne + (1 - \alpha) \times IndiceColonne$$

avec $\alpha \in [0, 1]$.
Soient les points $\{A_1(a, b_1), A_2(a, b_2), ..., A_p(a, b_p)\}$ correspondant à la meilleure valeur a de l'indice-ligne et des valeurs différentes de l'indice-colonne, avec
$b_1 > b_2 > ... > b_p$
et les points $\{B_1(a_1, b), B_2(a_2, b), ..., B_q(a_q, b)\}$ correspondant à la meilleure valeur b de l'indice-colonne et des valeurs différentes de l'indice-ligne, avec
$a_1 > a_2 > ... > a_q$
Le choix sera donc effectué entre les deux points $A_1(a, b_1)$ et $B_1(a_1, b)$ selon la valeur de α. Si $A_1 > B_1$, alors $IndiceGlobal(A_1) > IndiceGlobal(B_1)$.

Par conséquent,

$$\alpha a + (1 - \alpha)b_1 > \alpha a_1 + (1 - \alpha)b \Rightarrow \alpha > \frac{(b - b_1)}{(b - b_1) + (a - a_1)} = \alpha_0$$

Ainsi,

- Si $\alpha \in [0, \alpha_0]$ alors le meilleur point c'est $B_1(a_1, b)$.
- Si $\alpha \in]\alpha_0, 1]$ alors le meilleur point c'est $A_1(a, b_1)$.

Dans l'exemple de la figure 5.11, le choix doit être fait entre le couple (3,5) représenté par le point A(47.4, 18.8) et (2,2) représenté par (39.4, 20.6).
$\alpha_0 = 0.183$. Ainsi, pour $\alpha \geq 0.183$, le couple (2,2) représenté par le point encerclé correspond à la meilleure valeur de l'indice donc au bon nombre de classes sur les lignes et les colonnes. Dans le cas contraire, c'est le couple (3,9) qui est choisi.
D'autres stratégies sont bien évidemment possibles pour le choix du meilleur couple de classes.

5.3.2 Résolution graphique/Méthode de la différentielle

Dans l'algorithme Croki2, les meilleures partitions P et Q sur les lignes et les colonnes sont celles qui maximisent le critère de χ^2 du nouveau tableau de contingence obtenu en regroupant les lignes et les colonnes suivant ces deux partitions. Or, plus le nombre de classes sur les lignes et les colonnes augmente, plus la valeur de χ^2 augmente. Cependant, à partir d'une certaine valeur du couple (k, l), par exemple (4,4) pour le jeu des données JD4x4 et (6,3) pour le jeu des données JD6x3 (figure 5.12), le χ^2 stagne ou croit très lentement. Notre objectif est alors d'identifier le couple de classes (k^*, l^*) à partir duquel le χ^2 ne croit plus ou croit plus lentement.

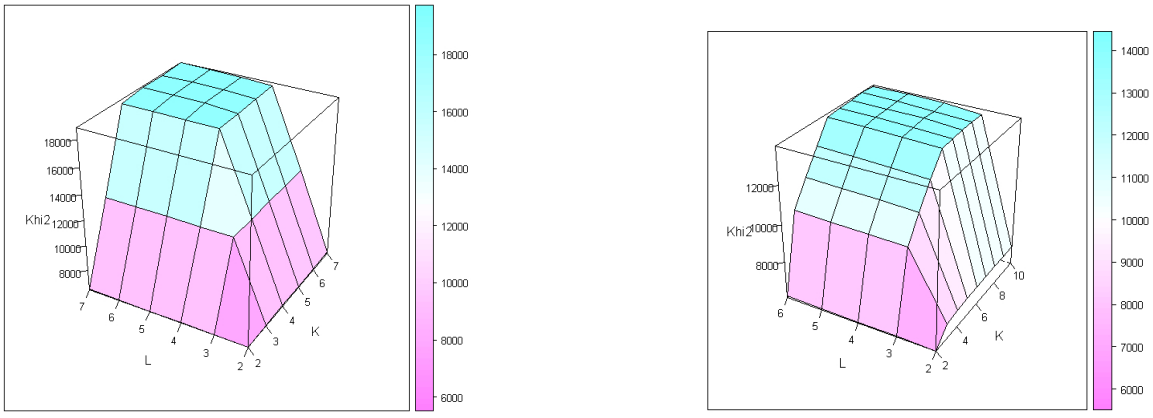


FIGURE 5.12 – χ^2 de contingence en fonction du nombre de classes sur les lignes et les colonnes (Jeux des données JD4x4 à gauche et JD6x3 à droite)

Comme nous remarquons dans la figure 5.12, la fonction χ^2 qu'on notera f(x,y) constitue une surface sur les axes de base x et y représentant respectivement le nombre de classes sur les lignes et le nombre de classes sur les colonnes.
Les définitions de dérivée première et seconde présentées dans la section 5.2.3

ne sont plus valables ; Nous définissons alors dans le cas discret d'une fonction à deux variables les dérivées partielles premières, la différentielle totale, somme des dérivées partielles de la fonction et le Laplacien discret, somme des dérivées secondes discrètes.

- Dans le cas discret, les dérivées partielles premières de f par rapport aux variables x et y sont définies par les équations suivantes :

$$\frac{df}{dx}(x,y) = \frac{f(x+h,y) - f(x,y)}{h}$$

$$\frac{df}{dy}(x,y) = \frac{f(x,y+h) - f(x,y)}{h}$$

- La différentielle totale de f en (x,y) est la somme des dérivées partielles de la fonction f :

$$Df(x,y) = \frac{f(x+h,y) - f(x,y)}{h} + \frac{f(x,y+h) - f(x,y)}{h}$$

  Géométriquement, la différentielle totale correspond au déplacement total selon les déplacements infinitésimaux le long de chacun des axes.

- Le Laplacien discret est la somme des dérivées secondes partielles selon x et y. Il est défini par l'équation suivante :

$$\Delta f = \frac{[f(x+h,y) + f(x-h,y) - 2f(x,y)]}{h^2}$$

$$+\frac{[f(x,y+h) + f(x,y-h) - 2f(x,y)]}{h^2}$$

  Cette quantité est proportionnelle à la différence entre la somme des valeurs extrêmes et la valeur centrale. La propriété se généralise à un nombre quelconque de variables.
  Le laplacien d'une fonction peut aussi être interprété comme la courbure moyenne locale de la fonction, que l'on visualise aisément pour une fonction à une seule variable $f(x)$.

- Dans le cas continu, la différentielle d'ordre 2 d'une fonction réelle à deux variables $\{x,y\}$ s'écrit :

$$d^2f = \frac{\partial^2 f}{\partial x^2}(dx)^2 + \frac{\partial^2 f}{\partial x \partial y}dxdy + \frac{\partial^2 f}{\partial y \partial x}dydx + \frac{\partial^2 f}{\partial y^2}(dy)^2$$

  Afin de discrétiser ces termes, nous avons recourt aux développements limités de Taylor.

$$f(x+h,y) = f(x,y) + \frac{df}{dx}(x,y)h + \frac{d^2f}{dx^2}(x,y)\frac{h^2}{2!} + \frac{d^3f}{dx^3}(x,y)\frac{h^3}{3!} + \dots$$

$$f(x-h,y) = f(x,y) - \frac{df}{dx}(x,y)h + \frac{d^2f}{dx^2}(x,y)\frac{h^2}{2!} - \frac{d^3f}{dx^3}(x,y)\frac{h^3}{3!} + \dots$$

En combinant les deux équations et en se limitant au développement limité d'ordre 3, on a :

$$\frac{d^2f}{dx^2}(x,y) = \frac{f(x+h,y) - 2f(x,y) + f(x-h,y)}{h^2}$$

Calculons maintenant les dérivées secondes croisées dans le cas discret. Dans le cas discret, la dérivée partielle de f par rapport à x s'écrit :

$$\frac{\partial f}{\partial x}(x,y) = \frac{f(x+h,y) - f(x,y)}{h}$$

$$\frac{\partial f}{\partial y}(\frac{\partial f}{\partial x})(x,y) = \frac{f(x+h,y+h) - f(x+h,y)}{h^2} - \frac{f(x,y+h) - f(x,y)}{h^2}$$

$\Rightarrow$

$$\frac{\partial^2 f}{\partial y \partial x}(x,y) = \frac{f(x+h,y+h) - f(x+h,y) - f(x,y+h) + f(x,y)}{h^2}$$

D'autre part, la dérivée partielle de f par rapport à y s'écrit :

$$\frac{\partial f}{\partial y}(x,y) = \frac{f(x,y+h) - f(x,y)}{h}$$

$$\frac{\partial f}{\partial x}(\frac{\partial f}{\partial y})(x,y) = \frac{f(x+h,y+h) - f(x,y+h)}{h^2} - \frac{f(x+h,y) + f(x,y)}{h^2}$$

$\Rightarrow$

$$\frac{\partial^2 f}{\partial y \partial x}(x,y) = \frac{f(x+h,y+h) - f(x+h,y) - f(x,y+h) + f(x,y)}{h^2}$$

Ainsi, la différentielle d'ordre 2 s'écrit :

$$d^2f(x,y) = \frac{2f(x+h,y+h) + f(x-h,y) + f(x,y-h)}{h^2}$$
$$- \frac{f(x+h,y) + f(x,y+h) + 2f(x,y)}{h^2}$$

Pour trouver le couple de classes à partir duquel les valeurs de χ^2 ne croissent plus, nous calculons la différentielle de la fonction de χ^2 pour tous les couples de classes. Le meilleur couple est celui qui correspond à la valeur la plus faible, ou la plus élevée en valeur absolue, de la différentielle d'ordre 2 (voir les figures 5.13 et 5.14).

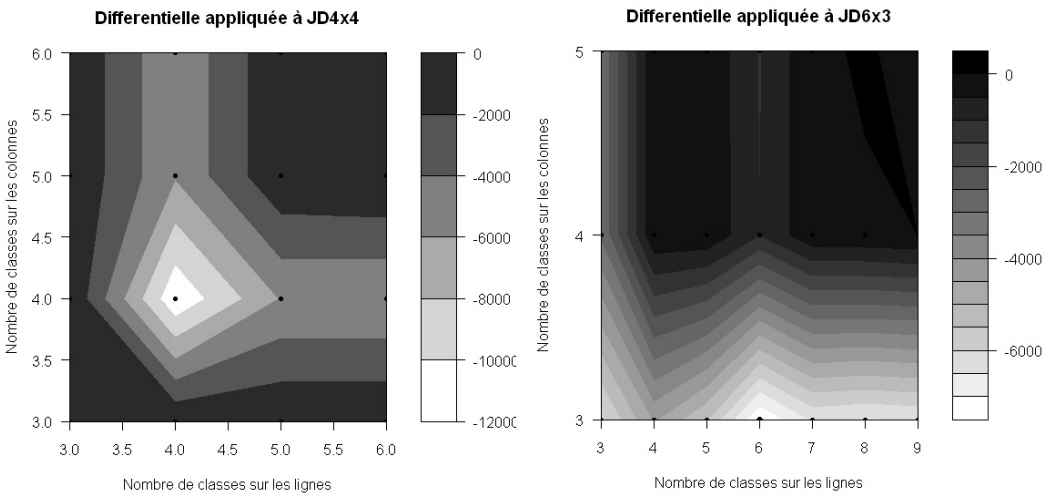


FIGURE 5.13 – Représentation graphique de la différentielle d'ordre 2 en fonction du nombre de classes sur les lignes et les colonnes des jeux des données JD4x4 et JD6x3.La valeur la plus faible de la différentielle est obtenue pour le couple (6,3) dans JD6x3 et le couple (4,4) dans JD4x4.

Dans le cas du JD3x8, la valeur la plus faible de la différentielle est obtenue pour le couple (3,4). Ainsi, nous remarquons que dans le cas où le nombre de classes sur les lignes et le nombre de classes sur les colonnes sont très différents, la différentielle échoue dans la détermination du bon couple de classes dans les données. D'autre part, la différentielle débute à partir du couple (3,3) puisque la formule de la différentielle comporte les termes $(x - h)$ et $(y - h)$. Donc avec ce critère, il n'est pas possible de détecter le couple (2,2) dans le cas où il correspond au bon couple de classes.

5.4 Expérimentations et résultats

Sachant que les deux versions de l'algorithme nécessitent de fixer le nombre de classes sur les lignes et les colonnes, des expérimentations sont effectuées pour tester l'ensemble des méthodes d'identification du bon nombre de classes citées précédemment et opter pour la méthode la plus adaptée aux données représentant le contenu textuel d'un site Web.

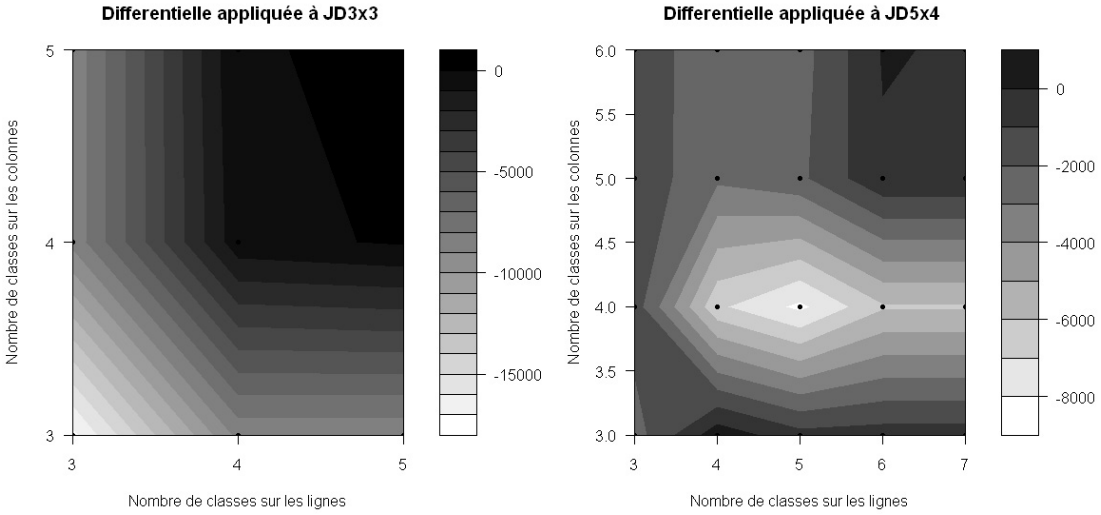


FIGURE 5.14 – Représentation graphique de la différentielle d'ordre 2 en fonction du nombre de classes sur les lignes et les colonnes des jeux des données JD3x3 et JD5x4. La valeur la plus faible de la différentielle est obtenue pour le couple (3,3) dans JD3x3 et le couple (5,4) dans JD5x4.

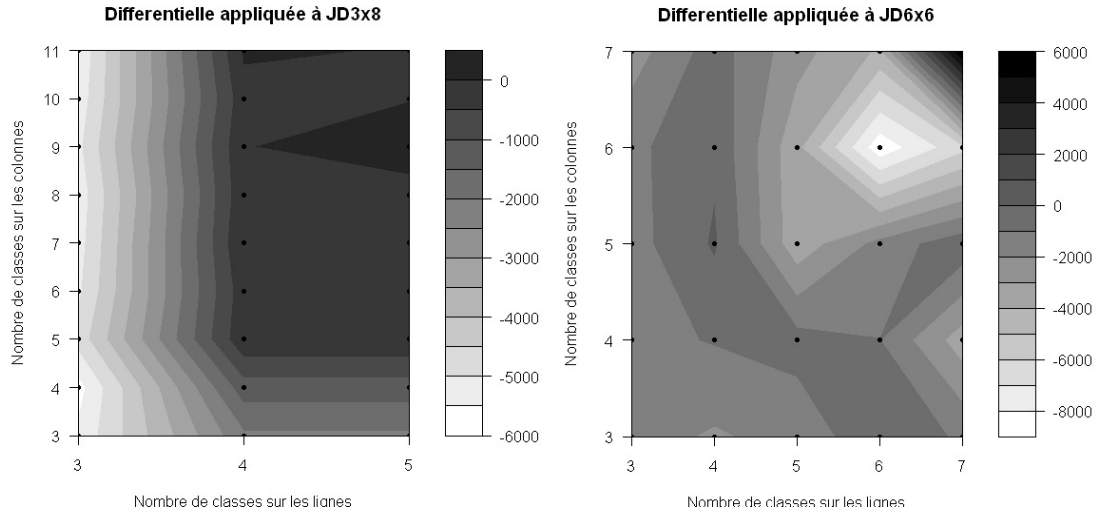


FIGURE 5.15 – Représentation graphique de la différentielle d'ordre 2 en fonction du nombre de classes sur les lignes et les colonnes des jeux des données JD3x8 et JD6x6. La valeur la plus faible de la différentielle est obtenue pour le couple (6,6) dans JD6x6. Mais dans le cas de JD3x8, les valeurs la plus faible est obtenue pour le couple (3,4) qui est différent du bon couple de classes (3,8).

Le tableau 5.5 présente les résultats de l'application des indices aux partitions obtenues sur les lignes et les colonnes par Croki2 OA et Croki2 OG appliqués au jeu des données JD3x3. Ce jeu de données présente le même nombre de classes sur les lignes et les colonnes.
D'après ce tableau, les indices *Dunn*, *BH*, *DB*, *CH* et *Silhouette* ont réussi à identifier le bon nombre de classes sur les lignes et les colonnes. Par contre, les indices *HL* et *KL* proposent des solutions différentes pour la partition-ligne et la partition-colonne.
La croix dans le tableau 5.5 indique l'absence d'un couple en commun entre la solution proposée pour la partition-Ligne et la solution proposée pour la partition-Colonne.

TABLE 5.5 – Application des indices aux partitions obtenues par Croki2 OA et Croki2 OG appliqués à JD3x3.

Indices	**Partition-Ligne**	**Partition-Colonne**	**Solution**
Dunn (Croki2 OA)	**(3,3)**(3,4)(3,5)	**(3,3)**(4,3)(5,3)	**(3,3)**
Dunn (Croki2 OG)	**(3,3)**(3,4)(3,5)(3,6)	**(3,3)**(4,3)(5,3)(6,3)	**(3,3)**
BH (Croki2 OA)	**(3,3)**(3,4)(3,5)	**(3,3)**(4,3)(5,3)	**(3,3)**
BH (Croki2 OG)	**(3,3)**(3,4)(3,5)(3,6)	**(3,3)**(4,3)(5,3)(6,3)	**(3,3)**
HL (Croki2 OA)	(6,2)	(2,6)	x
HL (Croki2 OG)	(6,2)	(4,6)	x
KL Croki2 OA)	(4,2)	(3,2)	x
KL (Croki2 OG)	(6,3)	(3,3)	x
DB (Croki2 OA)	**(3,3)**(3,4)(3,5)	**(3,3)**(3,4)(3,5)	**(3,3)**
DB (Croki2 OG)	**(3,3)**(3,4)(3,5)(3,6)	**(3,3)**(4,3)(5,3)(6,3)	**(3,3)**
CH (Croki2 OA)	**(3,3)**(3,4)(3,5)	**(3,3)**(3,4)(3,5)	**(3,3)**
CH (Croki2 OG)	**(3,3)**(3,4)(3,5)(3,6)	**(3,3)**(4,3)(5,3)(6,3)	**(3,3)**
S (Croki2 OA)	**(3,3)**(3,4)(3,5)	**(3,3)**(3,4)(3,5)	**(3,3)**
S (Croki2 OG)	**(3,3)**(3,4)(3,5)(3,6)	**(3,3)**(4,3)(5,3)(6,3)	**(3,3)**

Le tableau 5.6 présente les résultats de l'application des indices aux partitions obtenues sur les lignes et les colonnes par Croki2 OA et Croki2 OG appliqués au jeu des données JD5x4.
D'après ce tableau, seuls les indices *Dunn* et *BH* ont réussi à identifier le bon nombre de classes sur les lignes et les colonnes (5,4). Par contre, les indices *DB*,

CH et S proposent le couple (4,4) comme solution commune entre la partition-ligne et la partition-colonne du JD5x4. Les indices HL et KL, comme dans l'exemple précédent, présentent des solutions différentes entre la partition-ligne et la partition-colonne.
La croix dans le tableau 5.6 indique l'absence d'un couple en commun entre la solution proposée pour la partition-Ligne et la solution proposée pour la partition-Colonne.

TABLE 5.6 – Application des indices aux partitions obtenues par Croki2 OA et Croki2 OG appliqués à JD5x4.

Indices	**Partition-Ligne**	**Partition-Colonne**	**Solution**
Dunn (Croki2 OA/OG)	(5,3)(5,4)(5,5) (5,6)(5,7)(5,8)	(3,4) (4,4) (5,4) (6,4) (7,4) (8,4)	**(5,4)**
BH (Croki2 OA/OG)	(5,3)(5,4)(5,5) (5,6)(5,7)(5,8)	(3,4) (4,4) (5,4) (6,4) (7,4) (8,4)	**(5,4)**
HL (Croki2 OA/OG)	(6,6)	(2,6)	x
KL (Croki2 OA/OG)	(7,3)	(2,6)	x
DB (Croki2 OA/OG)	(4,4)(4,5)(4,6) (4,7)(4,8)	(3,4)(4,4)(5,4) (6,4)(7,4)(8,4)	**(4,4)**
CH (Croki2 OA/OG)	(4,4)(4,5)(4,6) (4,7)(4,8)	(3,4)(4,4)(5,4) (6,4)(7,4)(8,4)	**(4,4)**
S (Croki2 OA/OG)	(4,4)(4,5)(4,6) (4,7)(4,8)	(3,4)(4,4)(5,4) (6,4)(7,4)(8,4)	**(4,4)**

Les résultats de l'application de ces indices, ainsi que les indices de validation interne (l'indice de Cramer, le χ^2 pénalisé et l'indice de Tschuprow) et la méthode de la différentielle à tous les jeux des données sont présentés dans le tableau 5.7.

Les résultats des expérimentations montrent que :

– les indices de validation externe, *Dunn*, *DB*, *CH* et *S* sont performants dans l'identification du bon nombre de classes sur les lignes et les colonnes lorsque les données présentent une structure diagonale c'est-à-dire que le nombre de classes sur les lignes est égal au nombre de classes sur les colonnes. En effet, DB, CH et S ont échoué dans l'identification du bon couple de classes dans le cas des jeux des données JD5x4, JD6x3 et JD3x8 alors qu'ils ont réussi dans le cas des jeux des données JD3x3, JD4x4 et JD6x6.

TABLE 5.7 – Comparaison des indices sur les données simulées

Jeu des données	**JD3x3**	**JD4x4**	**JD5x4**	**JD6x3**	**JD6x6**	**JD3x8**
Dunn (Croki2 OA)	**(3,3)**	**(4,4)**	**(5,4)**	x	**(6,6)**	x
Dunn (Croki2 OG)	**(3,3)**	**(4,4)**	**(5,4)**	x	**(6,6)**	x
BH (Croki2 OA)	**(3,3)**	**(4,4)**	**(5,4)**	**(6,3)**	**(6,6)**	(3,4)
BH (Croki2 OG)	**(3,3)**	**(4,4)**	**(5,4)**	**(6,3)**	**(6,6)**	(3,6)
HL (Croki2 OA)	x	x	x	x	x	x
HL (Croki2 OG)	x	x	x	x	x	x
KL (Croki2 OA)	x	x	x	x	x	x
KL (Croki2 OG)	x	x	x	x	x	x
DB (Croki2 OA)	**(3,3)**	**(4,4)**	(4,4)	x	**(6,6)**	(3,3)
DB (Croki2 OG)	**(3,3)**	**(4,4)**	(4,4)	x	**(6,6)**	(3,5)
CH (Croki2 OA)	**(3,3)**	**(4,4)**	(4,4)	x	x	x
CH (Croki2 OG)	**(3,3)**	**(4,4)**	(4,4)	x	**(6,6)**	x
S (Croki2 OA)	**(3,3)**	**(4,4)**	(4,4)	x	**(6,6)**	(3,3)
S (Croki2 OG)	**(3,3)**	**(4,4)**	(4,4)	x	**(6,6)**	(3,5)
T (Croki2 OA)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)
T (Croki2 OG)	(2,2)	(2,2)	(2,2)	(2,2)	(3,3)	(2,2)
C (Croki2 OA)	(6,6)	(7,7)	(8,7)	(9,6)	(8,8)	(6,10)
C(Croki2 OG)	(6,6)	(7,7)	(8,7)	(10,6)	(9,9)	(6,10)
$\chi^2 p$ (Croki2 OA)	**(3,3)**	**(4,4)**	**(5,4)**	(3,3)	(5,5)	(3,4)
$\chi^2 p$ (Croki2 OG)	**(3,3)**	**(4,4)**	**(5,4)**	(3,3)	(5,5)	(3,4)
Laplacien (Croki2 OA)	**(3,3)**	**(4,4)**	**(5,4)**	**(6,3)**	(3,3)	(3,9)
Laplacien (Croki2 OG)	**(3,3)**	**(4,4)**	**(5,4)**	**(6,3)**	(3,3)	(3,4)
Differentielle (Croki2 OA)	**(3,3)**	**(4,4)**	**(5,4)**	**(6,3)**	**(6,6)**	(3,4)
Differentielle (Croki2 OG)	**(3,3)**	**(4,4)**	**(5,4)**	**(6,3)**	**(6,6)**	(3,4)

- L'indice BH par contre a réussi à identifier le bon couple de classes dans tous les cas sauf lorsque la différence entre le nombre de classes sur les lignes et celui sur les colonnes est grande (cas du JD3x8).
- Le Laplacien a donné des résultats satisfaisants lorsque le nombre de classes sur les lignes et les colonnes est réduit. Mais plus ce nombre augmente plus la fiabilité de cet indice diminue. C'est le cas aussi de l'indice de χ^2 pénalisé.
- La différentielle s'avère assez efficace pour l'identification de la structure initiale des données. En effet, elle a réussi à estimer le bon nombre de classes dans tous les jeux des données, sauf dans JD3x8. Dans ce dernier cas, elle propose le couple (3,4) comme le meilleur au lieu du (3,8).

5.5 Synthèse

Dans le cadre de la classification croisée, les résultats expérimentaux ont montré que l'indice BH et la différentielle sont les meilleurs critères de validation dans l'ensemble des indices utilisés dans cette étude. L'avantage de cette stratégie d'estimation du nombre de classes est qu'elle n'est pas spécifique à l'algorithme Croki2. Elle est applicable à tout algorithme de bipartitionnement nécessitant de fixer à priori le nombre de classes dans les données.
Nous confirmons également d'après les expérimentations sur l'identification du bon nombre de classes que les deux algorithmes Croki2 OA et Croki2 OG proposent toujours les mêmes solutions. Dans le chapitre suivant, nous travaillons avec des données réelles issues du contenu et de l'usage d'un site Web. Les deux critères d'estimation du nombre de classes retenues dans le cas des données artificielles sont alors appliqués aux données réelles afin de nous donner une idée sur le nombre de classes sur les lignes et les colonnes.

Chapitre 6

Une nouvelle approche WCUM pour l'analyse d'un site Web

6.1 Introduction

Notre contribution dans cette thèse réside principalement dans l'association de l'analyse du contenu à l'analyse de l'usage. L'approche WCUM que nous proposons dans ce chapitre consiste à analyser le contenu d'un site Web indépendamment de l'usage en appliquant l'analyse sémantique latente (ASL) et des techniques de block clustering. Cette phase d'analyse textuelle a pour objectif de réduire la quantité d'information contenue dans le site Web en un groupe de thèmes qui pourraient susciter l'intérêt des internautes. La seconde phase dans l'approche WCUM consiste à analyser le comportement des utilisateurs vis-à-vis de ces thèmes.

6.2 Approche de Web Content and Usage Mining

Sachant que le comportement des utilisateurs sur un site Web dépend fortement du contenu des pages du site et inversement le contenu du site devrait répondre aux attentes des usagers, nous proposons de faire la liaison entre le contenu et l'usage d'un site Web. Notre idée est d'exploiter les différentes informations relatives au contenu d'un site Web et de son usage en vue de l'analyser. L'approche WCUM que nous proposons relie l'analyse du contenu à l'analyse de l'usage d'un site Web (fig.6.1). Elle se déroule en deux étapes principales. La première consiste à l'analyse textuelle d'un site Web, indépendamment de son usage, afin de découvrir les thèmes des pages qui le composent. La seconde étape consiste à introduire ces thèmes dans l'analyse de l'usage du site. L'application de cette approche nécessite d'une part l'aspiration du site afin de transformer

ses pages en fichiers texte, et d'autre part la collecte des fichiers Logs contenant la trace des utilisateurs sur le site. Le point de départ de cette approche est le contenu textuel du site et les fichiers logs.

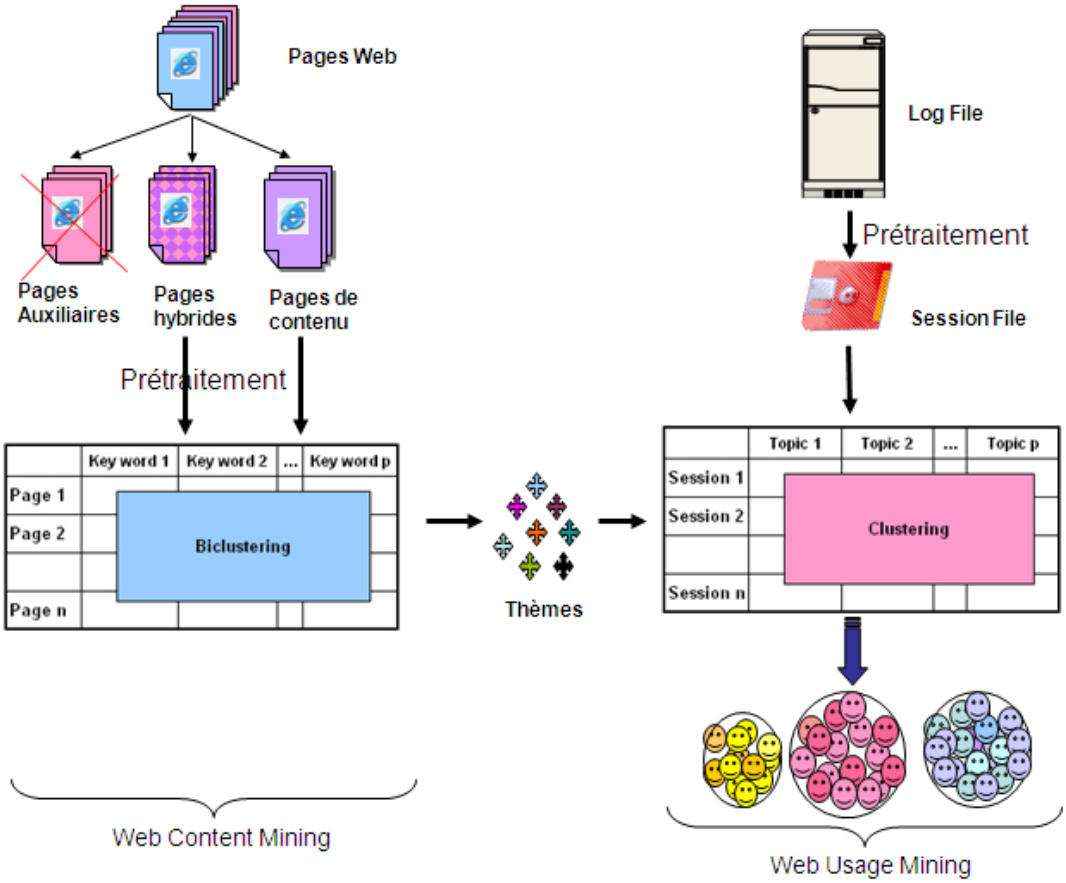


FIGURE 6.1 – ApprocheWCUM

6.3 Analyse textuelle d'un site Web

L'analyse textuelle d'un site Web se déroule en trois étapes principales. La première est celle de la classification des pages Web en pages de contenu présentant une information éventuellement utile aux utilisateurs, pages auxiliaires (ou de navigation) servant à faciliter la navigation sur le site et pages hybrides ayant les caractéristiques communes des deux classes précédentes. Cette étape a pour objectif d'exclure les pages auxiliaires de l'analyse et de limiter le travail de prétraitement aux pages de contenu. La deuxième étape consiste au prétraitement

linguistique et à la sélection de descripteurs des pages de contenu et des pages hybrides afin de résumer l'information contenue dans le site dans une matrice croisant les descripteurs aux pages du site. Un algorithme de block clustering est ensuite appliqué à la matrice pour découvrir des biclasses de pages et de descripteurs permettant d'attribuer un thème à chaque groupe de pages.

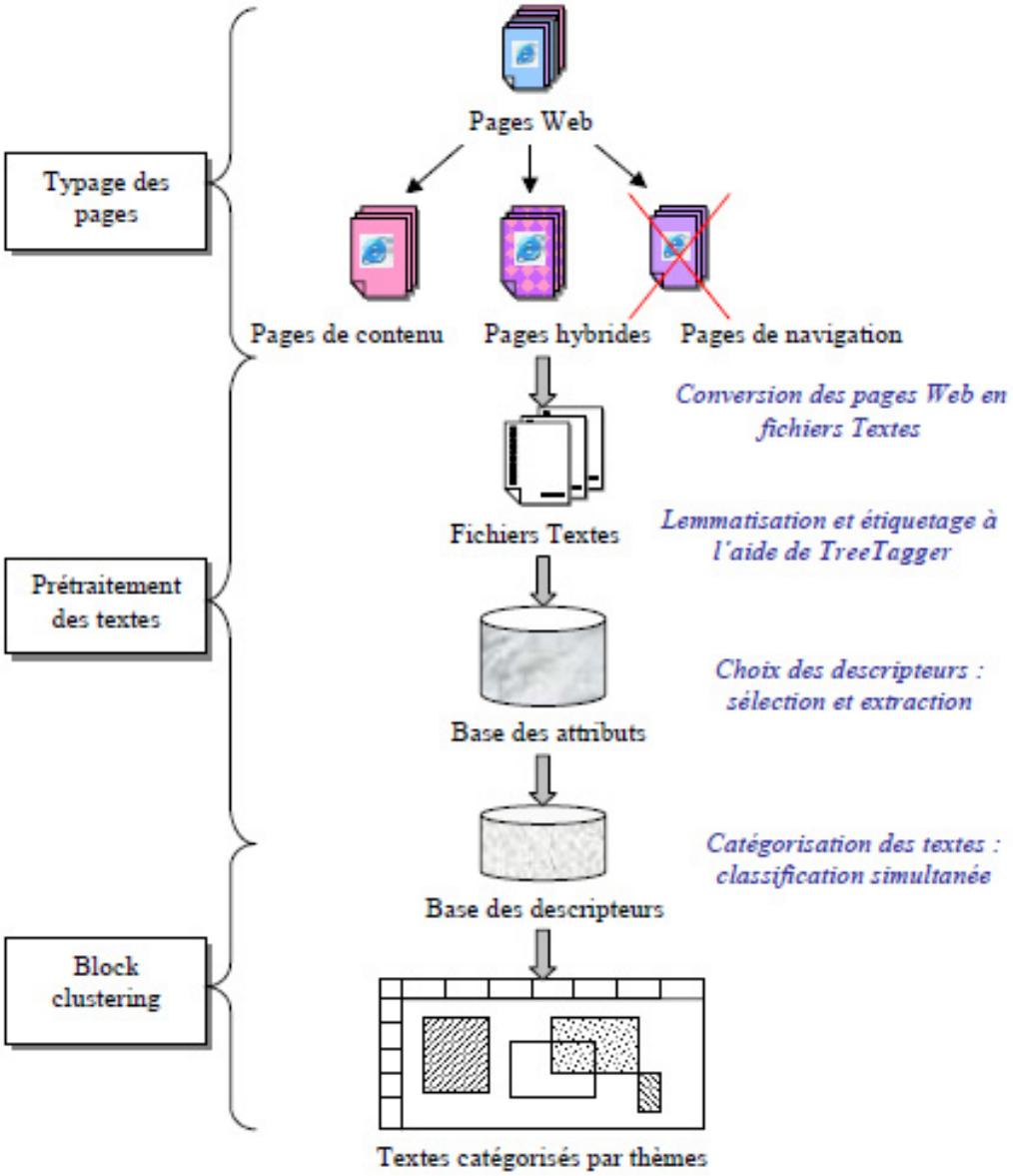


FIGURE 6.2 – Analyse textuelle

6.3.1 Typage des pages

L'objectif de cette étape est de distinguer les pages qui servent à faciliter la navigation sur le site, appelées *pages de navigation* ou *pages auxiliaires*, des pages contenant de l'information qui pourrait intéresser l'internaute. Ces pages sont appelées "*pages de contenu*". Certaines pages Web sont à la fois des pages de contenu et des pages de navigation. Ce sont les *pages hybrides*. Les figures 5.3, 5.4 et 6.5 présentent des exemples de pages de contenu, de pages de navigation et de pages hybrides.
Dans le cas général, les pages de navigation sont des pages de liens. Elles présentent un nombre important de liens entrants et de liens sortants permettant de passer d'une page à une autre. Les pages de contenu, par contre, présentent du contenu textuel et un nombre important de liens entrants puisque plusieurs pages de navigation pointent vers elles.

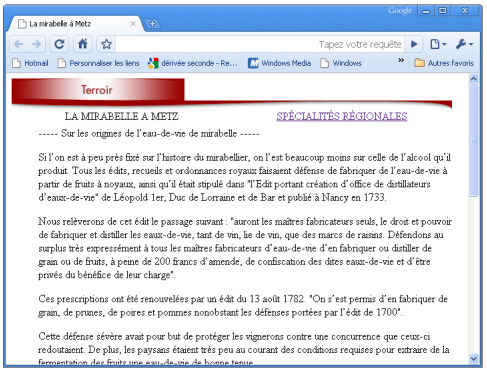


FIGURE 6.3 – Exemple de page de contenu.

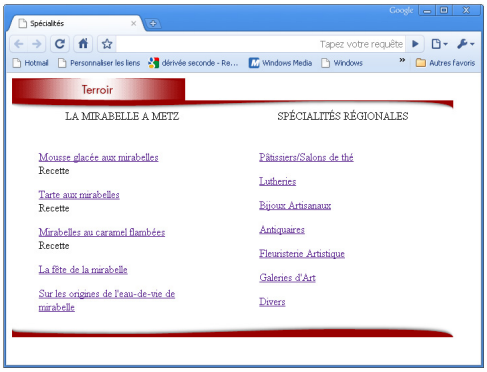


FIGURE 6.4 – Exemple de page de navigation.

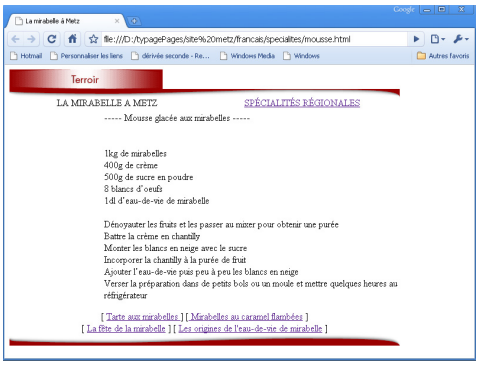


FIGURE 6.5 – Exemple de page hybride.

Dans [Kle99], le modèle de Kleinberg proposé pour améliorer les résultats des systèmes de recherche d'information (information retrieval) distingue les "authorities" qui sont des pages recevant beaucoup de liens des "hubs" qui sont des pages contenant beaucoup de liens vers de bonnes pages. Il est possible pour une page d'être les deux à la fois, mais certaines pages peuvent n'être qu'une bonne page de contenu ou qu'une bonne liste de liens. Cela dépend de la requête de l'utilisateur et du thème de la page. Kleinberg propose alors un algorithme HITS qui associe à chaque page un score Hubs et un score Authorities en réponse à toute requête d'un utilisateur.
Nous proposons d'appliquer la même stratégie mais dans un contexte différent. En fait, dans notre cas, la classification en pages de contenu et pages de navigation dépend seulement du contenu de la page. Par suite, les scores Hubs et Authorities, qui dépendent de la requête de l'utilisateur dans le modèle de Kleinberg, sont dans notre cas invariables pour chaque page sauf s'il y'a eu modification du contenu de la page par le webmaster.
D'autre part, nous proposons de recourir à une méthode de classification simple reposant sur l'utilisation d'un algorithme de classification tel que le Kmeans et un ensemble de variables, à savoir : le nombre de liens entrants (Inlinks), le nombre de liens sortants (Outlinks), la présence ou l'absence d'un contenu textuel dans la page et la taille du fichier (Size).

6.3.1.1 Calcul du nombre de liens entrants et sortants

Un site web est généralement représenté par un graphe dirigé $G = (V, E)$ où V est l'ensemble des noeuds (ou sommets) du graphe. Ces noeuds correspondent aux pages du site. E est l'ensemble des arcs du graphe. Ces arcs représentent les liens entre les pages du site (voir l'exemple de la figure 6.6).

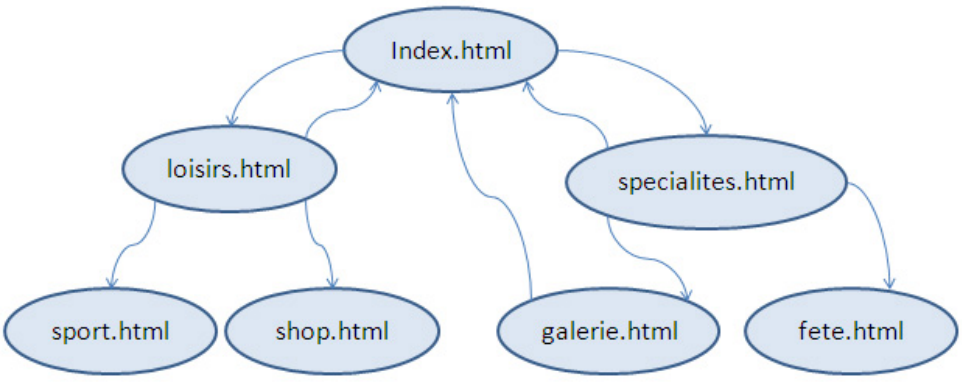


FIGURE 6.6 – Exemple de graphe dirigé représentant la liaison entre des pages d'un site

Ce graphe représentant la topologie du site Web peut être traduit par une matrice. La représentation matricielle présente l'avantage de simplifier la représentation sous forme de graphe qui devient incompréhensible dans le cas où le nombre de liens entre les pages est élevé. Chaque ligne de la matrice correspond à un noeud du graphe et représente une page du site. Il en est de même pour chaque colonne. Ainsi, si le site est composé de N pages différentes, la matrice d'adjacence sera de dimension (N, N). Chaque entrée (i,j) de la matrice prend la valeur 1 si la page i pointe vers la page j (présence d'un lien dans la page i qui pointe vers la page j) et la valeur 0 sinon. Cette matrice est utilisée pour calculer le nombre d'inlinks (nombre d'hyperliens qui mènent à la page en question à partir des autres pages) et le nombre d'outlinks (nombre d'hyperliens dans la page qui mènent vers d'autres pages). Ainsi, le nombre d'inlinks est le total sur les lignes alors que le nombre d'outlinks est le total sur les colonnes. Le tableau 6.1 représente la matrice d'adjacence du graphe de la figure 6.6.

TABLE 6.1 – Matrice d'adjacence

	index	loisirs	spécialités	shop	sport	galerie	fête	**Outlinks**
index	0	1	1	0	0	0	0	**2**
loisirs	1	0	0	1	1	0	0	**3**
spécialités	1	0	0	0	0	1	1	**3**
shop	0	0	0	0	0	0	0	**0**
sport	0	0	0	0	0	0	0	**0**
galerie	0	0	0	0	0	0	0	**0**
fête	0	0	0	0	0	0	0	**0**
Inlinks	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**8**

6.3.1.2 Algorithme HITS [Kle99]

Soit la requête d'un utilisateur notée σ. Les étapes de l'algorithme HITS sont les suivantes :

- Effectuer une recherche classique (en utilisant par exemple un modèle vectoriel avec $tf.idf$). Les pages les plus pertinentes retrouvées sont notées,R_σ.
- À partir de l'ensemble des pages trouvées R_σ, un plus grand ensemble S_σ est construit à travers :
  - les pages qui contiennent des liens vers R_σ.
  - les pages pointées par un lien se trouvant sur une page de R_σ.
- Ayant S_σ et R_σ, le score "authority" $(a(p))$ et le score "hub" $(h(p))$ sont calculés pour chaque page $p \in S_\sigma$. La mesure "authority" quantifie la qualité de la page en tant que page recevant des liens ou des "inlinks", c'est à dire en tant que page de contenu. Une bonne page de contenu obtiendra beaucoup de liens de la part de bonnes pages de liens. Par contre, la mesure "hub"

quantifie le statut de la page en tant que page de liens ou d'"outlinks". Une bonne page de liens pointe vers de bonnes pages de contenu. $a(p)$ et $h(p)$ sont normalisés de manière à ce que la somme de leurs valeurs au carré soit unitaire.

Ainsi les pages qui sont des "autorités" par rapport à la requête de l'utilisateur sont identifiées. Il suffit d'offrir à l'utilisateur les pages ayant le meilleur score $a(p)$.

6.3.1.3 Typage par la méthode des Hubs et des Authorities

Le calcul des scores "Authority" et "Hub" est effectué à l'aide des valeurs propres et vecteurs propres de la matrice d'adjacence multipliée par son transposé. En effet, le score "Authority" est le vecteur propre correspondant à la plus grande valeur propre de la matrice A^tA. Par contre, le score "Hub" est le vecteur propre correspondant à la plus grande valeur propre de la matrice AA^t [Kle99].
A chaque page, deux scores sont associés. Si le score "Authority" est supérieur au score "Hub", alors la page Web est considérée comme une page de contenu, sinon elle est considérée comme une page de navigation. Lorsque les deux scores sont proches l'un de l'autre, la page possède à la fois la qualité d'une page de contenu et d'une page de navigation. Nous proposons alors de fixer un seuil S au dessous duquel la page est considerée comme une page hybride.

- Si $|score"Authority" - score"Hub"| \leq S$ alors la page P est une page hybride.
- Si $score"Authority" - score"Hub" > S$ alors la page P est une page de contenu.
- Si $score"Hub" - score"Authority" > S$ alors la page P est une page de navigation.

6.3.1.4 Typage par classification

Nous proposons de comparer les résultats de l'adaptation de l'algorithme HITS au contexte de typage des pages à ce que donnerait une simple classification appliquée aux pages caractérisées par un ensemble de variables. Comme nous nous intéressons à l'analyse textuelle du site indépendamment de l'usage, nous utilisons les variables suivantes : le nombre de liens entrants (inlinks), le nombre de liens sortants internes ou externes au site (outlinks), la taille du fichier (size) et la présence d'un contenu textuel (texte).
La variable "texte" est binaire, elle indique la présence ou l'absence du texte dans une page web. Une première classification est effectuée sur les pages du site pour différencier les pages présentant un contenu textuel des pages présentant seulement des liens. Ainsi, la première hypothèse utilisée est que les pages ne

contenant pas du texte sont des pages de navigation. Par contre, les pages présentant un contenu textuel peuvent servir de pages de navigation ou de pages de contenu. Ainsi, seules les variables continues sont utilisées dans la classification par le K-means.
Le problème de détermination du nombre de classes ne se présente pas dans ce cas. En effet, on sait à priori que le nombre de classes est autour de 3 : les pages de navigation, les pages de contenu et les pages hybrides.

6.3.1.5 Comparaison des deux méthodes de typage

Pour sélectionner la méthode la plus adaptée au typage des pages, nous comparons la partition obtenue par chaque méthode aux résultats du typage par un expert. La comparaison des partitions repose sur la comparaison des classes qui forment les partitions. La comparaison de classes est effectuée sur la base de leur *extension*, c'est à dire, sur l'ensemble d'individus affectés aux classes. Cette stratégie s'oppose aux techniques de comparaison de partitions basées sur l'*intension*, c'est à dire, sur l'ensemble de représentants des classes qui peuvent être réels ou fictifs.
Dans notre approche, nous adoptons les critères de validation externes basés sur l'extension, à savoir l'indice de Rand et la F-mesure afin de comparer les résultats du typage par classification et par l'algorithme HITS. Dans le paragraphe suivant, nous présentons les résultats d'application de ces deux méthodes sur un site Web de tourisme.

Résultats du typage des pages par classification

L'application de l'algorithme de Kmeans (algorithme de Forgy (1965)[For65]) aux pages du site décrites par les variables : nombre d'inlinks, nombre total d'outlinks (liens internes + externes) et taille des fichiers permet de détecter les trois classes dont les centres sont présentés dans le tableau 6.2.

TABLE 6.2 – Centres des classes de pages.

	Inlinks	Outlinks	Size
Pages de contenu	2.0	1.42	22.85
Pages hybrides	2.47	3.75	5.3
Pages de navigation	3.55	8.88	4.37

Les caractéristiques des classes obtenues permettent d'attribuer une étiquette à chacune d'entre elles (Tab.6.3). En effet, la classe C1 (représentée en rouge sur fig. 6.7) comporte environ 7 % de pages caractérisées par la présence de liens entrants et d'un nombre très élevé de liens sortants utilisés pour passer d'une page à

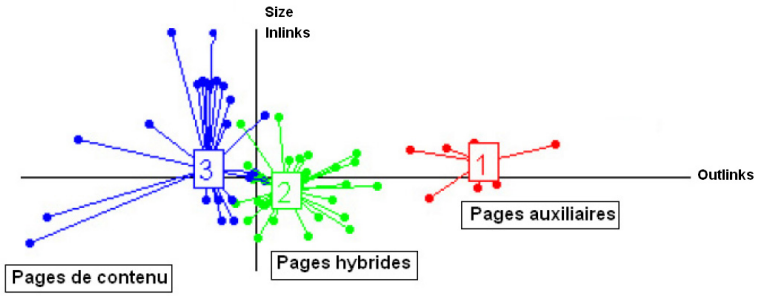


FIGURE 6.7 – Projection des classes de pages sur le premier plan factoriel d'une ACP.

TABLE 6.3 – Caractérisation des classes des pages.

	Texte	Inlinks	Outlinks	Size
Pages de contenu	+	+	-	++
Pages hybrides	+	+	+	-
Pages de navigation	+/-	+	++	-

une autre. La taille de ces fichiers n'est pas importante et le contenu textuel n'est pas toujours présent. Ces pages sont destinés principalement pour passer d'une page à une autre. Elles correspondent alors à des pages de navigation. Par contre, la classe C3 (représentée en bleu sur fig. 6.7) comporte les pages caractérisées par un nombre élevé de liens entrants, une taille importante et un nombre faible de liens sortants. Plusieurs pages pointent vers ces pages mais elles pointent vers un nombre faible de pages. Ces pages qui présentent toujours du contenu textuel sont des pages de contenu. La classe C2 est composée de pages ayant les caractéristiques de pages de contenu et de pages de navigation. Ce sont les pages "hybrides".

Comparaison des résultats de typage par les deux méthodes

L'application de l'algorithme des Hubs et des Authorities permet de déterminer les vecteurs des scores "Authorities" et "Hubs". Nous fixons le seuil S à 0.1 pour distinguer les pages "Authorities" des pages "Hubs". Ainsi trois classes de pages sont identifiées, à savoir la classe des pages de contenu, ou Authorities (26,6%), la classe des pages de navigation, ou hubs (5,4 %) et la classe des pages hybrides (68 %).

Nous proposons de comparer les résultats de typage des pages par cette méthode avec les résultats du typage par classification et du typage à l'aide d'un expert. Pour ce faire nous calculons la F-mesure et l'indice de Rand corrigé des différentes partitions obtenues par les trois méthodes (tab.6.4).

TABLE 6.4 – Comparaison des partitions obtenues par les trois méthodes en utilisant l'indice de Rand et la F-mesure.

	Rand corrigé	F-mesure
Expert vs Kmeans	0.032	0.58
Expert vs Hubs and Authorities	0.28	0.71
Kmeans vs Hubs and Authorities	0.1	0.6

L'indice corrigé de Rand fournit une mesure globale basée sur tout l'ensemble de classes dans les partitions comparées alors que la F-mesure effectue l'analyse classe par classe en cherchant la meilleure représentation d'une classe dans la première partition par une classe correspondante dans la deuxième partition.
D'après le tableau 6.4 de comparaison, nous remarquons que le typage des pages par la méthodes des hubs et des authorities donne des meilleurs résultats. Nous utilisons alors cette méthode pour distinguer les pages de navigation ou hubs des pages de contenu. Ces pages seront exclues de l'analyse textuelle puisqu'elles ne servent qu'à facilier la navigation sur le site Web.

6.3.1.6 Limites de l'approche

Cette étape de l'approche WCUM n'est applicable que dans le cas des sites Web statiques. En effet, dans le cas des sites web dynamiques, les pages sont générées dynamiquement en fonction des requêtes des visiteurs (saisie de texte dans des zones spéciales, listes déroulantes, cases à côcher, boutons d'option, etc.) et en sollicitant les bases des données du site. Il n'est pas donc souvent possible de distinguer les pages de contenu des pages de navigation. Dans ce cas, toutes les pages du site Web sont utilisées dans les autres phases de l'analyse textuelle.

6.3.2 Prétraitement des textes

Une fois les pages de contenu distinguées des pages de navigation, ces dernières sont exclues de l'analyse textuelle puisqu'elles ne sont pas conçues pour véhiculer de l'information.
L'objectif du prétraitement est de représenter chaque page du site par un vecteur de descripteurs qui donne une idée sur son contenu. Comme notre analyse est focalisée sur le contenu textuel du Web, nous considérons que la représentation des pages Web est similaire à celle d'un texte. Le prétraitement est réalisé en deux étapes. La première étape est celle de représentation des pages. La seconde est celle du choix des descripteurs.
Dans notre cas, le prétraitement nécessite tout d'abord la conversion des pages Web en fichiers textes, et le remplacement des images qu'ils contiennent par leurs

légendes. Ces textes sont par la suite traités par l'algorithme TreeTagger, développé à l'Institut de Linguistique Computationelle de l'Université de Stuttgart [Sch94].
L'étiquetage et la lemmatisation à l'aide de TreeTagger [1] permettent de remplacer les verbes par leur forme infinitive, les noms par leur forme au singulier et certaines formes des verbes tels que les participes présents et les participes passés par leurs racines. Afin de réduire la dimension de l'espace vectoriel des vecteurs représentant les textes, il s'avère nécessaire de supprimer :

- les formes de ponctuation,
- les mots vides tels que les prépositions, les déterminants, les numéros, les conjonctions, les pronoms et les abréviations,
- les mots inutiles à la classification tels que les adverbes et les adjectifs,
- les mots de type non reconnu par TreeTagger sont examinés manuellement afin de ne garder que les noms et les verbes. D'autre part, les termes auxquels TreeTagger attribue l'étiquette "Nom" sont examinés afin de supprimer les noms propres que TreeTagger n'arrive pas à identifier. Ainsi, seuls les noms et les verbes sont conservés dans la base des descripteurs.
- les mots très fréquents : nous avons adopté la méthode proposée par [Str00]. En effet, le rapport $R(t, P) = TF(t, P)/CF(t)$, tel que $TF(t, P)$ est l'occurrence du terme t dans la page P et $CF(t)$ est l'occurrence du terme t dans l'ensemble des pages, permet de classer les termes par ordre décroissant. Plus le terme t est fréquent, plus le ratio est faible et, inversement, plus un terme est rare, plus le ratio est élevé. Dans le cas limite où un terme n'apparaît qu'une seule fois dans l'ensemble des pages, ce ratio vaut 1 et le mot est classé en tête de liste.
- les mots très peu fréquents : ce sont les mots dont le nombre de documents dans lesquels ils apparaissent est inférieur à un certain seuil. Dans notre cas, nous supprimons les mots qui apparaissent dans une seule page du site Web et ceux dont le nombre d'occurrences dans la base est égal à 1 i.e. les mots qui apparaissent une seule fois dans toute la base.

TABLE 6.5 – Tableau de contingence croisant les descripteurs aux pages

	$P1$	...	P_i	...	P_q
D_1	0	...	1	...	5
...	...	...	...	...	...
D_i	1	...	0	...	0
...	...	...	...	...	...
D_n	1	...	0	...	6

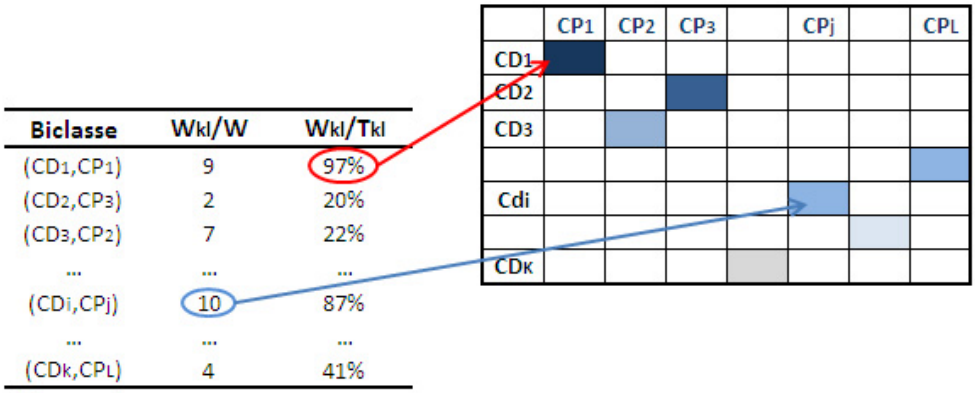
1. Les publications relatives à cet algorithme ainsi que les codes source sont disponibles sur le site : http ://www.ims.uni-stuttgart.de/projekte/corplex/DecisionTreeTagger.html

6.3.3 Block clustering

Au niveau du block clustering, nous appliquons l'algorithme Croki2 à la matrice croisant les descripteurs aux pages. Outre le nombre de tirages aléatoires des partitions initiales qu'on fixera à 100, l'algorithme Croki2 nécessite de fixer le nombre de classes sur les lignes et les colonnes. D'après les résultats expérimentaux présentés dans le chapitre précédent, la différentielle et l'indice de BH sont les meilleurs critères d'estimation du bon nombre de classes parmi les critères testés. Les biclasses $B_{kl} = (I_l, J_l)$ retenues sont les biclasses les plus pertinentes et les plus homogènes selon les deux critères $PB = W_{kl}/W$ et $HB = W_{kl}/T_{kl}$ avec

$$W_{kl} = f_{k.}f_{.l}(\frac{f_{kl}}{f_{k.}f_{.l} - 1})^2, W = \sum_{k,l} W_{kl}$$

$$T_{kl} = \sum_{i \in P_k} \sum_{j \in Q_l} f_{i.}f_{.j}(\frac{f_{ij}}{f_{i.}f_{.j} - 1})^2$$



Biclasse	Wkl/W	Wkl/Tkl
(CD1,CP1)	9	97%
(CD2,CP3)	2	20%
(CD3,CP2)	7	22%
...	...	...
(CDi,CPj)	10	87%
...	...	...
(CDK,CPL)	4	41%

FIGURE 6.8 – Biclasses obtenues suite à l'application de Croki2

Ainsi, à chaque classe de pages au moins une classe de descripteurs est associée. L'avantage de l'algorithme Croki2 est que chaque ligne (resp. chaque colonne) est affectée à une classe-ligne (resp. classe-colonne). Cette exhaustivité est essentielle dans le cas de l'analyse textuelle puisque chaque page devrait appartenir à un thème particulier. De même à chaque classe-ligne (resp.classe-colonne), au moins une classe-colonne (resp. classe-ligne) est associée selon les mesures de pertinence et d'homogénéité.

Soient les partitions P et Q définies respectivement sur l'ensemble de descripteurs et l'ensemble de pages, f une fonction définie de P dans Q.

$$f : P \rightarrow Q$$

$$\forall k \in [1, K], \exists l \in [1, L] / f(CD_k) = CP_l$$

avec $CD_k \in P$ et $CP_l \in Q$

La fonction f est surjective, c'est à dire, à chaque classe de descripteurs est associé au moins une classe de pages en se basant sur les critères de pertinence et d'homogénéité. Soient $l_1, l_2 \in [1, L]$ tels que $f(CD_k) = CP_{l_1}$ et $f(CD_k) = CP_{l_2}$. Dans ce cas, deux classes différentes de pages sont associées à la même classe de descripteurs. En examinant les deux critères HB et PB d'homogénéité et de pertinence (Chapitre 4, section 4.3.3.2), une des biclasses serait plus pertinente ou plus homogène. Il est possible, selon les besoins de l'analyse textuelle du site, de fusionner les deux biclasses puisque les deux classes de pages traitent du même sujet ou thème.
Soient $k_1, k_2 \in [1, K]$ tels que $f(CD_{k_1}) = CP_l$ et $f(CD_{k_2}) = CP_l$. Dans ce cas, la même classe de pages est associée à deux classes différentes de descripteurs donc à deux thèmes différents. Le critère de pertinence PB permet de détecter le thème principal et le thème secondaire des pages composant la classe CP_l.

6.3.4 Analyse Sémantique Latente

Dans l'analyse sémantique latente, les mots et les documents originaux sont positionnés dans un nouvel espace sémantique, ce qui permet de mesurer leur proximité par le cosinus entre les vecteurs qui les représentent. Nous proposons alors d'utiliser l'ASL pour valider la classification croisée du tableau croisant les descripteurs aux pages Web. Pour ce faire, nous appliquons l'ASL sur chaque biclasse de pages et descripteurs afin de mesurer la similarité entre les éléments (pages ou descripteurs) qui la compose. Le cosinus entre les descripteurs D_i et D_j de chaque biclasse est calculé par la formule suivante :

$$cos(D_i, D_j) = \frac{D_i.D_j}{\|D_i\| \, \|D_j\|}$$

La même stratégie est suivie pour mesurer la similarité entre les pages P_i et P_j en utilisant le cosinus.

$$cos(P_i, P_j) = \frac{P_i.P_j}{\|P_i\| \, \|P_j\|}$$

6.3.5 Evaluation de la structure du site à travers la catégorisation des pages

L'URL de chaque page est organisé de façon hiérarchique sous forme de rubriques et de sous-rubriques représentant, selon le point de vue du concepteur, le

contenu de la page. En examinant la structure arborescente des pages, la décomposition de l'URL de chaque page en rubriques permet de classer les pages selon la rubrique du niveau i, le premier niveau ou le deuxième niveau par exemple. Afin d'évaluer la cohérence entre le contenu et la structure, nous proposons de comparer les partitions résultantes de la classification en rubriques et celles résultantes de la catégorisation des pages. Soit l'exemple de la classe des pages et la classe des descripteurs de la figure 6.9.

Classe des pages	Rubrique 1	Rubrique 2	Classe des descripteurs			
info/flash_html	info		visiter	calendrier	départ	âge
visite/calendrier_2006_html	visite		jour	circuit	entrée	individuel
visite/groupe/visite1_html	visite	groupe	guide	programme	fermer	inscription
visite/groupe/visite31_html	visite	groupe	audioguide	tarif	heure	reservation
...			semaine	réduction	langue	modalité

FIGURE 6.9 – Exemple de classes de pages et de descripteurs

L'ensemble de ces pages traitent du même sujet "tarifs et réservations des visites". L'URL de la majorité de ces pages comporte "visite" comme première rubrique sauf la page "info/flash.html" dont la première rubrique est "info". De plus aucun hyperlien n'existe entre cette page et les autres pages bien qu'elles ont toutes la même thématique. Il serait donc mieux de lier cette page aux autres de la même biclasse pour qu'elle soit plus accessible et que son contenu soit à la portée des visiteurs du site Web.

6.4 Analyse de l'usage d'un site

Comme le format des fichiers log Web est impropre à une analyse directe par les diverses techniques de fouille des données, leur structuration est nécessaire avant toute analyse. D'autre part, plusieurs requêtes telles que les requêtes invalides, celles effectuées par des robots ou celles dont la méthode est différente de "GET", ne reflètent pas le comportement de l'utilisateur. Ces requêtes doivent être identifiées et exclues de l'analyse.
La phase de transformation consiste à l'identification des utilisateurs et des sessions et la construction des visites de chaque utilisateur. Le résultat du prétraitement effectué à l'aide de l'outil proposé par [ASL08] est une base de données relationnelle dont le schéma en étoile est présenté dans la figure 6.10.

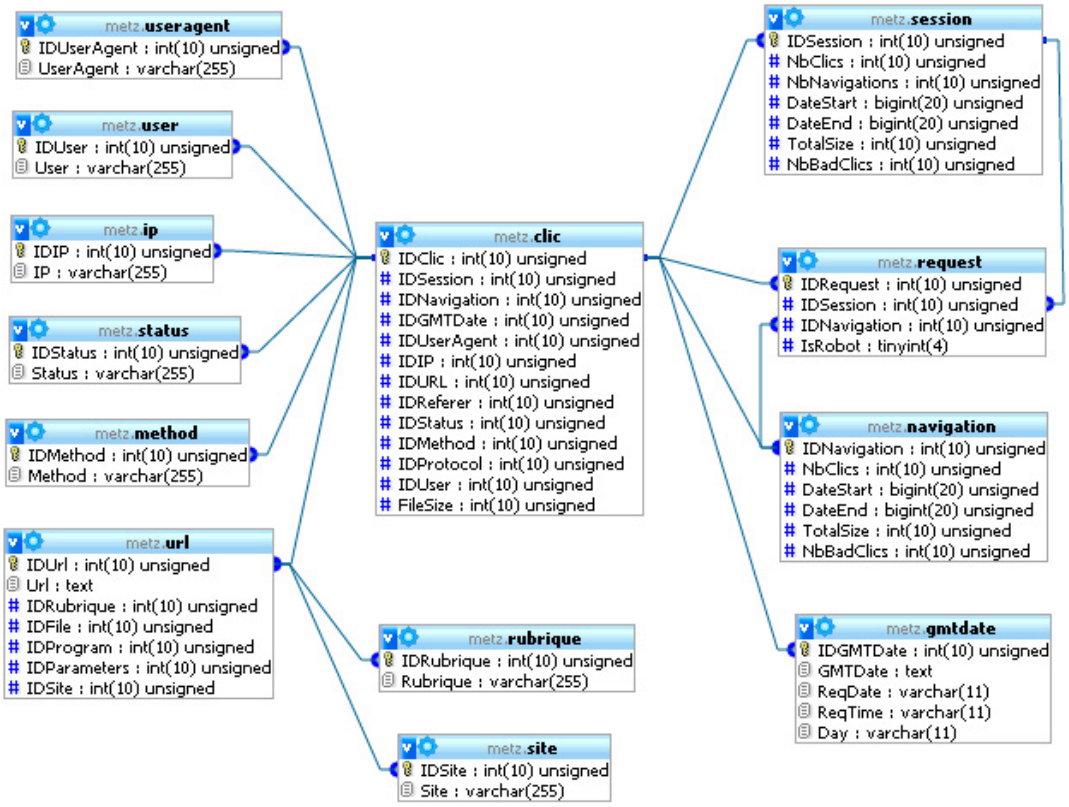


FIGURE 6.10 – Schéma en étoile de l'entrepôt des données généré par l'outil de prétraitement de [ASL08]

6.4.1 Identification des sessions et des utilisateurs

Une session est composée de l'ensemble de pages visitées par le même utilisateur durant la période d'analyse. Plusieurs moyens d'identification des utilisateurs ont été proposés dans la littérature (login et mot de passe, cookie, IP). Cependant, tous ces moyens présentent des défaillances à cause des systèmes de cache, des firewalls et des serveurs proxy. Un moyen qui s'avère plus efficace consiste à utiliser le couple (IP, User-Agent) comme un identifiant des utilisateurs. Dans ce cas, deux requêtes provenant de la même adresse IP mais de deux User-agents différents appartiennent à deux sessions différentes puisqu'elles sont effectuées par deux utilisateurs différents. Toutefois, nous ne pouvons nier la limite inhérente à cette méthode. En effet, une confusion entre deux utilisateurs différents utilisant la même adresse IP et le même User-Agent est toujours possible surtout en cas d'utilisation d'un serveur Proxy ou d'un firewall. Chaque session est caractérisée par le nombre de requêtes (ou clics), le nombre de navigations effectuées par

l'utilisateur durant cette session, la date du début et de la fin de la session et le nombre de requêtes invalides.

6.4.2 Identification des navigations

Une navigation (ou visite) est composée d'une série de requêtes séquentiellement ordonnées, effectuées pendant la même session et ne présentant pas de rupture de séquence de plus de 30 minutes (d'après les critères empiriques de Kimball [KM00]). La durée de consultation d'une page est le temps séparant deux requêtes http diminué du temps nécessaire au chargement de la page. Chaque navigation est caractérisée par le nombre de requêtes (ou clics) effectuées pendant cette visite, le nombre de requêtes invalides et les dates de début et de fin de la visite.

6.4.3 Structuration des données du contenu et de l'usage

Outre les données de l'usage obtenues par l'outil de prétraitement proposé par [ASL08], nous intégrons les données sur le contenu. Pour ce faire, nous créons les nouvelles tables *Descripteur* pour stocker les descripteurs de pages et *thème* pour stocker les thèmes de pages. Ces deux tables sont remplies à partir des résultats du block clustering. Les données sur l'usage sont extraites de la base des données de l'usage par des requêtes SQL. Le nouveau schéma de la base relationnelle est présenté dans la figure 6.11.

A partir de ces données, différentes analyses peuvent être effectuées. Dans notre approche, notre objectif est d'analyser le comportement des utilisateurs vis à vis des thèmes détectés précédemment par la classification croisée des pages et des descripteurs.
La nouvelle matrice des données croise les navigations aux thèmes des pages.

TABLE 6.6 – Matrice des navigations et des thèmes

	Thème 1	Thème 2	...	Thème L
Navigation 1	20	0	...	2
Navigation 2	0	11	...	0
...	...	...	...	...
Navigation n	0	43	...	10

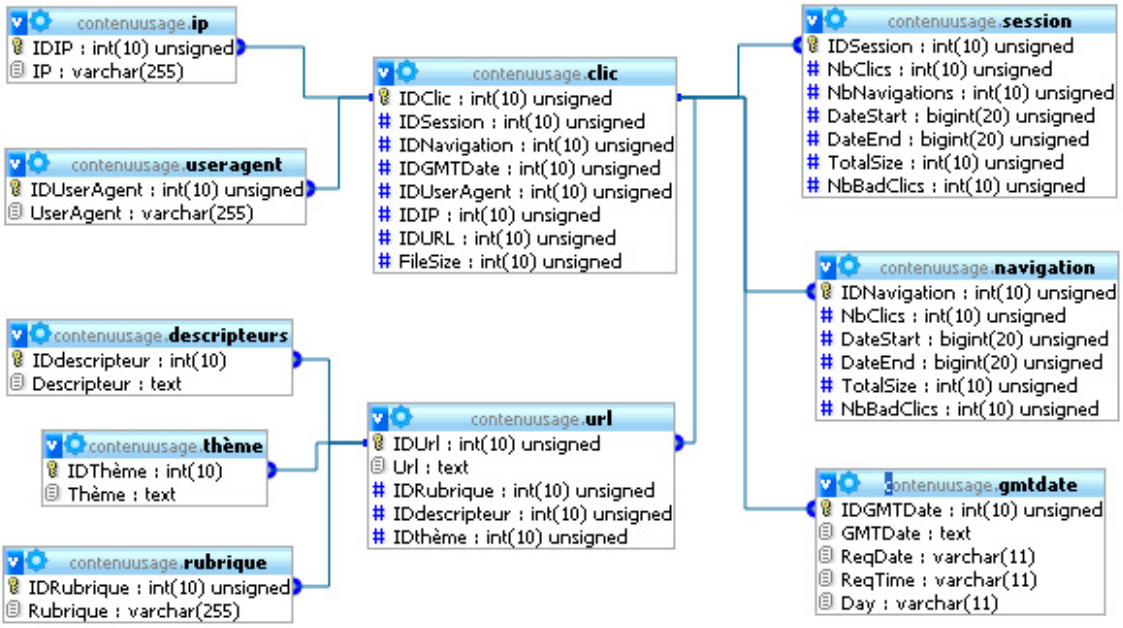


FIGURE 6.11 – Schéma de la nouvelle base des données de l'usage et du contenu

6.4.4 Classification floue

Dans la classification floue, les classes ne forment plus une partition nette de l'espace, mais une partition floue où le recouvrement est autorisé. De cette façon, un même individu peut appartenir à plusieurs classes. Un bon nombre de méthodes floues de reconnaissances des formes sont en fait des extensions de méthodes classiques. Le résultat délivré par un algorithme de regroupement flou est alors une matrice $n \times c$ des degrés d'appartenance, où n est le nombre d'individus et c le nombre de classes obtenues. L'algorithme le plus connu dans la classification floue est le Fuzzy C-Means.

6.4.4.1 Fuzzy C-Means (FCM)

L'algorithme des c -moyennes floues (fuzzy c-means : FCM) est une extension directe de l'algorithme classique des c -moyennes (c -means), où l'on a introduit la notion d'ensemble flou dans la définition des classes. Cet algorithme a été développé essentiellement par [Bez81], à partir des idées originales de Ruspini (clustering flou) [Rus69] et de Dunn (ISODATA flou) [Dun73].

Le principe de base est de former à partir des individus non étiquetés c groupes qui soient les plus homogènes et naturels possible. "Homogène" et "naturel" signifient que les groupes obtenus doivent contenir des individus les plus semblables

possible, tandis que des individus de groupes différents doivent être les plus dissemblables possible. Comme les autres algorithmes de classification non supervisée, il utilise un critère de minimisation des distances intra-classe et de maximisation des distances inter-classe, mais en donnant un certain degré d'appartenance à chaque classe pour chaque individu. Cet algorithme nécessite la connaissance préalable du nombre de clusters et génère les classes par un processus itératif en minimisant le critère suivant :

$$J_m = \sum_{j=1}^{c} \sum_{i=1}^{n} (\mu_{ij})^m d_j^2(x_i, \vartheta_j)$$

où

- ϑ_j désigne le centroide du cluster j, et x_i est l'individu numéro i.
- $d_j(x_i, \vartheta_j)$ est la distance entre l'échantillon x_i et le centroide du cluster j, définie par

  $$d_j^2(x_i, \vartheta_j) = (x_i - \vartheta_j)^T A_j (x_i - \vartheta_j)$$

  Si A_j est la matrice identité (cas de la version habituelle des FCM), le cluster j sera sphérique, sinon il sera ellipsoïdal [GK79].
- μ_{ij} est le degré d'appartenance de l'individu i à la classe j. Les μ_{ij} forment une matrice $c \times L$, dite matrice d'appartenance M.
- $m > 1$ est un index de *fuzziness*, qui règle le degré de *fuzziness* des clusters. Si m tend vers 1, la matrice M comportera uniquement des 0 et des 1, donc il y'a tendance vers des clusters "nets". Par contre, plus m est grand, plus les clusters sont "mous", c'est-à-dire que leur fonction d'appartenance est très étendue.

Etapes de l'algorithme

Les principales étapes de l'algorithme Fuzzy C-means sont :

- Etape 0 : fixation arbitraire du nombre de clusters c et d'une matrice d'appartenance $M = [\mu_{ij}]$. Cette matrice représente une partition floue des données, et doit vérifier la condition de normalisation

  $$\sum_{j=1}^{c} \mu_{ij} = 1, \forall i$$

- Etape 1 : calcul des centroïdes des classes.

  $$\vartheta_j = \frac{\sum_{i=1}^{n} (\mu_{ij})^m x_i}{\sum_{i=1}^{n} \mu_{ij}^m}$$

– Etape 2 : réajustement de la matrice d'appartenance suivant la position des centroïdes.

$$\mu_{ij} = \frac{(\frac{1}{d_j^2(x_i,\vartheta_j)})^{1/(m-1)}}{\sum_{j=1}^{c}(\frac{1}{d_j^2(x_i,\vartheta_j)})^{1/(m-1)}}$$

– Etape 3 : calcul du critère de minimisation et retour à l'étape 1 s'il y a non convergence de critère.

L'algorithme converge toujours vers un minimum local strict, qui dépend de la matrice M, initiale. Ainsi, il permet d'obtenir une partition floue des navigations en donnant à chaque navigation un degré d'appartenance (compris entre 0 et 1) à une classe donnée. Le cluster auquel est associé une navigation est celui dont le degré d'appartenance est le plus élevé.

Critères d'identification du nombre de classes

L'identification du nombre de clusters nécessite d'effectuer plusieurs essais en modifiant à chaque fois le nombre de clusters. Différents critères de validité sont proposés dans la littérature pour choisir la meilleure partition et par suite le bon nombre de classes. Les deux premiers critères que nous présentons ici, à savoir le coefficient de partition floue et l'entropie, sont proposés par Bezdek pour mesurer le degré de recouvrement des clusters. Xie et Beni [XB91] ont proposé un autre critère, très simple à calculer, et qui mesure la compacité globale et la séparation des clusters.

– Le coefficient de partition floue :

$$F(M) = \frac{1}{n}\sum_{j=1}^{c}\sum_{i=1}^{n}\mu_{ij}^2$$

– L'entropie :

$$E(M) = \frac{1}{n}\sum_{j=1}^{c}\sum_{i=1}^{n}\mu_{ij}log(\mu_{ij})$$

– Indice Xie-Beni :

$$XB = \frac{\pi}{s}$$

où

$$\pi = \frac{\sum_{j=1}^{c}\sum_{i=1}^{n}\mu_{ij}^2 d_j^2(x_i - c_j)}{n}$$

et

$$s = (d_{min})^2$$

d_{min} est la distance minimale entre les centres de classes, $d_{min} = min_{ij}d(c_i, c_j)$. Une valeur faible de π indique que les classes sont plus compactes et une

valeur élevée de s indique que les classes sont bien séparées. Ainsi, plus la valeur de l'indice est faible plus le partitionnement est bon.

6.4.5 Evaluation de la structure du site à travers l'analyse de l'usage

L'objet de cette section est de se focaliser sur la preception de l'organisation et du contenu du site par ses utilisateurs en faisant ressortir des rapprochements entre documents visités par les internautes. Les mêmes résultats pourraient être obtenues également par d'autres méthodes tels que les règles d'association ou la recherche des motifs fréquents. Cependant, nous estimons que la classification serait plus adaptée à notre problème car à partir des résultats de la classification il est possible d'appliquer d'autres types d'analyse et d'extraire d'autres connaissances sur la structure et l'usage. Soit la classe C composée de navigations ayant des motifs de visite relatifs à deux thèmes différents "Thème1" et "Thème2". Le fait que ces deux thèmes soient visités ensemble laisse deviner la présence d'hyperliens facilitant le passage d'une classe de pages (représentée par un thème) à une autre. Cependant, si en examinant la structure du site, on s'aperçoit que les pages relatives à chacun de ces thèmes sont reliées entre elles par des hyperliens mais qu'aucun hyperlien ne permet le passage d'un thème à l'autre autre que la page d'accueil, on peut déduire que selon le concepteur, ces thèmes sont indépendants alors que de point de vue visiteurs du site ces thèmes sont dépendants. Par conséquent, il serait intéressant d'ajouter un hyperlien pour faciliter la navigation entre ces thèmes.

6.5 Application à un site Web de tourisme

Nous proposons d'appliquer l'approche WCUM à un site Web de tourisme. Pour ce faire, nous procédons à l'aspiration du site et l'acquisition des fichiers logs en s'assurant que le site n'a pas été modifié pendant la période d'analyse.

6.5.1 Analyse textuelle

L'analyse textuelle du site Web débute par un typage de pages à l'aide des deux méthodes citées précédemment (méthode des hubs et des authorities et méthode de classification par un ensemble de variables) afin de distinguer les pages de navigation des pages de contenu. Seules les pages de contenu et les pages hybrides sont utilisées dans la suite des traitements.

6.5.1.1 Prétraitement et catégorisation des pages

La prétraitement des pages de contenu et des pages hybrides par Treetagger et la sélection des descripteurs de pages aboutissent à la construction d'une matrice croisant 418 descripteurs à 125 pages. Chaque cellule dans la matrice correspond au nombre d'occurrences du descripteur dans la page.

Détermination du nombre de classes

Dans le cas réel, il est beaucoup plus difficile de distinguer le couple de classes à partir duquel le χ^2 croit plus lentement ou devient constant (fig.6.12). D'après les expérimentations sur les données artificielles, seuls l'indice BH et la différentielle ont réussi à identifier le bon nombre de classes dans les données dans la majorité des cas. Pour cette raison, en les appliquant aux données réelles, on espère estimer, même grossièrement, le nombre de classes sur les lignes et les colonnes.

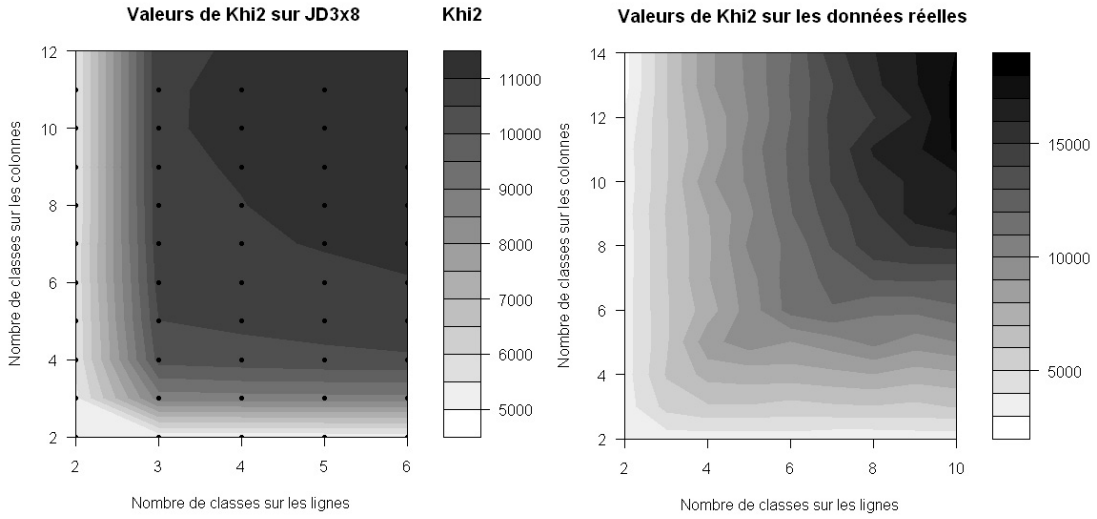


FIGURE 6.12 – Valeurs de χ^2 sur JD3x8 et sur les données réelles

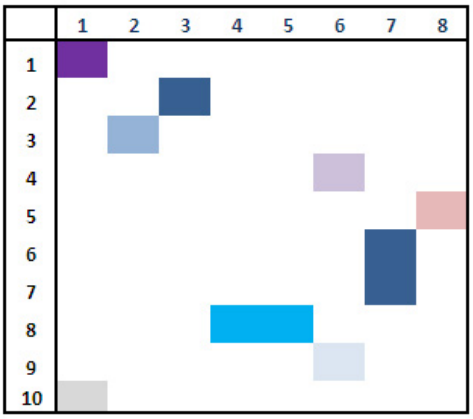
Selon l'indice BH, la meilleure partition sur les lignes est obtenue pour le couple (10,8) alors que la meilleure partition sur les colonnes est obtenue pour le couple (10,14). Comme l'indice-ligne et l'indice-colonne proposent deux solutions différentes (en particulier le nombre de classes sur les colonnes proposé par les deux indices est différent), nous avons eu recourt au calcul d'un indice pondéré tel que celui décrit dans la section 5.3.1. Ainsi, pour $\alpha = 0.5$, le meilleur couple de classes selon l'indice BH est (10,8).

TABLE 6.7 – Meilleurs couples de classes selon l'indice BH appliqué aux partitions-lignes et partitions-colonnes des données réelles.

(k,l)	BH-Colonne	Rang	(k,l)	BH-Ligne	Rang
(10,14)	0.7361	1	(10,8)	0.4162	1
(8,10)	0.7204	2	(9,4)	0.3999	2
(8,7)	0.7167	3	(10,12)	0.3701	3
(9,10)	0.7101	4	(8,5)	0.3650	4
(10,8)	0.7031	5	(4,7)	0.3551	5

Catégorisation des pages

L'application de l'algorithme CROKI2 la matrice croisant les descripteurs aux pages aboutit à un ensemble de biclasses. La sélection des meilleures nécessite le recours aux trois critères suivants : la taille des classes composant la biclasse (nombre de pages et nombre de descripteurs), l'homogénéité (part d'inertie conservée par la biclasse par rapport à l'inertie initiale des points de la biclasse) et la pertinence de la biclasse (part d'inertie conservée par la biclasse par rapport à l'inertie totale).
Dans l'exemple de la figure (6.13), les biclasses (2,3) et (1,1) sont les meilleures biclasses en terme de pertinence alors que les biclasses (1,1) et (7,7) sont les plus homogènes. En terme de pertinence et d'homogénéité, la biclasse (1,1) est la plus importante.



Biclasse	Wkl/W	Wkl/Tkl
(1,1)	9%	97%
(2,3)	1%	26%
(3,2)	7%	22%
(4,6)	6%	18%
(5,8)	6%	30%
(6,7)	3%	65%
(7,7)	2%	87%
(8,4)	2%	26%
(8,5)	4%	41%
(9,6)	2%	10%
(10,1)	1%	10%

FIGURE 6.13 – Sélection des meilleurs biclasses

L'examen de ces biclasses a pour objectif d'attribuer un thème à chaque groupe

de pages. Par exemple, la biclasse (2,3) composée de la classe 2 de descripteurs et la classe 3 des pages a pour thème "spécialités de cuisine" sachant que presque tous les descripteurs sont en relation avec l'alimentation (amande, crème, eau de vie, flamber, fruit, glacer, mirabelle, recette, purée, etc...) (tab. 6.8).

Classes de descripteurs	Descripteurs	Thèmes
Classe 1	Arme, Art, Artiste, balade, Château, Découverte, Église, Exposition, Galerie, Guerre, Habit, Histoire, Illustrer, Maréchal, Monument, Moyen-Age, Palais, Pasteur, Peintre, Peinture, Promeneur, Renaissance, République, Saint, Siècle, Spectacle, Trésor	Histoire et Monuments
Classe 2	Amande, Crème, Eau, Flamber, Fruit, Gastronomie, Glacer, Lait, Mirabelle, Oeuf, Purée, Recette, Sucre,Hospitalité	Recettes de cuisine
Classe 3	Cathédrale, Bibliothèques-médiathèques Boulevard, Capitale, Direction, Édifices, Gare, Guide, Hôpital, Information	Autres Adresses
Classe 5	Activité, Fête, Bal, Football, lieu, Manifestation, Occasion, Réunion	Activités et Manifestations
Classe 8	Bergerie, Brasserie, Centre, Distance, Fax, Hôtel, Magasin, Nord, Port, Zone, Restaurant, Sud, Technopôle, Tél, Village	Hôtels et Restaurants

TABLE 6.8 – Exemples de thèmes

Les biclasses (8,4) et (8,5) ont le même thème "hébergement" puisqu'il s'agit de la même classe en ligne. Par conséquent, il est possible de regrouper les pages de la classe 4 et de la classe 5 dans une même classe (4+5). La nouvelle biclasse obtenue après fusion des biclasses (8,4) et (8,5) a pour thème "informations sur les hôtels ou hébergement". Par contre, les biclasses (6,7) et (7,7) présentent en commun la même classe en colonne. Ainsi, les pages de la classe 7 auront deux thèmes différents "horaires et tarifs des lieux à visiter" et "réservation".

6.5.1.2 Analyse Sémantique Latente

L'exemple de la figure 6.14 présente les résultats de l'application de l'ASL à deux biclasses différentes. La valeur du cosinus est élevée lorsque les deux descripteurs sont affectés à la même biclasse. Par contre, elle est faible entre deux descripteurs appartenant à deux biclasses différentes. Par exemple, le cosinus entre deux descripteurs quelconques de la classe 2 composé des descripteurs : $Corr\{Sucre, Oeuf\} = 1.0$. Alors que le cosinus entre un descripteur de la classe 1 et un descripteur de la classe 2 est au plus 0.5, par exemple

$$Corr\{sucre, tourisme\} = 0.5.$$

Au niveau des pages, la corrélation est élevée entre deux pages appartenant à la même biclasse, par exemple

$$Corr\{specialites/mousse.html, specialites/tartes.html\} = 1.0$$

alors qu'elle est faible entre deux pages appartenant à deux biclasses différentes, par exemple

$$Corr\{specialites/mousse.html, specialites/antiquite.html\} = 0.39$$

	Sucre	Œuf	Mirabelle	Eau	Art	Antiquité	Tourisme
Sucre	1.00	1.00	0.97	0.91	0.40	0.44	0.19
Œuf	0.97	1.00	0.97	0.91	0.50	0.44	0.20
Mirabelle	0.97	0.97	1.00	0.98	0.30	0.50	0.43
Eau	0.91	0.91	0.98	1.00	0.50	0.30	0.50
Art	0.40	0.50	0.30	0.50	1.00	0.99	0.92
Antiquité	0.44	0.44	0.50	0.30	0.99	1.00	0.97
Tourisme	0.19	0.20	0.43	0.50	0.92	0.97	1.00

FIGURE 6.14 – Application de l'ASL aux biclasses : Corrélation entre les descripteurs

	specialites/mira_html	specialites/mousse_html	specialites/tartes_html	specialites/antiquite_html
specialites/mira_html	1	1	1	0.39
specialites/mousse_html	1	1	1	0.39
specialites/tartes_html	1	1	1	0.39
specialites/antiquite_html	0.39	0.39	0.39	1

FIGURE 6.15 – Application de l'ASL aux biclasses : Corrélation entre les pages

6.5.2 Analyse de l'usage

L'analyse de l'usage débute par la collecte des données enregistrées dans les logs. Ces données doivent correspondre aux visites des utilisateurs aux pages utilisées dans l'analyse textuelle. En d'autres termes, si le contenu du site a subi des modifications les fichiers logs doivent comporter les traces des utilisateurs sur la nouvelle version du site. Au niveau du prétraitement des fichiers logs, nous adoptons la méthodologie proposée par [ASL08] [AJL08] et inspirée des travaux de [CMS99][TT04]. Cette méthodologie consiste à appliquer les étapes de *prétraitement classique* (fusion et structuration de données) et de *prétraitement avancé* (transformation et extension de données) (fig. 6.16).

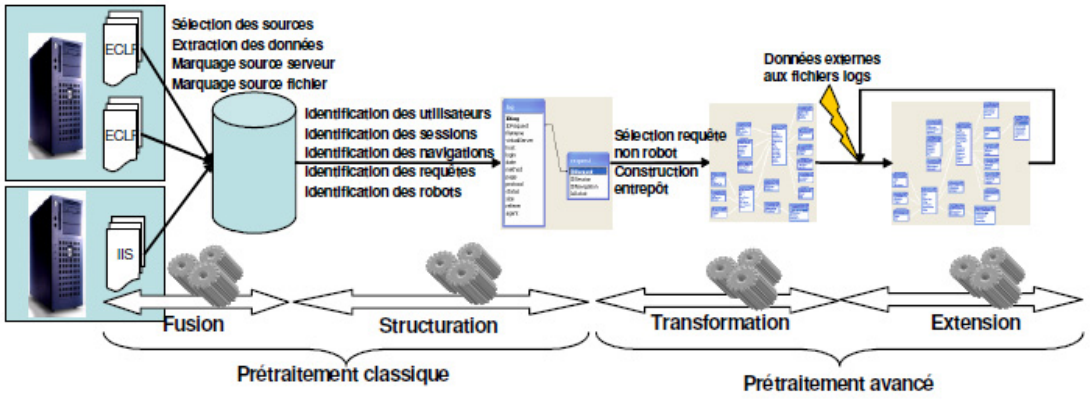


FIGURE 6.16 – Méthodologie de prétraitement de fichiers logs Web [AJL08].

La première étape dans cette méthodologie est celle de *fusion* des données. Les fichiers logs Web issus de différents serveurs Web et formatés selon différents standards sont sélectionnés et fusionnés. L'étape suivante est celle de *structuration*. Elle consiste à identifier les robots, les requêtes provenant de ces robots, les requêtes effectuées par les utilisateurs (humains), les navigations et les sessions qui leurs sont associées. Une navigation est définie comme une suite de requêtes séquentiellement ordonnées, provenant d'un même utilisateur et ne présentant pas de rupture de séquence de plus de 30 minutes.
Dans l'étape de *transformation*, les données sont organisées dans un entrepôt sous forme d'un schéma en étoile. La dernière étape est celle d'*extension* des données. Elle consiste à enrichir les données transformées par des connaissances externes (structure du site, informations additionnelles sur les utilisateurs, etc...). Cette approche incrémentale au niveau temporel et spatial présente l'avantage de prétraiter les fichiers logs dans leur ordre d'arrivée sans avoir à refaire les prétrai-

tements déjà accomplis [2].

Suite à la structuration des requêtes provenant des fichiers logs, une matrice croisant les navigations aux pages est construite. Or d'après l'analyse du contenu, chaque page est affectée à une biclasse donc à au moins un thème. Par suite, il est possible de croiser les navigations aux thèmes. Chaque cellule de la matrice correspond au nombre total de visites effectuées aux pages appartenant au thème j au cours de la navigation i.

6.5.2.1 Classification floue des navigations

Afin de comprendre le comportement des visiteurs sur le site, nous appliquons un algorithme de classification floue, le Fuzzy C-Means. Ce choix de procéder à la classification floue au lieu de la classification "simple" est justifié par le fait qu'un utilisateur peut être intéressé par plusieurs thèmes pendant la même navigation. Par conséquent, une navigation peut être affectée à plusieurs classes à la fois mais avec des degrées différents. A titre d'exemple, pendant la navigation 37, l'utilisateur a visité des pages appartenant à trois thèmes différents.

Identifiant Navigation	Page	Thème
27	/francais/adr/alpha.html	adresses utiles
27	/francais/adr/alpha.html	carte de la ville
27	/francais/geo/carte.html	carte de la ville
27	/francais/hebergement/mtz2et.html	adresses utiles
27	/francais/info/flash.html	adresses utiles
27	/francais/visite/vind.html	programmes de visites
27	/francais/visite/virtu.html	programmes de visites
27	/francais/visite/visite.html	programmes de visites

FIGURE 6.17 – Exemple de navigation

L'application de l'algorithme Fuzzy C-Means à la matrice croisant les navigations aux thèmes permet d'identifier trois classes de navigation. La classe C3 (fig. 6.18) est composée de navigations intéressés par les pages traitant du thème 4 (Informations utiles). Ils sont alors à la recherche des informations sur les horaires d'ouverture et de fermetures de certains établissements, des tarifs et des calendriers. Les navigations de la classe C2 sont par contre effectuées à des pages ayant pour thème "Recettes de cuisine", "histoire et monuments" et "Hôtels et Restaurants" (càd thème1, thème2 et thème6). La classe C1 regroupe les visites dont le motif est la recherche des adresses utiles (Thème 3), des manifestations et

2. L'outil de prétraitement permettant la mise en oeuvre de cette méthodologie est développé en open source et disponible à l'adresse suivante : https ://gforge.inria.fr/projects/webloghousing/

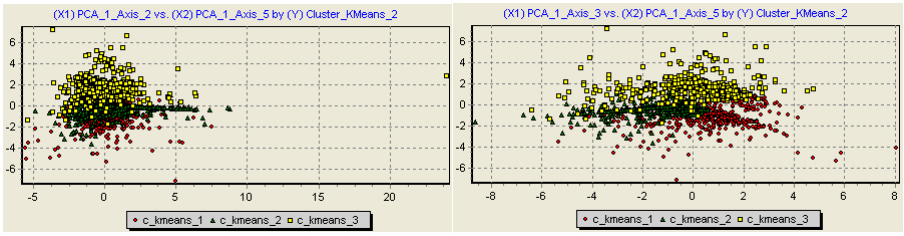


FIGURE 6.18 – Projection des classes d'utilisateurs

des activités culturelles (thème 5) et des informations utiles (thème4). Comme la classe majoritaire est C2 (70% des navigations), on déduit que ce sont les thèmes 1,2 et 6 qui intéressent le plus les visiteurs. Il s'en suit que les pages traitant de ces thèmes devraient être facilement accessibles à partir de la page d'accueil d'une part. D'autre part, il serait intéressant d'examiner de près les liens existant entre les pages traitant de ces thèmes et ajouter des hyperliens pour faciliter le passage d'un thème à l'autre.

6.6 Avantages de l'approche WCUM

L'apport de l'approche WCUM à l'analyse des données du Web réside dans les points suivants :

- Résumé des données textuelles contenues dans les pages web par le block clustering. Ainsi, au lieu de considérer toutes les pages du site Web dans l'analyse de l'usage, seuls les thèmes des pages sont utilisés ce qui facilite l'analyse des sites web volumineux. D'autre part, remplacer les URLs dans les matrices croisant les sessions ou navigations aux pages permet de pallier la difficulté d'appliquer des méthodes de classification à des matrices creuses (*problem of matrices sparsity*).
- Suivi des changements dans le contenu et l'influence de ce changement sur le comportement des utilisateurs. En effet, en impliquant dans l'analyse de l'usage seulement les pages visitées et enregistrées dans les logs, il est difficile de détecter les thèmes qui n'intéressent pas les utilisateurs ou les pages non accessibles. Par conséquent, les connaissances sur le contenu ou la structure d'un site Web en utilisant seulement des données sur l'usage sont tronquées.
- Évaluation de la structure du site à partir de l'analyse du contenu et de l'analyse de l'usage. Les résultats de cette évaluation permettent d'améliorer la structure du site afin de garantir la cohérence entre la structure et le contenu et faciliter la navigation et l'accès aux pages les plus intéressantes

aux utilisateurs.

- Indépendance totale de la méthode de classification aussi bien au niveau de l'analyse textuelle qu'au niveau de l'analyse de l'usage. L'utilisateur final peut choisir entre plusieurs algorithmes de block clustering adaptés aux tableaux de contingence (section 3.6.3). La méthode de classification se place sur un plus haut niveau d'abstraction.
- Indépendance par rapport à la langue du site Web. En effet, l'approche WCUM propose une méthodologie générique pour l'introduction du contenu dans l'analyse de l'usage d'un site Web. Seules la lemmatisation et l'étiquetage dépendent de la langue des textes. Il suffit donc de choisir un étiqueteur dans la langue du site. Treetagger, par exemple, est adapté à plusieurs langues telles que l'anglais, l'allemand, l'italien, le chinois, le portugais, etc.

6.7 Synthèse

Dans ce chapitre, nous avons présenté une nouvelle approche WCUM reliant l'analyse du contenu à l'analyse de l'usage. L'apport d'une telle approche consiste à analyser le contenu du site indépendamment de son usage et améliorer l'organisation des pages du site afin de faciliter l'accès à l'information qu'il contient. D'autre part, la liaison entre l'analyse du contenu et l'analyse de l'usage permet d'identifier les thèmes qui intéressent le plus les visiteurs du site. Le concepteur sera donc tenu à promouvoir le contenu des pages relatives à ces thèmes et améliorer l'organisation de ces pages afin d'y faciliter l'accès.

Chapitre 7

Conclusion et perspectives

L'analyse des comportements des internautes sur des sites Web est un enjeu important, même en dehors de la sphère marchande. Les motivations de telles pratiques consistent à améliorer l'accès à l'information, revoir la structure et le contenu de certaines parties du site faiblement visitées, abandonner des développements sans succès, assister l'internaute pendant sa visite et adapter le site aux différents modes d'utilisation. En d'autres termes, l'analyse des comportements de navigation sur un site permet au concepteur d'évaluer si le site défini par sa structure et son contenu atteint ses objectifs. Ces différentes considérations, nous ont incités à proposer une nouvelle approche d'analyse d'un site Web.

7.1 Apports de la thèse

Cette étude a débuté avec un constat important mais assez évident : le contenu et l'usage d'un site Web ne sont point indépendants. En d'autres termes, analyser l'usage d'un site nécessite tout d'abord d'analyser son contenu d'une part et d'avoir une idée sur sa structure d'autre part. Ainsi, les trois axes du Web mining sont dépendants l'un de l'autre. Notre idée dans ce travail de recherche était d'analyser un site en se basant sur le contenu et l'usage. Les résultats de cette analyse servent d'une part à comprendre le comportement des visiteurs du site Web vis à vis du contenu et d'autre part à apporter des améliorations à la structure du site afin de faciliter l'accès à l'information véhiculée par les pages. Pour ce faire, nous avons alors proposé une approche générique composée de deux étapes principales : l'analyse du contenu textuel du site et l'analyse de l'usage. Les avantages de cette approche est qu'elle permet de résumer l'information contenue dans le site sous forme de thèmes ce qui facilite l'analyse de sites Web volumineux et pallier le problème de matrices creuses souvent rencontré dans le cas des tableaux croisant les navigations aux pages. En plus, l'introduction du contenu dans l'analyse de l'usage facilite le suivi des changements dans le contenu et l'in-

fluence de ces changements sur le comportement des internautes, ce qui s'avère essentiel avec la dynamicité que connaît le Web actuellement et le flux important d'information qu'il transmet. D'autre part, l'indépendance de l'approche WCUM par rapport aux méthodes de classification utilisées et par rapport à la langue et au thème du site favorisent son application à d'autres sites Web.

L'apport de cette thèse concerne également le soulèvement du problème d'estimation du nombre de classes dans les méthodes de bipartitionnement. En effet, bien que le recours à la classification simultanée des lignes et des colonnes d'un tableau des données a débuté depuis les années soixante-dix, et que les travaux sur le choix du nombre de classes dans les méthodes de classification simple sont abondants et variés, aucun critère n'a été proposé pour pallier ce problème dans le cadre des méthodes de bipartitionnement. Dans le cadre de cette thèse, nous avons adapté certains critères existant à ce contexte et proposé un nouveau critère applicable dans le cas bidimensionnel. Ces critères ont été testés sur des données artificielles qui simulent le contenu textuel d'un site Web dans sa représentation matricielle et dont la méthodologie de génération a été exposée en détail.

7.2 Perspectives

Au cours de ce travail de recherche, nous avons été confrontés à certaines limites qui ouvrent de nouvelles perspectives de recherche. Ces limites concernent en particulier les critères proposés pour l'estimation du nombre de classes dans le cadre de la classification croisée. Nous pouvons donc envisager de compléter l'étude expérimentale des critères de détermination du nombre de classes dans les méthodes de bipartitionnement par une étude portant sur des données réelles. Il est également possible d'adapter au contexte de bipartitionnement d'autres indices de validation externe tels que les critères d'information (AIC, BIC, etc.) et de tester l'ensemble des critères proposés dans le cadre de cette thèse à d'autres algorithmes de bipartitionnement qui nécessitent de fixer le nombre de classes sur les lignes et les colonnes, puisqu'aucun de ces critères, à part le Chi2 pénalisé, n'est spécifique à l'algorithme Croki2
D'autre part, l'application de l'algorithme Croki2 au contexte particulier de données artificielles ne présentant pas de structure de mélange et dont les proportions de classes sont quasi-égales a donné de résultats satisfaisants. Il serait donc intéressant de l'appliquer à des données présentant une structure de mélange et comparer les résultats obtenus avec ceux de l'algorithme CemCroki2.
Au niveau de l'approche WCUM, il est envisageable de l'appliquer auprès de concepteurs et des animateurs de sites Web en apportant à la structure du site les modifications jugées utiles par l'analyse afin de vérifier la cohérence entre la

structure sémantique et la structure logique du site et de confronter la vision éditoriale du site avec celle des internautes (telle qu'elle apparait dans les données de l'usage).
A long terme, il serait intéressant de proposer aux utilisateurs du site des pages reliées sémantiquement à la page visitée à travers un système de recommandations qui servira à faciliter la recherche d'information sur le site et l'accès à des pages "éloignées" de la page visitée au niveau de la structure mais qui lui sont proches sémantiquement.

Liste des publications

.1 Articles publiés dans des conférences

1. M. Charrad, Y. Lechevallier, G. Saporta, M. Ben Ahmed. *Block clustering for Web pages categorization.* Editeurs : Corchado, Emilio ; Yin, Hujun. Lecture Notes in Computer Science LNCS 5788, Intelligent Data Engineering and Automated Learning - IDEAL 2009, 10th International Conference, Burgos, Spain, pages 260-267, ISBN 978-3-642-04393-2, Springer, 2009.

2. M. Charrad, M. Ben Ahmed, Y. Lechevallier, G. Saporta. *Determining the number of clusters in CROKI2 algorithm.* In Proceedings of 1^{st} Meeting on Statistics and DataMining MSDM'09, Hammamet, Tunisia, pages 143-148, mars 2009.

3. M. Charrad, Y. Lechevallier, G. Saporta, M. Ben Ahmed. *Web Content Data Mining : la classification croisée pour l'analyse textuelle d'un site Web.* In Actes des 8èmes journées francophones Extraction et Gestion des Connaissances 2008, EGC'08, Revue des Nouvelles Technologies de l'Information (RNTI-E-11), Cépaduès-éditions, Volume I, Pages 43-54, Sophia Antipolis, France, 2008.

4. M. Charrad, M. Ben Ahmed, Y. Lechevallier. *Web Usage Mining : WWW Pages classification from Log files.* Actes de la 2ème conférence internationale sur les machines intelligentes, 5-7 Nov.2005, Tozeur, Tunisie.

.2 Autres genres de publications

5. M. Charrad, Y. Lechevallier , G. Saporta et M. Ben Ahmed. *Le bi-partitionnement : Etat de l'art sur les approches et les algorithmes.* Ecol'IA'08, mars 2008, Hammamet, Tunisie.

6. M. Charrad, M. Ben Ahmed, Y. Lechevallier. *Extraction des connaissances à partir des fichiers Logs.* Atelier fouille du Web, 6èmes journées francophones Extraction et Gestion des Connaissances EGC'06, Janvier 2006, Lille, France.

7. M. Charrad, Y. Lechevallier, G. Saporta, M. Ben Ahmed. *Détermination du nombre des classes dans l'algorithme CROKI2 de classification croisée.* In Actes des 9èmes journées francophones, Extraction et Gestion des Connaissances 2009, EGC'09, Cépaduès-éditions, pages 447-448, Strasbourg, France, 2009.

8. M. Charrad, Y. Lechevallier, G. Saporta, M. Ben Ahmed. *Approaches and algorithms for simultaneous clustering : State of the art.* Soumis à la revue MJAI (Mediterranean Journal of Artificiel Intelligence).

9. M. Charrad, Y. Lechevallier, M. Ben Ahmed, G. Saporta. *A Web Content and Usage based approach for web site analysis.* Accepté au 2010 IEEE International Symposium on Mining and Web (24th IEEE AINA Conference), 20-23 April 2010, Perth, Australie.

10. M. Charrad, Y. Lechevallier, M. Ben Ahmed, G. Saporta. *On the Number of Clusters in Block Clustering Algorithms.* Accepté au 23ème International FLAIRS Conférence, Ed. AAAI., 19-21 mai 2010, Daytona Beach, Floride, USA.

11. M. Charrad, Y. Lechevallier, M. Ben Ahmed, G. Saporta. *Détermination du nombre de classes dans les méthodes de bipartitionnement.* Accepté aux 17ème rencontres de la Société Francophone de la Classification 2010, du 9-11 juin 2010, Saint-Denis de la Réunion, France.

Bibliographie

[AJL08] A. Anli, Z. Jrad, and Y. Lechevallier. Une approche incrémentale dans le prétraitement des fichiers logs web : Application à l'analyse des usages d'un site web touristique sémantique. *Atelier Integration, interrogation et analyse de LOgs (ILO'08). XXVIème Congrès INFORSID (Fontainebleau)*, 2008.

[Aka73] H. Akaike. Information theory as an extension of the maximum likelihood principle. *In B.N. Petrov and F. Csaki (Eds), Proceedings of the Second international Symposium on information Theory, Budapest, Hongrie*, pages 267–281, 1973.

[AR94] R. Agrawal and R.Srikant. Fast algorithms for mining association rules. *In Jorge B. Bocca, Matthias Jarke and Carlo Zaniolo, editors, Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499, 1994.

[AS95] R. Agrawal and R. Srikant. Mining sequential patterns. *Proceedings of the 11th Int. Conf. on Data Engineering (ICDEŠ95)*, 1995.

[ASL08] A. Anli, A. Da Silva, and Y. Lechevallier. Spécification et développement du module d'analyse de l'usage. *Livrable sp8.2 projet eiffel, INRIA Rocquencourt, 2 septembre*, 2008.

[AZG00] M. Amini, H. Zaragoza, and P. Gallinari. Learning for sequence extraction tasks. *In proceedings Content-Based Multimedia Information Access, RIAO'2000, Paris, France*, pages 476–490, 2000.

[BDCKY02] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data : The order-preserving submatrix problem. *In Proceedings of the 6th International Conference on Computacional Biology (RECOMB'02)*, pages 49–57, 2002.

[Bes05] J. Besson. Découvertes de motifs pertinents pour l'analyse du transcriptome : Application à l'insulino-résistance. *Thèse de doctorat, INSA de Lyon*, 2005.

[Bez81] J.C. Bezdek. Pattern recognition with fuzzy objective function algorithms. *Plenum Press, New York*, 1981.

[B.G84] B.Ganter. Two basics algorithms in concept analysis. *Technical report, Technische Hochschule Darmstadt*, 1984.

[BH75] F. B. Baker and L. J. Hubert. Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association*, pages 31–38, 1975.

[BJK02] S. Busygin, G. Jacobsen, and E. Kramer. Double conjugated clustering applied to leukemia microarray data. *In Proceedings of the 2nd SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data, Arlington Virginia, USA*, 2002.

[BLL98] C. Burgess, K. Livesay, and K. Lund. Explorations in context space : Words, sentences, discourse. *Discourse Processes, vol. (25)*, pages 211–257, 1998.

[BLS03] Y. Bestgen, L.Degand, and W. Spooren. On the use of automatic techniques to determine the semantics of connectives in large newspaper corpora : an explorative study. *In Lagerwerf L., Spooren W. et Degand L. (Eds), Determination of information and tenor in texts : Multidisciplinary approaches to discourse 2003. Nodus Publikationen*, pages 189–202, 2003.

[Blu67] H. Blum. A transformation for extracting new descriptors of shape. *Models for Perception of Speech and Visual Form*, pages 362–380, 1967.

[BMEBS98] B. Bigi, R. De Mori, M. El-Bèze, and T. Spriet. Detecting topic shifts using a cache memory. *Proceedings of the fifth Conference on Spoken Language Processing*, 1998.

[BMSN02] B. Berendt, B. Mobasher, M. Spiliopoulou, and M. Nakagawa. The impact of site structure and user environment on session reconstruction in web usage analysis. *Proceedings of the 4th WebKDD 2002 Workshop, at the ACM SIGKDD Conference (KDD'2002) on Knowledge Discovery in Databases*, 2002.

[Boc79] H. Bock. Simultaneous clustering of objects and variables. *In E., editor, Analyse des Données et Informatique*, pages 187–203, 1979.

[Boc03] H. Bock. Two-way clustering for contingency tables maximizing a dependence measure. *In Schader, M., Gaul, W., and Vichi, M., editors, Between Data Science and Applied Data Analysis*, pages 143–155, 2003.

[Bor86] J. Bordat. Calcul pratique du treillis de galois d'une correspondance. *Mathématique, Informatique et Sciences Humaines*, 1986.

[Boz81] H. Bozdogan. Multi-sample cluster analysis and approaches to validity studies in clustering individuals. *Thèse de Doctorat, Department of Mathematics, University of Illinois*, 1981.

[Boz83] H. Bozdogan. Determining the number of component clusters in the standard multivariate normal mixture model using model-selection criteria. *Rapport Technique UIC/DQM/A83-1, Quantitative Methods Department, University of Illinois*, 1983.

[BT03] A. Benedek and B. Trousse. Visualization adaptation of self-organizing maps for case indexing. *In 27th Annual Conference of the Gesellschaft für Klassifikation, Germany*, 2003.

[BWS03] A. Basu, C. Watters, and M. Shepherd. Support vector machines for text categorization. *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03). Big Island, Hawaii*, 2003.

[BYR99] R. Baeza-Yates and B.d.A.N. Ribeiro. Modern information retrieval. *New York : ACM Press /Addison-Wesley*, 1999.

[CAL06] M. Charrad, M. Ben Ahmed, and Y. Lechevallier. Extraction des connaissances à partir des fichiers logs. *Atelier fouille du Web EGC2006*, pages 768–769, 2006.

[CBT04] S. Chelcea, P. Bertrand, and B. Trousse. A new agglomerative 2-3 hierarchical clustering algorithm. *Innovations in Classification, Data Science, and Information Systems. Proc. 27th Annual GfKl Conference*, pages 3–10, 2004.

[CC00] Y. Cheng and G. M. Church. Biclustering of expression data. *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB'00)*, pages 93–103, 2000.

[CCB02] K. Chevalier, V. Corruble, and C. Bothorel. Surfminer : Connaître les utilisateurs d'un site. *in DVP2002 (Documents Virtuels Personnalisables), ENST Bretagne Brest, France*, pages 1–12, 2002.

[CDF+98] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. *In Proceedings of the Fifteenth National Conference on Artificial Intellligence (AAAI98)*, pages 509–516, 1998.

[CFMT07] G. Castellano, A.M. Fanelli, C. Mencar, and M.A. Torsello. Similarity-based fuzzy clustering for user profiling. *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops, November, Silicon Valley*, pages 75–78, 2007.

[CH74] R.B. Calinsky and J. Harabsz. A dendrite method for cluster analysis. *Communications in statistics*, pages 1–27, 1974.

[CH06] R.C. Chen and C.H. Hsieh. Web page classification based on a support vector machine using a weighted vote schema. *Expert Systems with Applications, Vol. 31*, pages Pages 427–435, 2006.

[Cha00] S. Chakrabarti. Data mining for hypertext : a tutorial survey. *ACM SIGKDD Explorations*, pages 1–11, 2000.

[Cha05] M. Charrad. Techniques dŠextraction des connaissances appliquées aux données du web. *Mémoire de Mastère présenté en vue de lŠobtention du diplôme de Mastère en Informatique, Ecole Nationale des Sciences de lŠInformatique de Tunis, Laboratoire RIADI*, 2005.

[Che69] M. Chen. Algorithme de recherche des sous-matrices premières d'une matrice. *Bulletin Mathématique de la Sociologie Scientifique de la R.S. de Roumanie*, 1969.

[CHM+00] I.V. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Visualization of navigation patterns on a web site using model-based clustering. *In Proceedings of the sixth ACM SIGKDD*, pages 280–284, 2000.

[Cho00] F.Y.Y Choi. Advances in domain independent linear text segmentation. *Proceedings of NAACL-00*, pages 26–33, 2000.

[CLP01] M. Chavent, C. Lacomblez, and B. Patouille. Critère de rand asymétrique. *Actes des 8èmes rencontres de la Société Francophone de Classification (SFC01), Pointe à Pitre, Guadeloupe*, pages 82–88, 2001.

[CMS97] R. Cooley, B. Mobasher, and J. Srivastava. Grouping web page references into transactions for mining world wide web browsing patterns. *Technical Report TR 97-021.Dept. of Computer science, Univ. of Minnesota, Minneapolis, USA*, 1997.

[CMS99] R. Cooley, B. Mobasher, and J. Sirvastava. Data preparation for miningworldwide web browsing patterns. *Journal of Knowledge and Information Systems, 1(1)*, pages 55–32, 1999.

[Coh99] W. W. Cohen. What can we learn from the web? *In Proceedings of the Sixteenth International Conference on Machine Learning (ICML'99)*, pages 515–521, 1999.

[Coo00] R. Cooley. Web usage mining : Discovery and application of interesting patterns from web data. *Phd thesis, University of Minnesota*, 2000.

[CP95] L.D. Catledge and J.E. Pitkow. Characterizing browsing strategies in the world wide web. *Computer Networks and ISDN Systems 27, Elsevier Science*, pages 1065–1073, 1995.

[CR96] C. Carpineto and G. Romano. A lattice conceptual clustering system and its applications to browsing retrieval. *Machine Learning, Volume 24, Number 2, Springer*, pages 95–122, 1996.

[CST00] A. Califano, G. Stolovitzky, and Y. Tu. Analysis of gene expression microarrays for phenotype classification. *In Proceedings of the 6th International Conference on Computacional Biology (RECOMB'02)*, pages 75–85, 2000.

[CWHM01] F. Choi, P. Wiemer-Hastings, and J. Moore. Latent semantic analysis for text segmentation. *In Proceedings of NAACL'01*, pages 109–117, 2001.

[CY96] M.S. Chen and J.S. Yu. Data mining for path traversal patterns in a web environment. *In Sixteenth International Conference on Distributed Computing Systems*, pages 385–392, 1996.

[DA97] D. Beeferman D. and A. Berger A. Text segmentation using exponential models. *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 35–46, 1997.

[DAF98] DAF. Dictionnaire de l'académie française. *5ème Edition (1798)*, page 767, 1798.

[DAF32] DAF. Dictionnaire de l'académie française. *8ème Edition*, 1932.

[DB79] D.L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Trans. Pattern Anal. Machine Intell. 1 (4)*, pages 224–227, 1979.

[DB03] L. Degand and Y. Bestgen. Towards automatic retrieval of idioms in french newspaper corpora. *Literary and Linguistic Computing, vol. (18)*, pages 249–259, 2003.

[DDF+90] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic indexing. *Journal of the American Society for Information Science, 41(6)*, pages 391–407, 1990.

[DGG+94] J. Dubois, L. Guespin, M. Giacomo, C. Marcellesi, J-B. Marcellesi, and J-P. Mével. Dictionnaire de linguistique et des sciences du langage. *Collection Trésors du Français, Larousse, Paris*, 1994.

[Dhi01] Dhillon. Coclustering documents and words using bipartite spectral graph partitioning. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), California, USA*, pages 269–274, 2001.

[DL99] P. Dessus and B. Lemaire. Apex, un système dŠaide à la préparations des examens. *Sciences et Technologies Educatives. 6(2)*, pages 409–415, 1999.

[DLV00] P. Dessus, B. Lemaire, and A. Vernier. Free-text assessment in virtual campus. *in K. Zreik (ed), proc third internationnal conference on human system learning (CAPS'3)*, pages 61–76, 2000.

[DMM03] I.S. Dhillon, S. Mallela, and D.S. Modha. Information-theoretic coclustering. *In ACM SIGKDD, Washington, DC,USA, ACM*, pages 89–98, 2003.

[DPHS98] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. *Proceedings of the seventh International Conference on Information and Knowledge Management (CIKM' 98)*, pages 148–155, 1998.

[Dum95] Dumais. Latent semantic indexing (lsi) : Trec-3 report. *In Harman D. (Ed.), Proceedings of The 3rd Text Retrieval Conference (TREC-3) l*, pages 219–230, 1995.

[Dum97] S.T. Dumais. Using latent semantic indexing for information retrieval, information filtering and other things. *Cognitive Technology Conference*, 1997.

[Dun73] J.C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-seperated clusters. *J. Cybernetics*, pages 32–57, 1973.

[Dun74] J. C. Dunn. Well separated clusters and optimal fuzzy partitions. *Journal Cybern.*, pages 95–104, 1974.

[ERCG+04] A. ElGolli, F. Rossi, B. Conan-Guez, D. Tanasa, B. Trousse, and Y. Lechevallier. Une application des cartes topologiques autoorganisatrices à l'analyse des fichiers logs. *In Actes des onzièmes journées de la Société Francophone de Classification, Bordeaux, France*, pages 181–184, 2004.

[Fay96] U. M. Fayyad. Data mining and knowledge discovery : making sense out of data. *IEEE Expert*, pages 20–25, 1996.

[FKL98] P.W. Foltz, W. Kintsch, and T.K. Landauer. The measurement of textual coherence with latent semantic analysis. *Discourse Processes, vol. 25*, pages 285–307, 1998.

[FKS+04] M. Friedman, A. Kandel, M. Schneider, M. Last, B. Shapka, Y. Elovici, and O. Zaafrany. A fuzzy-based algorithm for web document clustering. pages 524–527, 2004.

[FL05] F.M. Facca and P.L. Lanzi. Mining interesting knowledge from weblogs : a survey. *Data and Knowledge engineering, Vol. 53, No. 3*, pages 225–241, 2005.

[FLL99] P.W. Foltz, D. Laham, and T.K. Laundauer. The intelligent essay assessor : applications to educational technology. *Intercative Multimedia Electronic Journal of computer Enhanced Learning, 1(2)*, 1999.

[For65] E.W. Forgy. Cluster analysis of multivariate data : efficiency versus interpretability of classifications. *Biometrics*, pages 768–769, 1965.

[For06] D. Forest. Application de techniques de forage de textes de nature prédictive et exploratoire à des fins de gestion et d'analyse thématique de documents textuels non structurés. *Thèse de doctorat, Montréal, Université du Québec à Montréal*, 2006.

[Fér87] J-F. Féraud. Dictionnaire critique de la langue française, classification (page 458). *(Marseille, Mossy 1787-1788)*, 1787.

[FSS00] Y. Fu, K. Sandhu, and M. Shih. A generalization-based approach to clustering of web usage sessions. *In Proceedings of the 1999 KDD Workshop on Web Mining, San Diego, Springer*, pages 21–38, 2000.

[Gav02] G. Gavray. Personnalisation des sites web : Elaboration d'une méthodologie de mise en oeuvre et application au cas de dgtre. *Mémoire projet en vue de l'obtention du titre Ingénieur de gestion, Université Catholique de Louvain*, 2002.

[GK54] L. Goodman and W. Kruskal. Measures of association for cross-validation. *J. Am. Stat. Assoc. 49*, pages 732–764, 1954.

[GK79] D.E. Gustafson and W. C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. *In Proc. IEEE Conf. on Decision Control Internat. 17th Symp. on Adaptive Processes, San Diego, California*, pages 761–766, 1979.

[GLD00] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci. USA*, 2000.

[GMA91] R. Godin, R. Missaoui, and H. Alaoui. Learning algorithms using a galois lattice structure. *In Proceedings of the 3rd International Conference on Tools for Artificial Intelligence, IEEE Computer Society Press, San Jose, California*, pages 22–29, 1991.

[GMD08] C. Gomathi, M. Moorthi, and K. Duraiswamy. Preprocessing of web log files in web usage mining. *Journal of Information Technology, Vol. 4, No. 1*, pages 55–66, 2008.

[GN03] G. Govaert and M. Nadif. Clustering with block mixture models. *Pattern Recognition*, pages 463–473, 2003.

[GN05a] G. Govaert and M. Nadif. Classification croisée d'un tableau de contingence et modèle probabiliste. *RNTI-E-3, Revue des Nouvelles Technologies de l'Information, Vol. 1*, pages 213–218, 2005.

[GN05b] G. Govaert and M. Nadif. An em algorithm for the block mixture models. *IEEE Transactions on Pattern Analysis and machine intelligence*, pages 643–647, 2005.

[GN08] G. Govaert and M. Nadif. Block clustering with mixture models : Comparison of different approaches. *Computational Statistics and Data Analysis, 52*, pages 3233–3245, 2008.

[Gov77] G. Govaert. Algorithme de classification d'un tableau de contingence. *In First international symposium on data analysis and informatics*, pages 487–500, 1977.

[Gov83] G. Govaert. Classification croisée. *Thèse de doctorat d'état, Paris*, 1983.

[HA85] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification, 2*, pages 193–198, 1985.

[Har72] J.A. Hartigan. Direct clustering of a data matrix. *Journal of American Statistical Association, Vol.67, N.337*, pages 123–129, 1972.

[Har75] J. Hartigan. Direct splitting. *Dans Clustering Algorithms. John Wiley and Sons, New York*, pages 251–277, 1975.

[HBV01] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems 17 :2/3*, pages 107–145, 2001.

[Hea97] M. Hearst. Texttiling : Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics, Vol. 23*, pages 33–64, 1997.

[HKKM97] E.H. Han, G. Karypis, V. Kumar, and B. Mobasher. Clustering based on association rule hypergraphs. *In Research Issues on Data Mining and Knowledge Discovery*, pages 9–13, 1997.

[HL76] L.J. Hubert and J.R. Levin. A general statistical framework for assesing categorical clustering in free recall. *Psychological Bulletin*, pages 1072–1080, 1976.

[HMS09] T. Hasan, S.P. Mudur, and N. Shiri. A session generalization technique for improved web usage mining. *Proceeding of the 11th international workshop on Web information and data management, Hong Kong, China*, pages 23–30, 2009.

[HPY00] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *In Weidong Chen, Jeffrey Naughton, and Philip A.*

Bernstein, editors, 2000 ACM SIGMOD International Conference on Management of Data, ACM Press, pages 1–12, 2000.

[HZDS02] X. He, H. Zha, C. Ding, and H. Simon. Web document clustering using hyperlink structures. *Computational Statistics and Data Analysis, Vol. 41, No. 1*, pages 19–45, 2002.

[HZS06] J. Huang, T. Zhu, and D. Schuurmans. Web communities identification from random walks. *Lecture Notes in Computer Science, N4213, Springer*, pages 187–198, 2006.

[ISS04] F. Ibekwe-SanJuan and E. SanJuan. Mining textual data through term variant clustering : the termwatch system. *RIAO 2004 Coupling approaches, coupling media and coupling languages for information retrieval*, pages 487–503, 2004.

[Jal03] R. Jalam. Apprentissage automatique et catégorisation de textes multilingues. *thèse de doctorat en informatique, Université lumière lyon2*, 2003.

[JD88] A. K. Jain and R. C. Dubes. Algorithms for clustering. *Data. Englewood Cliffs, NJ : Prentice Hall, New Jersey*, 1988.

[JMF99] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering : a review. *ACM Computing Surveys Vol.31, no. 3*, pages 264–323, 1999.

[Joa98] T. Joachims. Text categorization with support vector machines : Learning with many relevant features. *Proceedings of the Tenth European Conference on Machine Learning (ECML'98), Springer Verlag*, pages 137–142, 1998.

[Joa02] T. Joachims. Learning to classify text using support vector machines. *Dordrecht : Kluwer Academic Publishers*, 2002.

[Jol03] F.-X. Jollois. Contribution de la classification automatique à la fouille de données. *Thèse de doctorat, Université de Metz, 12 décembre 2003*, 2003.

[KBJG03] Y. Kluger, R. Basri, T. Joseph, and C.M. Gerstein. Spectral biclustering of microarray data : coclustering genes and conditions. *In Genome Research, vol. 13*, pages 703–716, 2003.

[Kin00] W. Kintsch. Metaphor comprehension : A computational theory. *Psychonomic Bulletin and Review, vol. (7)*, pages 257–266, 2000.

[KKM98] M.Y. Kan, J.L. Klavans, and K.R. McMeown. Linear segmentation and segment significance. *In Proceedings of the 6th International Workshop of Very Large Corpora (WVLC-6), Montreal, Canada*, pages 197–205, 1998.

[KL88] W.J. Krzanowski and Y.T. Lai. A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics 44*, pages 23–34, 1988.

[Kle99] J.M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the Association for Computing Machinery 46*, pages 604–632, 1999.

[KM00] R. Kimball and R. Merz. Le data webhouse : Analyser des comportements clients sur le web. *Editions Eyrolles, Paris*, 2000.

[KNY00] H. Kato, T. Nakayama, and Y. Yamane. Navigation analysis tool based on the correlation between contents distribution and access patterns. *In Worskhop on Web Mining for E-Commerce - Challenges and Opportunities, WEBKDD 2000, Boston, MA, USA*, pages 95–104, 2000.

[Koz93] H. Kozima. Text segmentation based on similarity between words. *In Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 286–288, 1993.

[KPWD03] H. Kum, J. Pei, W. Wang, and D. Duncan. Approxmap : Approximate mining of consensus sequential patterns. *Proceedings of SIAM Int. Conf. on Data Mining*, 2003.

[KV09] V.A. Koutsonikola and A.I. Vakali. A fuzzy bi-clustering approach to correlate web users and pages. *Int. J. Knowledge and Web Intelligence, Vol. 1, Numbers 1-2*, pages 3–23, 2009.

[KWH02] B. Klebanov and P. Wiemer-Hastings. Using lsa for pronominal anaphora resolution. *In Computational Linguistics and Intelligent Text Processing : Third International Conference, CICLing.*, pages 249–259, 2002.

[Lab06] N. Labroche. Fast ant-inspired clustering algorithm for web usage mining. *In Proc. of the Information Processing and Management of Uncertainty Conference, IPMU'06, Paris, France*, pages 2668–2675, 2006.

[Lab07] N. Labroche. Learning web users profiles with relational clustering algorithms. *In Workshop on Intelligent Techniques for Web Personalization AAAI Conference*, 2007.

[Lab09] A. Labadié. Segmentation thématique de texte linéaire et non-supervisée : Détection active et passive des frontières thématiques en français. *Thèse de doctorat, Université Montpellier II*, 2009.

[LBSN01] B. Lemaire, M. Bianco, E. Sylvestre, and I. Noveck. Un modèle de compréhension de textes fondé sur lŠanalyse de la sémantique

latente. *In Paugam Moisy H., Nyckees V. et Caron-Pargue J.(Eds), La cognition entre individu et société (actes du colloque de lŠARCo). Hermès*, pages 309–320, 2001.

[LD97a] T.K. Landauer and S.T. Dumais. A solution to plato's problem : the latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review, vol. 104*, pages 211–240, 1997.

[LD97b] T.K. Landauer and S.T. Dumais. A solution to plato's problem : the latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review, vol. 104*, page 211Ũ240, 1997.

[Lec04] Y. Lechevallier. Classification croisée d'un tableau de données symboliques. *in : 11èmes Rencontres de la Société Francophone de Classification (SFC), Bordeaux, France*, pages 245–248, 2004.

[Lew92] D. D. Lewis. Representation and learning in information retrieval. *Ph.D.thesis, Graduate School of the University of Massachusetts*, 1992.

[LFL98] T.K. Landauer, P.W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes, vol.(25)*, pages 259–284, 1998.

[LHY05] X. Liu, P. He, and Q. Yang. Mining user access patterns based on web logs. *Canadian Conference on Electrical and Computer Engineering , Saskatoon Sk, Canada*, pages 2280–2283, 2005.

[LLY07] N. Labroche, M.J. Lesot, and L. Yaffi. A new web usage mining and visualization tool. *19th Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, Vol. 01, Patras, Greece*, pages 321–328, 2007.

[LMC03] B. Lazzerini, F. Marcelloni, and M. Cococcioni. A system based on hierarchical fuzzy clustering for web users profiling. *IEEE International Conference on Systems, Man and Cybernetics*, pages 1995–2000, 2003.

[LMP00] L. Lebart, A. Morineau, and M. Piron. Statistique exploratoire multidimensionnelle. *(3ème édition). Dunod.*, 2000.

[LNCC04] M. Lin, J. F. Nunamaker, M. Chau, and H. Chen. Segmentation of lecture videos based on text : A method combining multiple linguistic features. *In Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS-37), Big Island, HI, Track 1, Vol. 1*, 2004.

[LO00] L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Technical report, Stanford University*, 2000.

[LS92] L. Lebart and A. Salem. Statistique textuelle. *Dunod*, 1992.

[LS98] M. Liquière and J. Sallantin. Structural machine learning with galois lattice and graphs. *In Proceedings of 15th International Conference on Machine Learning, San Francisco, California*, pages 305–313, 1998.

[LTTV03] Y. Lechevallier, D. Tanasa, B. Trousse, and R. Verde. Classification automatique : Applications au web mining. *In Yadolah Dodge and Giuseppe Melfi, editors, Méthodes et Perspectives en Classification, 10èmes Rencontre de la Société Francophone de Classification, Neuchâtel, Suisse*, pages 157–160, 2003.

[Luh57] H.P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, pages 309–317, 1957.

[Mar87] F. Marchotorchino. Block seriation problems : A unified approach. *Applied Stochastic Models and Data Analysis*, pages 73–91, 1987.

[MBB04] I. Van Mechelen, H. Bock, and P. De Boeck. Two-mode clustering methods : a structured overview. *Statist. Methods Medical Res., 13(5)*, pages 363–394, 2004.

[MBNL99] K. Madria, S. S. Bhowmick, W. K. Ng, and E.P. Lira. Research issues in web data mining. *In Proceedings of Data Warehousing and Knowledge Discovery*, pages 303–312, 1999.

[MC86] G.W. Milligan and M.C. Cooper. Study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research 21, no.4*, pages 441–458, 1986.

[MCP98] F. Masseglia, F. Cathala, and P. Poncelet. The psp approach for mining sequential patterns. *Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery*, 1998.

[MH91] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, pages 21–48, 1991.

[MHLN02] B. Mobasher, D. Honghua, T. Luo, and M. Nakagawa. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery, Vol. 6, No. 1*, pages 61–82, 2002.

[Mic02] C. Michel. Le web usage mining, méthodes et expérimentations pour l'extraction d'information et de connaissances sur les données

du web. *Actes de la 7ème conférence AIM : Affaire Electronique et société de savoir, opportunités et défis, Hammamet, Tunisie*, 2002.

[Mir96] B. Mirkin. Mathematical classification and clustering. *Dordrecht : Kluwer*, 1996.

[MJJ09] N. Mustapha, M. Jalali, and M. Jalali. Expectation maximization clustering algorithm for user modeling in web usage mining systems. *European Journal of Scientific Research, Vol.32, No.4*, pages 467–476, 2009.

[MJM09] M.N.Noorhuzaimi, S. Junaida, and R. M. Mazrul. An analysis of services using association rules. *ICCSIT, 2nd International Conference on Computer Science and Information Technology*, pages 469–473, 2009.

[MM06] A. Marascu and F. Masseglia. Extraction de motifs séquentiels dans les flots de données d'usage du web. *Extraction et Gestion des Connaissances (EGC'06), Lille*, pages 627–638, 2006.

[MO04] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis : A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 24–45, 2004.

[Mob99] B. Mobasher. A web personalization engine based on user transaction clustering. *Proceedings of the 9th Workshop on Information Technologies and Systems*, pages 179–184, 1999.

[Mou97] I. Moulinier. Apprentissage et acquisition de connaissances. *Thèse de l'université Paris VI*, 1997.

[MPC99a] F. Masseglia, P. Poncelet, and R. Cicchetti. An efficient algorithm for web usage mining. *Networking and Information Systems Journal (NIS), Vol. 2*, pages 571–603, 1999.

[MPC99b] F. Masseglia, P. Poncelet, and R. Cicchetti. Webtool : un environnement intégré de data mining. *INFORSID'1999, La Garde, France*, pages 393–411, 1999.

[MRGG05] J.A. Mojica, D.A. Rojas, J. Gomez, and F. Gonzalez. Page clustering using a distance based algorithm. *Third Latin American Web Congress, October, Buenos Aires, Argentina*, page 7, 2005.

[MS99] C. D. Manning and H. Schütze. Foundations of statistical natural language processing. *Cambridge (Mass.) : MIT Press*, 1999.

[MTT03] F. Masseglia, D. Tanasa, and B. Trousse. Diviser pour découvrir : une méthode d'analyse du comportement de tous les utilisateurs d'un site web. *19ème journées de Bases de Données Avancées, Lyon, France*, 2003.

[Nak07] D. Nakache. Extraction automatique des diagnostics à partir des comptes rendus médicaux textuels. *Thèse de doctorat en informatique, CNAM Paris*, pages 245–248, 2007.

[NER02] E.S. Nan Niu and M. El-Ramly. Understanding web usage mining for dynamic web-site adaptation : A case study. *In proceedings of the 4th International Workshop on Web Site Evolution (WSE'02), IEEE, Montréal, Canada*, pages 53–64, 2002.

[NG05] M. Nadif and G. Govaert. Block clustering of contingency table and mixture model. *Intelligent Data Analysis IDA'2005, LNCS 3646, Springer-Verlag Berlin Heidelberg*, pages 249–259, 2005.

[NN98] E. M. Nguifo and P. Njiwoua. Using lattice-based framework as a tool for feature extraction. *In European Conference on Machine Learning*, pages 304–309, 1998.

[Nor78] E.M. Norris. An algorithm for computing the maximal rectangles in a binary relation. *Revue Roumaine de Mathématiques Pures et Appliquées 23 (2)*, 1978.

[Oue01] T. Ouerfelli. La segmentation des documents techniques composites dans une perspective d'indexation : vers la définition d'un modèle dans une optique d'automatisations. *Thèse de Doctorat en sciences de l'information et de la communication, Université Stendhal de Grenoble 3*, 2001.

[PAV07] G. Pallis, L. Angelis, and A. Vakali. Validation and interpretation of web users. *Information, Processing and Management, Vol. 43, No. 5*, pages 1348–1367, 2007.

[PBZ+06] A. Prelic, S. Bleuler, P. Zimmermann, A.Wille, P. Bhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics, 9(22)*, page 1122Ű1129, 2006.

[PE98] M. Perkowitz and O. Etzioni. Adaptive web sites : automatically synthesizing web pages. *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 727–732, 1998.

[PHMA+01] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Prefixspan : Mining sequential patterns efficiently by prefix-projected pattern growth. *17th International Conference on Data Engineering (ICDE)*, 2001.

[PKVP08] S. Petridou, V. Koutsonikola, A. Vakali, and G. Papadimitriou. Time-aware web users clustering. *IEEE Transaction on Knowledge and Data Engineering, Vol. 20, No. 5*, pages 653–667, 2008.

[Por80] M. Porter. An algorithm for suffix stripping. *Automated Library and Information Systems*, pages 130–137, 1980.

[PPPS03] D. Pierrakos, G. Paliouras, C. Papatheodorou, and C.D. Spyropoulos. Web usage mining as a tool for personalization : A survey. *User Modeling and User-Adapted Interaction Vol. 13, No. 4*, pages 311–372, 2003.

[PPTK02] B. Prasetyo, I. Pramudiono, K. Takahashi, and M. Kitsuregawa. Naviz : Website navigational behavior visualizer. *In Ming-Shan Cheng, Philip S. Yu, and Bing Liu, editors, Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, Springer, Vol. 2336, Taipei, Taiwan*, pages 276–289, 2002.

[QS04] D. Qi and B. Sun. A genetic k-means approaches for automated web page classification. *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, IRI'2004*, pages 242–246, 2004.

[Qui93] J.R. Quinlan. C4.5 : Programs for machine learning. *San Mateo (Calif.) : Morgan Kaufman*, 1993.

[Rak05] R. Rakotomalala. Tanagra, une plate-forme d'expérimentation pour la fouille de données. *Revue MODULAD*, pages 70–85, 2005.

[Ram05] A.A.B. Ramli. Web usage mining using apriori algorithm : Uum learning care portal case. *International Conference on Knowledge management, Malaysia*, pages 1–19, 2005.

[Ran71] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association 66, no. 336*, pages 846–850, 1971.

[RB97] M. Rajman and R. Besançon. Text mining : Natural language techniques and text mining applications. *In Proceedings of the seventh IFIP 2.6 Working Conference on Database Semantics. Chapam and Hall*, 1997.

[RC06] M. Roche and J. Chauché. Lsa : les limites d'une approche statistique. *In Actes de l'atelier FDC'06 (Fouille de Données Complexes), conférence EGC'2006*, pages 95–106, 2006.

[Rey98] J.C. Reynar. Topic segmentation : Algorithms and applications. *Phd thesis, University of Pennsylvania, Seattle, WA*, 1998.

[RGL05] F. Rossi, A. El Golli, and Y. Lechevallier. Usage guided clustering of web pages with mediann self organizing map. *In proceedings of ESANN'05*, 2005.

[Réh05] S. Réhel. Catégorisation automatique de textes et cooccurrence de mots provenant de documents non étiquetés. *Thèse. Maître ès sciences (M.Sc.), Université Laval*, 2005.

[RHS09] A.M. Rahmani, Z. Hossaini, and S. Setayeshi. Link processing for fuzzy web pages clustering and classification. *European Journal of Scientific Research, Vol. 27, No. 4*, pages 620–627, 2009.

[Rij79] V. Rijsbergen. Information retrieval. *London : Butterworths*, 1979.

[Ril95] Riloff. In proceedings of the 18th annual international acm sigir. *Conference on Research and Development in Information Retrieval*, pages 130–136, 1995.

[Rob02] C. Robardet. Contribution à la classification non superviséee : proposition d'une méthode de bi-partitionnement. *thèse de doctorat, Université Claude Bernard - Lyon 1*, 2002.

[Rou85] M. Roux. Algorithmes de classification. *Edition Masson*, 1985.

[Rou87] P.J. Rousseeuw. Silhouettes : a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, pages 53–65, 1987.

[RSW+98] B. Rehder, M.E. Schreiner, M.B. Wolfe, D. Laham, T.K. Landauer, and W. Kintsch. Using latent semantic analysis to assess knowledge : some technical considerations. *Discourse Processes, 25*, pages 337–354, 1998.

[Rus69] E. H. Ruspini. A new approach to clustering. *Inform. Control, 15(1)*, pages 22–32, 1969.

[RVZB09] C. Romero, S. Ventura, A. Zafra, and P.D. Bra. Applying web usage mining for personalising hyperlinks in web-based adaptive educational systems. *Computers and Education, Vol. 53*, pages 828–840, 2009.

[Sah98] M. Sahami. Using machine learning to improve information access. *Ph.D. Dissertation, Stanford*, 1998.

[Sal88] G. Salton. Automatic text processing : The transformation, analysis and retrieval of information by computer. *Addison-Wesley Editors*, 1988.

[SCDT00] J. Srivastava, R. Cooley, M. Deshpande, and P.-N Tan. Web usage mining : Discovery and applications of usage patterns from web data. *SIGKKD Explorations*, 2000.

[Sch78] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics 6*, pages 461–464, 1978.

[Sch94] H. Schmidt. Probabilistic part-of-speech tagging using decision tree. *In Proc. Of the International Conference on New Methods in Language Processing, Manchester*, pages 44–49, 1994.

[Sch01] E. Schwarzkopf. An adaptive web site for the u.m.2001 conference. *In Proceedings of the U.M.2001 Workshop on Machine Learning for User Modeling*, pages 77–86, 2001.

[Seb99] F. Sebastiani. A tutorial on automated text categorisation. *Proceedings of ASAI-99*, 1999.

[Seb02] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys. Vol. 34*, pages 1–47, 2002.

[Seb05] F. Sebastiani. Text categorization. *In Alessandro Zanasi (ed.), Text Mining and its Applications, WIT Press, Southampton, UK. Forthcoming*, 2005.

[SFW99] M. Spiliopoulou, L.C. Faulstich, and K. Winkler. A data miner analysing the navigational behaviour of web users. *In Proc. of the workshop on Machine Learning in User Modeling of the ACAI'99 Int. Conf., Creta, Greece*, pages 588–589, 1999.

[SHP95] H. Schütze, D. A. Hull, and J. O. Pedersen. A comparison of classifiers and document representations for the routing problem. *Proceedings of the 18th Annual International Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 229–238, 1995.

[Sil09] A. G. Da Silva. Analyse des données évolutives : application aux données d'usage du web. *Thèse de doctorat, Paris*, 2009.

[SM99] S. Scott and S. Matwin. Feature engineering for text classication. *In Proceedings of Sixteenth International Conference on Machine Learning (ICML-99)*, 1999.

[Sow00] J.F. Sowa. Knowledge representation–logical, philosophical, and computational foundations. *Pacific Grove, Brooks/Cole*, 2000.

[SP01] M. Spiliopoulou and C. Pohle. Data mining for measuring and improving the success of web sites. *Special issue on applications of data mining to electronic commerce, Journal of Data Mining and Knowledge Discovery, Vol. 5*, pages 85–114, 2001.

[SS01] S.Kuznetsov and S.Obiedkov. Algorithms for the construction of concept lattices and their diagram graphs. *Principles of Data Mining and Knowledge Discovery (PKDD'2001), Freiburg, Germany*, 2001.

[STB+02] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal. Computing iceberg concept lattices with titanic. *Data and Knowledge Engineering, Vol.42, Amsterdam, Pays-Bas*, pages 189–222, 2002.

[Str00] M. Stricker. Réseaux de neurones pour le traitement automatique du langage : conception et réalisatin de filtres d'information. *Thèse de Doctorat, Electronique, ESPCI*, 2000.

[SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM, 18*, pages 613–620, 1975.

[Tan05] D. Tanasa. Web usage mining : Contributions to intersites logs preprocessing and sequential pattern extraction with low support. *Ph. D. Thesis, University of Nice Sophia Antipolis*, 2005.

[TG97] L. Tauscher and S. Greenberg. How people revisit web pages : Empirical findings and implications for the design of history systems. *International Journal of Human Computer Studies, Special issue on World Wide Web Usability 47(1)*, pages 97–138, 1997.

[TMT06] D. Tanasa, F. Masseglia, and B. Trousse. Gwum : une généralisation des pages web guidée par les usages. *INFORSID 2006*, pages 783–798, 2006.

[Tro00] B. Trousse. Evaluation of the prediction capability of a user behaviour mining approach for adaptative web sites. *In RIAO 2000, 6th Conference on Content-Based Multimedia Information Access, Collège de France, Paris, France*, 2000.

[TSS02] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *In Bioinformatics, volume 18 (Suppl. 1)*, pages 136–144,, 2002.

[TSS04] A. Tanay, R. Sharan, and R. Shamir. Biclustering algorithms : A survey. *In Handbook of Computational Molecular Biology, Edited by Srinivas Aluru, Chapman*, 2004.

[TT03] D. Tanasa and B. Trousse. Le prétraitement des fichiers logs web dans le web usage mining multi-sites. *In Journées Francophones de la Toile, Tours, France*, 2003.

[TT04] D. Tanasa and B. Trousse. Advanced data preprocessing for intersites web usage mining. *IEEE Intelligent Systems, Vol. 19, No. 2*, pages 59–65, 2004.

[TTM04] D. Tanasa, B. Trousse, and F. Masseglia. Classer pour découvrir : une nouvelle méthode d'analyse du comportement des tous les utilisateurs d'un site web. *In J.M. Petit, G. Hébrail, L. Lebart,*

editor, Revue des Nouvelles Technologies de l'Information (RNTI), Cépaduès-Editions, pages 549–560, 2004.

[TVJ06] G.S. Tomar, S. Verma, and A. Jha. Web page classification using modified naïvebayesian approach. *TENCON'2006, 2006 IEEE Region 10 Conference*, pages 1–4, 2006.

[UI01] M. Utiyama and H. Isahara. A statistical model for domain-independent text segmentation. *Proceedings of 39th Annual Meeting on Association for Computational Linguistics ACL'2001, Toulouse, France*, pages 499–506, 2001.

[Vic00] M. Vichi. Double k-means clustering for simultaneous of objects and variables. *In Borra, S. e. a., editor, Advances in Classification and Data Analysis*, 2000.

[VK01] M. Vichi and H.A.L. Kiers. Factoriel k-means analysis for two-way data. *Computational Statistics and Data Analysis, Vol. 37, No.1*, pages 49–64, 2001.

[VP07] Y. Kodratoff V. Prince. Le défi fouille de textes : Quels paradigmes pour la reconnaissance automatique d'auteurs ? *Revue des Nouvelles Technologies de l'Information - RNTI- E10 Cépaduès*, 2007.

[WH04] J. Wang and J. Han. Bide : Efficient mining of frequent closed sequences. *Proceedings of the International Conference on Data Engineering (ICDEŠ04)*, 2004.

[Wie93] E. D. Wiener. A neural network approach to topic spotting in text. *Ph.D. Dissertation, Stanford,*, 1993.

[WPW95] E. D. Wiener, J. O. Perdersen, and A. S. Weigend. A neural network approach for topic spotting. *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, pages 317–332, 1995.

[XB91] X. L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 84 1–847, 1991.

[XZC+02] G. Xue, H.J. Zeng, Z. Chen, W.Y. Ma, and C.J. Lu. Log mining to improve the performance of site search. *In WISE2002 Workshop on Mining Enhanced Web Search, Singapore*, 2002.

[YCG+98] J. Yamron, Carpi, L. Gillick, P. Lowe, and P. Van Mulbregt. A hidden markov model approach to text segmentation and event tracking. *IEEE ICASSP*, 1998.

[YJGMD96] T.W. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. Forme user access patterns to dynamichypertext linking. *WWW5/Computer Networks*, pages 1007–1014, 1996.

[YP97] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, pages 412–420, 1997.

[YP05] Y. Yang and B. Padmanabhan. Ghic : a hierarchical pattern-based clustering algorithm for grouping web transactions. *IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 9*, pages 1300–1304, 2005.

[YS04] G. Youness and G. Saporta. Une méthodologie pour la comparaison de partitions. *Revue de statistique Appliquée 52*, pages 97–120, 2004.

[YWWY03] J. Yang, H. Wang, W. Wang, and P. Yu. Enhanced biclustering on expression data. *In : 3rd IEEE International Symposium on BioInformatics and BioEngineering (BIBE 2003), IEEE Computer Society, Los Alamitos, CA*, pages 321–327, 2003.

[ZCM02] H-J. Zeng, Z. Chen, and Y-M. Ma. A unified framework for clustering heterogeneous web objects. *Proceedings of the 3rd International Conference on Web Information Systems Engineering*, pages 161–172, 2002.

[ZEH07] X. Zhang, J. Edwards, and J. Harding. Personalised online sales using web usage data mining. *Computers in Industry, Vol. 58*, pages 772–782, 2007.

[ZL01] V. Zampa and B. Lemaire. Latent semantic analysis for user modelling. *Journal of intelligent information systems. 18(1)*, pages 15–30, 2001.

[ZO07] B. Ziani and Y. Ouinten. Etude cas en web usage mining : catégorisation des utilisateurs de la connexion internet de l'uatl. *Revue de l'Information Scientifique et Technique, Vol. 17, No. 1,*, pages 89–96, 2007.

[ZR01] V. Zampa and F. Raby. Entre modèle dŠacquisition et outil pour l'apprentissage de la langue de spécialité : le prototype r.a.f.a.l.e.s. *Asp (Anglais de spécialité) 31-33*, pages 163–179, 2001.

[ZRL96] T. Zhang, R. Ramakrishnan, and M. Livny. Birch : An efficient data clustering method for very large databases. *in , H. V. Jagadish, , I. S. Mumick (eds), Proceedings of the 1996 ACM SIGMOD Int. Conf. on Management of Data*, pages 103–114, 1996.

[ZRZ01] C. Tang C.and L. Zhang, M. Ramanathan, and A. Zhang. Interrelated two-way clustering : An unsupervised approach for gene expression data analysis. *Proceedings of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering*, 2001.