Extraction et Gestion des Connaissances

Bordeaux, 31 janvier 2012

9ème Atelier sur la Fouille de Données Complexes

Complexité liée aux données multiples et massives

Responsables

Guillaume Cleuziou (LIFO, Université d'Orléans)

Cyril de Runz (CreSTIC, Université de Reims)

Nicolas Labroche (LIP6, Université Paris 6)

Mustapha Lebbah (LIPN, Université Paris 13)

Cédric Wemmert (LSIIT, Université de Strasbourg)

9ème Atelier sur la Fouille de Données Complexes Complexité liée aux données multiples et massives

Guillaume Cleuziou*, Cyril de Runz**, Nicolas Labroche***, Mustapha Lebbah****, Cédric Wemmert♯

*LIFO, Université d'Orléans
guillaume.cleuziou@univ-orleans.fr
**CReSTIC, Université de Reims
cyril.de-runz@univ-reims.fr
***LIP6, Université Paris 6
nicolas.labroche@univ-paris6.fr
****LIPN, Université Paris 13
mustapha.lebbah@univ-paris13.fr
♯LSIIT, Université de Strasbourg
wemmert@unistra.fr

Résumé. L'atelier sur la fouille de données complexes est proposé à l'instigation du groupe de travail EGC "Fouille de données complexes". Chaque année les organisateurs proposent une thématique de recherche qui suscite l'intérêt des chercheurs et des industriels.

Les deux précédentes éditions de l'atelier ayant permis de fédérer avec succès des recherches portant sur la problématique de complexité liée à la multiplicité des données, nous avons choisi de renouveler l'affichage de cette priorité thématique pour cette neuvième édition et de la compléter par deux nouvelles thématiques privilégiées, transversale pour la première et émergente pour la seconde, à savoir : les méthodes et avancées en clustering et la complexité liée aux données massives (fouille de données dans le *cloud*).

1 Le groupe de travail "Fouille de Données Complexes"

La neuvième édition de l'atelier sur la fouille de données complexes est organisée par le groupe de travail EGC "Fouille de Données Complexes". Ce groupe de travail rassemble une communauté de chercheurs et d'industriels désireux de partager leurs expériences et problématiques dans le domaine de la fouille de données complexes telles que le sont les données non-structurées (ou faiblement), les données obtenues à partir de plusieurs sources d'information ou plus généralement les données spécifiques à certains domaines d'application et nécessitant un processus d'extraction de connaissance sortant des itinéraires usuels de traitement.

Les activités du groupe de travail s'articulent autour de trois champs d'action progressifs :

– l'organisation de **journées scientifiques** une fois par an (vers le mois de juin) où sont présentés des travaux en cours ou plus simplement des problématiques ouvertes et pendant lesquelles une large place est faite aux doctorants,
– l'organisation de l'**atelier "Fouille de Données Complexes"** associé à la conférence EGC qui offre une tribune d'expression pour des travaux plus avancés et sélectionnés sur la base d'articles scientifiques par un comité de relecture constitué pour l'occasion,
– la préparation de **numéros spéciaux** de revue nationale, dans lesquels pourront être publiés les études abouties présentées dans un format long et évaluées plus en profondeur par un comité scientifique. Le second numéro spécial a été publié en 2011.

2 Contenu scientifique de l'atelier

Nous avons reçu cette année 10 propositions, chacune d'elle a été relue par au moins deux rapporteurs. Pour la grande majorité des propositions nous avons été en mesure de proposer trois rapports d'experts afin d'offrir un processus scientifique constructif aux auteurs. Nous avons retenu 7 propositions en fonction de leur degré d'avancement ainsi que de leur cohérence avec les thématiques de l'atelier.

Les articles qui vous sont proposés explorent :

– de nouvelles **problématiques** de recherche : introduction de connaissances expertes dans les processus de fouille, utilisation des technologies récentes d'informatique dans un envionement *cloud*, etc.
– de nouvelles **méthodologies** d'extraction de connaissance : nouvelles méthodes de classification, nouvelles formalisations pour l'entreposage et la sécurité des données, etc.

Une particularité commune à la majorité des travaux qui seront présentés dans cet atelier concerne le traitement de grands volumes de données (nombre d'individus ou dimensionalité) dans le cadre de la fouille de données complexes (données mixtes, prise en compte de connaissance experte, recherche de relations complexes).

3 Comité de programme

- Hanane Azzag (LIPN, Université Paris 13)
- Boutheina Ben Yaghlane (LARODEC, IHEC Carthage)
- Frédéric Blanchard (CreSTIC, Université de Reims Champagne-Ardenne)
- Omar Boussaïd (ERIC, Université de Lyon 2)
- Martine Cadot (LORIA, Nancy)
- Faicel Chamroukhi, (Southern-University of Toulon-Var, France)
- Guillaume Cleuziou (LIFO, Université d'Orléans)
- Laurent D'Orazio (LIMOS, Université de Clermont-Ferrand)
- Cyril De Runz (CReSTIC, Université de Reims Champagne-Ardenne)
- Thomas Delavallade (Société Thalès, CLEAR)
- Mounir Dhibi (ISSAT, Gafsa)
- Gaël Harry Dias (University of Beira Interior, Portugal)
- Zied Elouedi (ISG, Université de Tunis)
- Matthieu Exbrayat (LIFO, Université d'Orléans)
- Rim Faiz (IHEC, Université du 7 Novembre de Carthage)
- Sami Faiz (INSAT, Université du 7 Novembre de Carthage)
- Germain Forestier (MIPS, Université Haute Alsace)
- Pierre Gançarski (LSIIT-AFD, Université de Strasbourg)
- Lamia Hadrich (Laboratoire MIRACL, Université de Sfax)
- Nicolas Labroche (LIP6, Université Pierre et Marie Curie, CLEAR)
- Anne Laurent (LIRMM, Université de Montepellier 2)
- Mustapha Lebbah (LIPN, Université Paris 13)
- Eric Lefèvre (LGI2A, Université d'Artois)
- Arnaud Martin (IRISA, Université Rennes 1)
- Florent Masséglia (AxIS-Inria Sophia Antipolis)
- Frédéric Moal (LIFO, Université d'Orléans)
- Christophe Osswald (ENSIETA, Brest)
- Sébastien Régis (LAMIA, Université des Antilles et de la Guyane)
- Adrien Revault d'Allonnes (LIP6, Université Pierre et Marie Curie, CLEAR)
- A. Malek Toumi (ENSIETA, Brest)
- Cédric Wemmert (LSIIT-AFD, Univesité de Strasbourg)

4 Remerciements

Nous tenons à remercier les auteurs pour la qualité de leurs contributions, les membres du comité de programme et plus généralement tous les relecteurs de cet atelier pour le travail accompli et pour la qualité de leurs prestations.

Nous remercions également les responsables des ateliers pour EGC 2012.

Enfin nous remercions vivement les présidents : Yves Lechevallier, président du comité de programme, Guy Melançon et Bruno Pinaud co-présidents du comité d'organisation d'EGC 2012.

5 Programme

8h45 Accueil

9h00	Ouverture de l'atelier
9h15	Confidentialité et disponibilté des données entreposées dans les nuages *Kawthar Karkouda, Nouria Harbi et Jérôme Darmont*
9h45	Application de K-Means à la définition du nombre de VM optimal dans un Cloud *Khaled Tannir, Hubert Kadima et Maria Malek*

10h15-10h45 Pause

10h45	Licorn* : construction de réseaux de régulation chez l'homme *Ines Chebil, Celine Rouveirol, Christophe Rodrigues, Mohamed Elati et Remy Nicolle*
11h15	Apport des relations spatiales dans l'extraction automatique d'informations à partir d image *Bruno Belarte et Cédric Wemmert*
11h45	Vers davantage de flexibilité et d'expressivité dans les hiérarchies contextuelles des entrepôts de données *Yoann Pitarch, Cécile Favre, Anne Laurent et Pascal Poncelet*

12h15-14h Pause déjeuner

14h	Weighted Hierarchical Mixed Topological Map : une méthode de classification hiérarchique à deux niveaux basée sur les cartes topologiques mixtes pondérées *Mory Ouattara, Ndèye Niang, Sylvie Thiria et Fouad Badran*
14h30	Classification croisée par approche évolutionnaire itérative *Lydia Boudjeloud-Assala et Alexandre Blansché*
15h	Discussion sur l'atelier
15h30	Clôture

Summary

The workshop on mining complex is done at the incitement of the work group EGC "Complex data mining". Each year the organizers propose a topic that interests the researchers and companies.

Last year the workshop focused with success on the complexity associated with multiple data. Indeed we decided to keep this research field as one of the priority for the present edition of the workshop. In addition two other fields have been focused this year : the methods and advances in clustering and data mining in the cloud.

Confidentialité et disponibilté des données entreposées dans les nuages

Kawthar Karkouda*, Nouria Harbi**
Jérôme Darmont ** Gérald Gavin ***

Laboratoire Eric, Université Lumière Lyon 2, 5 avenue Mendès France, Bât K, 69677 Bron Cedex
* kawthar.karkouda@gmail.com
** nouria.harbi,jerome.darmont@univ-lyon2.fr
*** gerald.gavin@univ-lyon1.fr

Résumé. Avec l'avènement de l'informatique dans les nuages (Cloud Computing) comme un nouveau modèle de déploiement des systèmes informatiques, les entrepôts de données profitent de ce nouveau paradigme. Dans ce contexte, il devient nécessaire de bien protéger ces entrepôts de données des différents risques et dangers qui sont nés avec l'informatique dans les nuages. En conséquence nous proposons dans ce travail une façon de limiter ces risques à travers l'algorithme le partage de clés secret de Shamir et nous mettons cette contribution en pratique (Karkouda 2011).

1 Introduction

L'informatique décisionnelle représente de nos jours un facteur primordial pour les grandes entreprises qui se trouvent face à un grand volume de données. En effet, ce grand volume des données accumulées au cours du temps est considéré comme une source riche pour les décideurs puisqu'il permet à ces derniers d'avoir une vue d'ensemble sur les différentes activités de l'entreprise et les aide à prendre des décisions. D'autre part, avec le développement du concept de l'informatique dans les nuages et les différents avantages qu'elle offre en termes de puissance de calcul, de temps de réponse et de réduction des coûts, les entreprises peuvent bénéficier pleinement de service qui permet de satisfaire leurs besoins à moindres coûts. En effet, la mise en œuvre d'un entrepôt de données dans les nuages représente pour chaque entreprise une bonne solution vue son efficacité et sa rentabilité. Toutefois, comme chaque avancée technologique, l'informatique dans les nuages apporte aussi son lot de risques notamment en terme de sécurité qu'il faut prendre en compte pour pouvoir bénéficier de tous les avantages apportés par cette solution.
D'après Mansfield-Devine, les systèmes traditionnels sont protégés par des firewalls et des passerelles d'où les cybercriminels doivent collecter des renseignements intensifs pour savoir qu'ils existent. Alors que dans le cloud computing les systèmes sont très visibles et sont conçus pour être accessibles de n'importe où. En plus, les applications avec ce type de dispositif sont accessibles à travers un navigateur ; il s'agit d'une interface dont les faiblesses sont bien connues (Steve 2008). Les risques de l'informatique dans les nuages s'accroissent avec

l'utilisation de la virtualisation, qui est l'une des techniques de base dans ce type de dispositif (Mladen et Vouk 2008). En effet, Zhou et son équipe ont montré qu'il existe une faille dans l'hyperviseur XEN puisqu'il permet aux machines virtuelles de consommer le temps CPU pour d'autres utilisateurs et permet le vol de service (Fangfei et al. 2011). Wei et al. ont aussi abordé les problèmes de la gestion de la sécurité des images de la machine virtuelle (Jinpeng et al. 2009).
A vrai dire, les dangers de l'informatique dans les nuages ne sont pas limités à ce stade puisqu'il faut s'assurer aussi que les services soient disponibles à tout moment ce qui n'est pas toujours le cas. Par exemple, en 2009 a eu lieu une coupure du courant dans le nuage d'Amazon dans les locaux abritant ses serveurs en Virginie (1). Une telle panne peut provoquer une grande perte pour les entreprises puisque l'activité de l'entreprise qui héberge son infrastructure est arrêtée. Plusieurs travaux ont noté que les entreprises sont réticentes à l'informatique dans les nuages parce qu'elle impose l'externalisation de leurs données. En fait, le problème qui revient toujours est celui de ne pas vouloir laisser leurs données entre les mains de la concurrence, chose plus difficile à contrôler lorsque les données sont confiées à un prestataire externe dont la localisation physique est souvent inconnue. Il ne faut donc pas penser à une solution utilisant l'informatique dans les nuages et particulièrement les entrepôts de données sans répondre aux questions suivantes : Est-il raisonnable de confier des données importantes et sensibles à un fournisseur de cloud computing ? Comment peut-on s'assurer que le fournisseur de cloud computing ne disparaisse pas un jour ? Nos données seront-elles effacées lorsqu'on veut changer de fournisseur ? Ya-t-il un risque de perte de données stockées dans les nuages ? Le transfert des données vers les nuages est-il sécurisé ?...
Dans cet article nous commençons par présenter un état de l'art sur la sécurité de l'informatique dans les nuages et plus particulièrement celle des entrepôts de données. Il s'agit d'une présentation synthétique et d'un examen critique des principaux travaux. Ensuite, nous décrirons la solution proposée qui permet de résoudre quelques problèmes liés à la sécurité des données, puis nous présenterons la concrétisation de cette solution à travers l'implémentation d'un prototype et nous terminerons par une conclusion et les perspectives.

2 Etat de l'art

L'informatique dans les nuages est un nouveau modèle de prestation de service informatique utilisant de nombreuses technologies existantes. Mais, comme toute nouvelle technologie elle a besoin de nombreuses améliorations et de la mise en place de normes précises (Sean et Kevin 2011) pour éviter les risques. La sécurité est souvent considérée comme le frein principal à l'adoption des services du cloud computing (Grange 2010). C'est ainsi que de nombreux travaux ont été consacrés à la recherche de solutions pour remédier à ce problème. Nous essayerons dans cette partie de présenter les principaux travaux de recherche qui ont proposé des solutions pour assurer la sécurité de l'informatique dans les nuages.

2.1 Sécurité de l'informatique dans les nuages

On peut classer les aspects de la sécurité dans les nuages en trois catégories qui sont la sécurité des données, la sécurité logique et la sécurité physique (Grange 2010). Nous ne nous intéressons ici à quelques travaux consacrés uniquement aux deux premières catégories :

2.1.1 Sécurité des accès et du stockage de données dans les nuages

Jensen et al. ont énuméré les différentes techniques utilisées dans cloud computing pour sécuriser les accès et ils ont dégagé les lacunes de ces techniques pour mettre en œuvre leur solution qui est basée sur le protocole TLS et la cryptographie XML (Jorg et al. 2009). Cette solution vient répondre au problème de navigateur web qui présente des lacunes au niveau sécurité. L'idée proposée consiste à utiliser le protocole TLS et à adapter le navigateur en intégrant la cryptographie XML. Cependant Wang et al. ont proposé une solution qui se base sur le code erasurecorrecting afin de permettre la redondance et garantir la fiabilité des données. Ils ont utilisé le jeton homomorphique pour l'exactitude du stockage et pour localiser les erreurs. La solution proposée est capable de détecter la corruption des données lors du stockage, elle peut garantir la localisation des données erronées et identifier le serveur qui a un mauvais comportement (Cong et al. 2009). Dans le cloud computing, aucune hypothèse sur la robustesse d'un nœud ne peut être faite. Divers facteurs imprévus peuvent tous entrainer une inaccessibilité temporaire de certains nœuds ou de l'inaccessibilité définitive des données. Dans un tel cas, les moyens traditionnels de protection des données sont souvent impuissants. Danwei Chen et Yanjun He ont proposé un algorithme de sécurité qui assure la restauration des données en cas d'échec de certains serveurs. Il s'agit d'un algorithme de séparation de données (Danwei et Yanjun 2010). Cet algorithme n'est qu'une extension du théorème fondamental de K équations en algèbre, l'algorithme du partage de la clé secrète de Shamir qui est un algorithme de cryptographie basé sur le partage du secret (2), l'algorithme de stockage de données en ligne de Abhishek (Parakh et Kak 2009) et la théorie du nombre. L'idée est de partager la donnée d en k parties d= d_1,d_2,d_3,..d_k. Ce partage est fait à l'aide de l'algorithme de séparation de données pour les stocker ultérieurement sur des serveurs choisis aléatoirement noté S= s_1, s_2,s_3,..s_m avec m>k. Le processus de stockage de données dans les nuages se fait donc sur deux étapes, la première consiste à diviser et stocker les données sur un serveur choisi arbitrairement et la deuxième consiste à pouvoir restituer ces données. A travers ces processus, les données sont prêtes à être transférées, stockées, traitées, en toute sécurité puisqu'elles sont cryptées. Les chercheurs ont déduit que la complexité temporelle de l'algorithme est la même pour générer k blocs de données et pour la restauration des données. Ils ont prouvé que même si un attaquant envahit un nœud de stockage, vole un bloc de données et essaie de rétablir la série de données, la complexité temporelle nécessaire pour faire les traitements ne peut pas être supportée par les environnements informatiques actuels. Parmi les autres avantages qui distinguent cette proposition on trouve la capacité de restaurer les données même si un ou plusieurs nœuds de stockage ne sont pas disponibles ce qui ne peut pas être le cas avec une solution traditionnelle de cryptographie.

2.1.2 Sécurité logique de l'informatique dans les nuages

Dans les nuages IAAS (Infrastructure as a service), les utilisateurs ont accès aux machines virtuels VM (3) sur lesquels ils peuvent installer et exécuter leurs logiciels. Ces machines virtuelles sont créées et gérées par un moniteur de machine virtuel VMM qui est une couche logicielle entre la machine physique et le système d'exploitation. Le VMM contrôle les ressources de la machine physique et crée plusieurs machines virtuelles qui partagent ces ressources. Les machines virtuelles ont des systèmes d'exploitation indépendants exécutant des applications indépendantes et sont isolées les unes des autres par le VMM. Ce type de dispositif a provoqué

beaucoup de problèmes de vulnérabilité de la machine virtuelle ce qui a poussé les auteurs à travailler dans ce domaine pour trouver des solutions efficaces. Zhou et al. ont proposé une solution pour éliminer la vulnérabilité de la machine virtuelle. La découverte des limites de l'hyperviseur XEN utilisé par AMAZON était leur point de départ. Ils ont proposé quatre approches pour améliorer la performance de cet hyperviseur qui sont basées sur la loi de Poisson, la loi de Bernoulli, la loi uniforme et enfin la loi exacte. Après une comparaison entre les quatre nouveaux modèles, ils ont déduit que la stratégie basée sur la loi de Poisson est la meilleure dans la pratique pour empêcher le vol de cycle (Fangfei et al. 2011). Wei et al. ont proposé un système de gestion des images de la machine virtuelle qui contrôle l'accès et la provenance de ces images à travers des filtres et des scanners qui permettent de détecter et réparer les violations en utilisant les techniques de fouille de données, ce système s'appelle Mirage (Jinpeng et al. 2009). S. Berger et al. ont aussi développé une technologie qui répond aux problèmes rencontrés par la machine virtuelle. Cette technologie est appelée Trusted Virtual Data Center (TVDc), elle assure que les charges de travail ne peuvent être facturées qu'aux clients qui ont bénéficié du service. Elle assure aussi dans le cas de certains programmes malveillants comme les virus qu'ils ne peuvent pas se propager à d'autres nœuds et elle permet également de prévenir les problèmes de mauvaise configuration. La TVDc utilise la politique d'isolement qui se base sur la séparation des ressources matérielles utilisées par les clients. Elle gère le centre de données, l'accès aux machines virtuelles et le passage d'une machine virtuelle à une autre (Fei et al. 2011).

2.2 Discussion

Nous avons présenté dans la partie précédente quelques travaux qui proposent de résoudre les problèmes liés à la sécurité dans le cloud computing. Quelques approches semblent être pertinentes et assurent un niveau de sécurité acceptable mais qui reste insuffisant. En plus, ces travaux ont mis en évidence de nouveaux problèmes : Danvei Chen et Yanjun He ont relevé la redondance des données. Mais ça n'empêche pas que l'idée proposée par ces chercheurs est très fiable pour sécuriser le transfert des données à travers le réseau et élimine le problème de non disponibilité du service en cas de panne de l'un des serveurs. Jensen et al ont proposé d'utiliser le protocole TLS et adapter le navigateur en intégrant la cryptographie XML. Une telle proposition n'est pas suffisante pour assurer la sécurité du transfert sur les réseaux puisqu'elle est basée sur une technologie dont ses faiblesses sont bien connues. En ce qui concerne l'idée proposée par Wang et ses collaborateurs, elle est basée sur le chiffrement homomorphique qui est la réponse à la question de confidentialité des données lors de transfert et lors de traitement dans le cloud. Néanmoins, le laboratoire de recherche en cryptographie dans le cloud de Microsoft a annoncé que cette nouvelle façon de chiffrer les données n'en est encore qu'à ses débuts et qu'ils sont loin de pouvoir exécuter sur les machines virtuelles des données cryptées avec le chiffrement homomorphique (4).
Généralement les solutions existantes pour sécuriser le transfert et les traitements des données sont basées sur la cryptographie des données qui n'est pas toujours une solution complète pour protéger les données, en plus le mécanisme de cryptage et de décryptage des données peut être intensif sur les processeurs ce qui engendre un gaspillage des ressources une chose que les fournisseurs des nuages ne veulent pas (Grange 2010).

3 Sécurisation des données par le secret sharing

3.1 Motivation

L'utilisation de l'informatique dans les nuages est basée sur la confiance qu'on peut accorder aux fournisseurs de ce type de service. Une telle situation est difficile à renforcer avec l'architecture traditionnelle du cloud qui repose sur un seul fournisseur. Cette dépendance menace la confidentialité des données des clients puisque ces dernières sont hébergées chez un seul prestataire externe qui risque de les exploiter. Pour cela nous proposons une autre façon d'hébergement des entrepôts de données afin d'éliminer la dépendance par rapport à un seul fournisseur en utilisant plusieurs fournisseurs. Cette solution rend les données hébergées chez chaque fournisseur non significatives et donc non exploitables.

3.2 Proposition

Notre proposition consiste à partager chaque donnée stockée dans l'entrepôt sur plusieurs fournisseurs des nuages à travers l'algorithme de secret sharing (Shamir 1979). Dans ce chapitre nous présentons en détail la solution pour sécuriser le stockage et l'exploitation d'un entrepôt de données dans les nuages, cette dernière est inspirée de l'idée proposée par Danwei Chen et Yanjun He dans leur article intitulé 'A Study on Secure Data Storage Strategy in Cloud Computing' (Danwei et Yanjun 2010). Il s'agit d'une solution basée sur l'algorithme de secret sharing qui partage les données en n-uplet et les stocke chez un seul fournisseur. Dans la solution que nous proposons, notre apport consiste à stoker le n-uplet chez plusieurs fournisseurs. Cette façon de répartir les données permet d'une part de stocker au niveau de chaque fournisseur une partie de l'information, celles-ci sont alors non compréhensibles et non exploitables par un utilisateur malveillant en cas d'intrusion et d'autre part de ne pas dépendre d'un seul fournisseur, ce qui minimise le risque de non disponibilité des données. Les étapes de la démarche sont :
-Chaque fournisseur des nuages possède une copie de l'architecture de l'entrepôt du client.
-Chaque donnée de l'entreprise est partagée et stockée chez les différents fournisseurs de manière à la rendre inexploitable par chaque fournisseur car non significative.
-Le nombre de fragments dépend du nombre de fournisseurs choisis par le client.
-Pour la restauration d'une donnée, le client doit récupérer les fragments stockés chez les différents fournisseurs pour reconstituer la donnée initiale (figure 1).

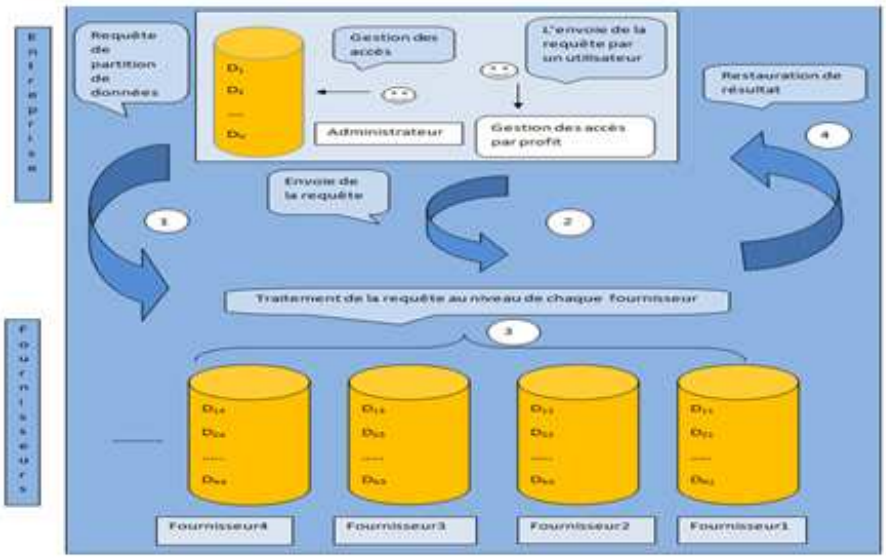


FIG. 1 – *Scénario d'un entrepôt de données partagé dans les nuages*

3.3 Le niveau de sécurité attendu

Notre solution assure trois niveaux de sécurité : - Capacité de restauration des données en cas de non disponibilité du service ou de la disparition d'un fournisseur puisque l'idée est basée sur l'algorithme de secret sharing qui est capable de reconstituer la donnée initiale à partir d'un nombre prédéfini de fragments qui peut être inférieur au nombre de fragments stockés chez les différents fournisseurs.
- Sécurité des transactions entre le client et les fournisseurs puisque les données qui transitent sur le réseau sont partielles et inexploitables.
- Sécurité des données stockées chez les différents fournisseurs puisque chacun d'eux n'a qu'une partie d'une donnée non significative.
Pour atteindre le niveau de sécurité attendu il faut respecter les deux règles énoncées dans les deux sous sections suivantes.

3.3.1 Choix des coefficients

Le choix des coefficients a une grande influence sur la sécurité des données. On suppose par exemple que les a_i 1<=i<=k-1 coefficients choisis aux hasards sont tous égaux à zéro, le polynôme f(x) =a_0+a_1x+a_2x_2+a_3x_3+...+$a_{K-1}x_{k-1}$ devient f(x)=a_0 ce qui n'assure pas la sécurité du transfert. Par conséquent l'algorithme nécessite que les coefficients ne doivent pas être nuls en même temps.

3.3.2 Complexité temporelle

On suppose qu'un attaquant envahit un nœud de stockage, et vole un bloc de donnée r_i et veut restaurer la donnée d'origine D avec les méthodes agressives basées sur r_i et décode les coefficients. Il a besoin de [P_{K-1}/(K-1) !] complexité temporelle, avec p» K» 2. Un tel rapport ne peut pas être calculé avec les capacités des traitements actuels des ordinateurs. En se référant à cette théorie, nous pouvons remarquer que si on augmente le K on peut augmenter la complexité temporelle ce qui signifie la diminution des risques.
D'autre part le nombre K est un facteur qui dépend du nombre de pièces qu'on espère obtenir après la partition de la donnée d'origine D puisqu'il ne peut pas être plus grand que ce nombre. C'est ainsi qu'afin de diminuer les risques il faut augmenter le facteur K par conséquent augmenter le nombre de partitions N. Ce qui signifie qu'il faut augmenter le nombre de fournisseurs de l'informatique dans les nuages pour diminuer les risques de reconstitution de l'information.
La figure 2 présente l'augmentation de la complexité temporelle en fonction d'augmentation de facteur K qui est le nombre des fournisseurs dont on a besoin pour reconstituer l'information. Dans cet exemple le nombre entier p=9

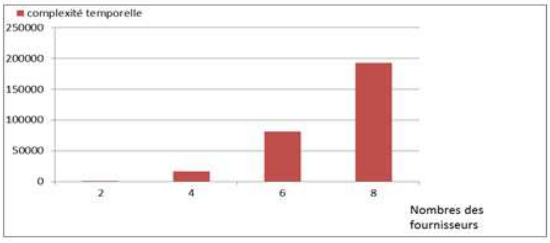


FIG. 2 – *L'augmentation de la complexité temporelle en fonction du nombre des Fournisseurs*

3.4 Résultats théoriques : Rapport coût/risque

L'avantage le plus important de l'informatique dans le nuage est que l'entreprise paye ce qu'elle consomme " pay as you go ", c'est-à-dire, le coût de consommation est en fonction de l'espace mémoire consommé, des accès sur le réseau et du temps de traitement des requêtes. D'où on peut résumer la fonction de coût par l'équation suivante :
D *C_S +T*C_T +T_{rq} *C_{Tr}= C_{tot} ; ou
D : les données stockées chez le fournisseur de nuage
C_s : Coût de stockage qui dépend de chaque fournisseur
T : Temps de traitement des requêtes
C_T : Coût de traitements de requêtes qui dépend de chaque fournisseur
T_{rq} : Taille de la requête et le résultat
C_{Tr} : Coût de transfert sur le réseau qui dépend de chaque fournisseur
Revenons maintenant sur la fonction de coût qui concerne notre proposition, elle change en fonction de nombre des fournisseurs avec lesquels l'entreprise s'est engagée d'où la fonction de coût se transforme en :
$\sum_{i=1}^{n}$ (D$_i$ *C_{Si} +T_i*C_{Ti} +T_{rqi} *C_{Tri})= C_{tot} ; où n est le nombre de fournisseurs des nuages

D'après cette formule on peut remarquer que le coût à payer à un seul fournisseur par l'entreprise avec notre proposition est n fois plus grand qu'une solution traditionnelle, mais elle assure aussi n fois plus la sécurité et réduit les risques. Le coût du risque correspond à l'impact que la perte de tout ou partie de l'entrepôt de données aura sur l'activité de l'entreprise, celle ci se calcule en temps de travail pour reconstituer les données. Plus précisément, notre proposition est basée sur l'algorithme de secret sharing qui a besoin d'un nombre k de partie fixé dès le début pour la reconstitution de donnée. Ce nombre K représente dans notre proposition le nombre de fournisseurs du cloud qui doivent être disponibles pour la reconstitution des données. Néanmoins ce nombre de fournisseurs peut dépasser le nombre K nécessaire fixé dès le départ pour trouver la marge de sécurité en cas de non disponibilité de service de quelques fournisseurs par exemple et atteindre un nombre n qui dépend du choix de l'entreprise.
Pour diminuer le risque de non disponibilité, il faut alors augmenter le facteur n qui est le nombre de fournisseurs des nuages utilisés par l'entreprise. Cette dépendance influe sur le coût d'utilisation de l'informatique dans les nuages C_{tot} qui va augmenter proportionnellement au nombre de fournisseurs des nuages. Le rapport entre le coût et le risque reste alors un compromis qui dépend du niveau de sécurité que l'entreprise veut assurer : plus on diminue le risque de coalition par l'augmentation du nombre des fournisseurs plus on augmente le coût d'utilisation de l'informatique dans les nuages.
D'autre part la gestion des risques est devenue une partie intégrée dans les activités des entreprises. Elle consiste à gérer efficacement les risques auxquels le système informatique de l'entreprise est exposé et à préparer les contre-attaques pour dépasser ces risques. Une telle politique nécessite un budget financier important pour garantir la fiabilité et la continuité des services de l'entreprise puisqu'un accident peut provoquer des dégâts financiers importants (IBM 2008). En plus, en cas de faille de système informatique de l'entreprise elle risque de perdre ces clients et sa réputation sur le marché. Réduire ce type des risques est devenu une priorité stratégique pour les entreprises dans un contexte concurrentiel rude. En réalité la définition des risques associés au cloud computing est beaucoup plus large elle englobe les incertitudes, les dangers et les pertes (5) ce qui nécessite une gestion des risques plus efficace et un budget financier plus grand. Cette gestion des risques est assurée dans les entreprises par des moyens spécifiques comme la méthode Mehari et la méthode Marion. Ces méthodes aboutissent généralement à des résultats efficaces, qui consistent à estimer le coût des risques en cas de faille de système informatique. A travers ces deux facteurs qui sont le coût total de l'informatique dans les nuages calculé Ctot et le coût du risques estimé par les méthodes de gestion de risque, l'entreprise peut savoir à partir de quel moment une solution cloud computing devient coûteuse en tenant compte de la contrainte suivante : C_{tot} /coût des risques $<$ 1. Sachant que le coût des risques est une constante, l'entreprise peut diminuer ce rapport en diminuant le nombre de fournisseurs.

4 Mise en pratique de notre proposition

Dans ce chapitre nous allons présenter un prototype pour la mise en pratique de notre proposition. Ce prototype est constitué de trois fournisseurs de cloud computing chez lesquels nous hébergeons notre entrepôt de données contenant les données sur les ventes de produits dans plusieurs magasins, il s'agit d'une simulation d'un grand volume de données. Nous avons

partagé les données en utilisant l'algorithme de secret sharing. La suite du chapitre contient les différentes fonctionnalités qui sont assurées par notre prototype.

4.1 Manipulation des données

4.1.1 Partition des données

Le partage des données se fait au niveau de l'entreprise à travers la première étape de l'algorithme de secret sharing. Cette première étape assure le partage de données en fonction de nombre des fournisseurs et inclut aussi l'envoie de chaque partie arbitrairement à un fournisseur (figure 3).

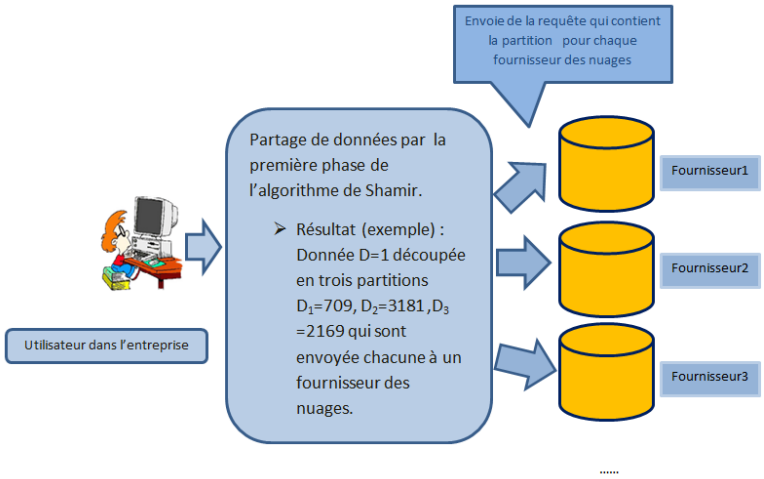


FIG. 3 – *La phase de partition de données*

4.1.2 Restitution de données

En fonction des différentes parties constituant l'information, la deuxième partie de l'algorithme de secret sharing est capable de restituer la donnée d'origine. Cette étape est faite au niveau de l'entreprise (figure 4).

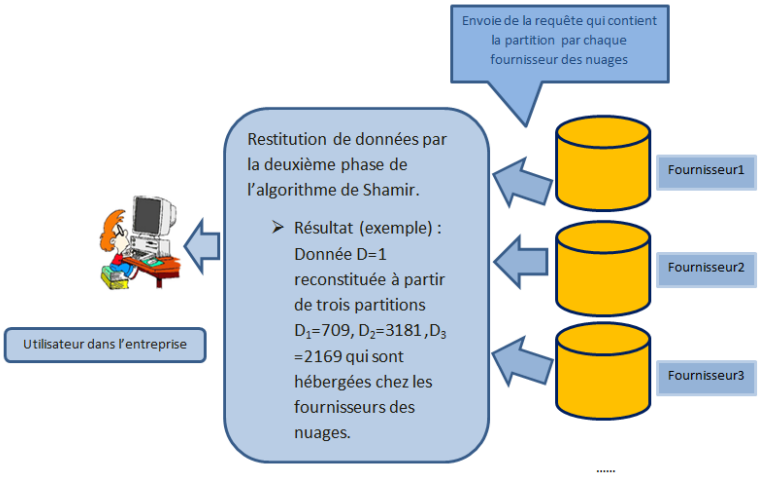


FIG. 4 – *La phase de restitution de données*

4.2 Autres opérateurs pour les analyses OLAP

On peut facilement faire des additions sur les données partagées, mais les analyses OLAP nécessitent d'autre types d'agrégation : moyenne, écart-type, min, max... qui nécessitent des adaptations pour qu'ils puissent fonctionner avec notre modèle. Dans notre prototype nous avons implémenté la variance et le maximum. Les deux sous sections qui suivent présentent le principe que nous avons utilisé pour l'implémenter.

4.2.1 La variance

Afin d'intégrer la variance dans notre travail nous avons proposé d'ajouter une colonne où on fait la multiplication de chaque valeur de xi au niveau de l'entreprise puis on partage la colonne au niveau des différents fournisseurs pour qu'on puisse se servir après pour la restitution de données. Ceci nous oblige à ajouter une autre colonne pour le stockage des xi aux carrés. La figure 5 explique le principe adopté pour assurer le calcul de la variance sur les données. L'exemple présenter consiste à calculer la variance sur les chiffres d'affaires par magasin, par département et par mois.

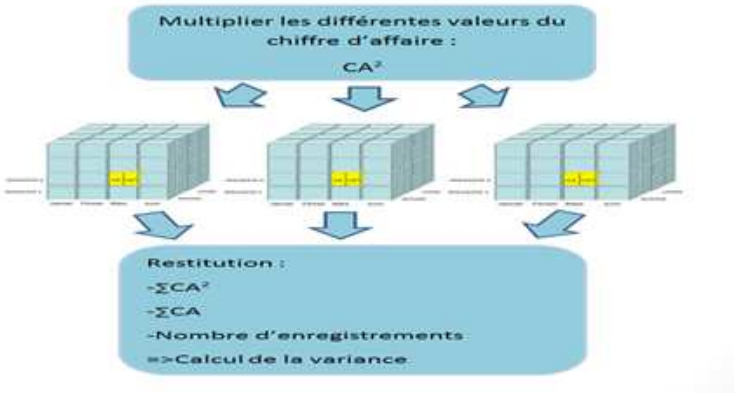


FIG. 5 – *Le processus de calcul de la variance*

4.2.2 Le maximum

L'idée que nous proposons est de calculer les valeurs des colonnes dont on veut estimer son maximum par la formule suivante : chaque enregistrement exposant un nombre très grand dans une première étape. Ensuite, on partage les nouvelles valeurs avec les différents fournisseurs. Une fois que nous voulons savoir le maximum des valeurs prix par la colonne on applique la fonction d'agrégation Max sur chaque entrepôt. En effet les résultats obtenues n'ont aucune signification, pour cela il faut appliquer la méthode de restitution de donnée proposée par Shamir et ensuite multiplier le résultat par (1/le nombre que nous avons utilisé dans la première étape). Pour plus d'explication la figure 6 présente le processus de calcul du maximum sur le chiffre d'affaire.

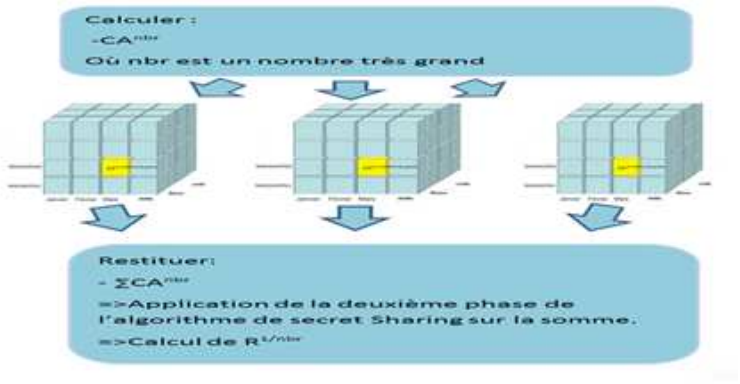


FIG. 6 – *Le processus de calcul de maximum*

4.3 Discussion

Comme il est déjà annoncé l'implémentation de la variance et du maximum nécessite l'ajout d'une colonne pour chacun. Alors que l'ajout des colonnes entraîne une nouvelle augmentation du volume de données, et donc du coût de la solution. Mais à priori, ça ne s'applique qu'aux mesures de l'entrepôt, donc à un ensemble limité d'attributs. Les opérateurs d'agrégation Max, variance, count, moyenne qui sont nécessaires pour les analyses OLAP sont mis en œuvre dans ce prototype.

5 Conclusion et perspectives

Cette étude nous a permis de constater que les problèmes majeurs d'hébergement d'un entrepôt de données dans les nuages est le manque de confiance accordée aux fournisseurs du cloud computing, de disponibilité de service et de sécurité de transfert des données vers le cloud computing. Pour diminuer ces risques, nous avons proposé une solution basée sur l'algorithme de secret Sharing, qui consiste à partager l'information chez plusieurs fournisseurs au lieu d'un seul. L'objectif de cette solution est de diminuer la vulnérabilité des données transmises. Chaque partie est stockée chez un des fournisseurs, ce qui les rend d'une part non significatives pour chaque fournisseur et d'autre part résoud le problème de non disponibilité de service en cas de panne chez un des fournisseurs. Nous avons créé un prototype qui non seulement assure les traitements nécessaires pour le partage et la restitution des données mais aussi intègre quelques opérateurs d'agrégation (Max, Count, variance, moyenne) pour l'analyse OLAP.
Les résultats obtenus dans le cadre de ce travail montrent que les perspectives de recherche sur le sujet sont nombreuses :
- Etudier la possibilité d'intégrer la cryptographie traditionnelle pour sécuriser le transfert des requêtes sur les réseaux.
- Implémenter d'autres opérateur d'agrégation :exemple le Min qui est nécessaires pour les analyses OLAP.

- Appliquer une méthode de gestion des risques pour déterminer le coût de risque réel.
- Renforcer la sécurité des accès en intégrant dans la solution la gestion des accès à l'entrepôt dans le cloud computing en fonction des profits utilisateurs de l'entreprise.

6 Bibliography

Cong Wang, KuiRen ,Qian Wang ,Wenjing Lou (2009). Ensuring Data Storage Security in Cloud Computing, In the 17th IEEE International Workshop on Quality of Service (IWQoS'09), Charleston, South Carolina, July 13-15.

Danwei Chen, Yanjun He (September 2010). A Study on Secure Data Storage Strategy in Cloud Computing, Journal of Convergence Information Technology, Volume 5, Number 7.

Fangfei Zhou, Manish Goel, Peter Desnoyers, Ravi Sundaram (mars 2011). Scheduler Vulnerabilities and Attacks in Cloud Computing, College of Computer and Information Science Northeastern University, Boston, USA.

Fei Hu, MeikangQiu, Jiayin Li, Traavis Grant, Draw Tylor, Seth McCaleb, Lee Bulter and Richard Hamner (2011). A Review on cloud computing :Design challenges in Architecture and Security . Journal of Computing and Information Technology - CIT 19,1, 25-55, p. 17

Grange Philippe (quatrième trimestre 2010). Livre blanc sécurité du Cloud computing, analyse des risque, réponses et bonnes pratiques, Syntec numérique, p7.

IBM (Septembre 2008) :Méthodologie de gestion de risque informatique pour les directeurs des systèmes d'information :un levier exceptionnel de création de valeur et de croissance,p2,p11.

Jinpeng Wei, Xiaolan Zhang, Glenn Ammons (2009). Managing Security of Virtual Machine Images in a Cloud Environment, Proceedings of the 2009 ACM workshop on Cloud computing security, New York.

Jorg Schwenk, Luigi Lo Lacono, Meiko Jensen, Nils Gruschka (2009). On Technical Security Issues in Cloud Computing, IEEE International Conference on Cloud Computing.

Karkouda K (septembre2011).Sécurité des entrepôts de données dans les nuages , mémoire de stage du master 2 ECD (extraction des connaissances à partir des données), encadré par Mme Nouria Harbi, Mr Jérome Darmont et Mr Gerald Gavin, laboratoire ERIC, université Lyon 2.

Mladen A. Vouk (September, 2008). Cloud Computing - Issues,Research and Implementations.Journal of Computing and Information Technology- CIT 16, 2008.Received : June, 2008 Accepted,p237

A.Parakh , S. Kak (2009). Online data storage using implicit security, Information Sciences, vol.179, no 3323-3331.

Sean Carlin et Kevin Curran (janvier-mars 2011). Cloud computing Security, International journal of Ambient Computing and Intelligence, vol.3, issue 9, p. 14.

A.Shamir (1979). How to Share a Secret, Communications of the ACM, vol.22, no.11, pp.612-613.

Steve Mansfield-Devine (December 2008). Danger in the clouds, Network Security, Volume 2008, Issue 12, Pages 9-11.

[1]www.presence-pc.com/actualite/Amazon-EC2-37511/

[2]http ://fr.wikipedia.org/wiki/partage de clé secrète de Shamir.

[3]Virtual Computing Lab. http ://vcl.ncsu.edu/.

[4]http ://blogs.orange-business.com/securite/2011/08/chiffrement-homomorphique-une-bete-concue-pour-le-cloud.html

[5]http ://www.cairn.info/resume.php ?ID-ARTICLE=RIGES-275-0063

Summary

With the rise of cloud computing as a new model for deploying computer systems, data warehouses benefit from this new paradigm. In this context, it becomes necessary to adequately protect these data warehouses of various risks and dangers that are born with cloud computing .Therefore, we propose in this work a way to limit these risks through the algorithm Shamir's secret Sharing and we make this contribution in practice .

Application de K-Means à la définition du nombre de VM optimal dans un Cloud

Khaled Tannir; Hubert. Kadima ; Maria Malek

Laboratoire LARIS/EISTI Ave du Parc 95490 Cergy-Pontoise – France
contact@khaledtannir.net, hubert.kadima@eisti.fr, maria.malek@eisti.fr

Résumé

Ce papier présente les premiers éléments de définition d'un algorithme permettant de déterminer le nombre optimal de machines virtuelles (*VM – Virtual Machines*) lors de l'exécution des applications de fouille de données dans un environnement Cloud. L'efficacité de traitement des problèmes de fouille de données requiert d'obtenir au préalable un *partitionnement intelligent de données* par *clustering* de manière à effectuer le plus indépendamment que possible les traitements des fragments de données à cohérence sémantique forte.

Nous pensons que l'exécution sur les données distribuées dans le Cloud d'une variante parallèle de l'algorithme de *clustering h-means* adaptée en phase de présélection du processus PMML [18] pour (*Predictive Model Markup Language*) permettrait d'assurer un partitionnement optimal des données et de déterminer un nombre de VM optimal avant l'exécution de l'application.

Mots-clés : *Cloud computing, h-means, classification, parallélisme dans des grilles, partitionnement de données, fouille de données.*

1. Introduction

Ce papier présente les premiers éléments de définition d'un algorithme permettant de déterminer le nombre optimal de machines virtuelles pour une exécution efficace des applications de fouille de données dans un environnement Cloud. Un environnement *Data Intensive Computing*, tel que celui-ci, devrait permettre d'exécuter efficacement des applications qui permettent la production, manipulation ou analyse des gros volumes de données – de quelques centaines de *megabytes* aux *petabytes* et au-delà [1]. A cause des coûts de communication élevés dans cet environnement, les traitements parallèles doivent être les plus autonomes que possibles et effectués avec le moins de synchronisations possibles. L'efficacité de traitement des problèmes de fouille de données requiert d'obtenir au préalable un *partitionnement intelligent de données* de manière à effectuer le plus indépendamment possible les traitements des fragments de données en tenant compte des caractéristiques spécifiques aux Clouds (*élasticité, monitoring* fin d'utilisation des ressources, *speed-up* …).

L'utilité du *partitionnement intelligent* obtenu à partir du *clustering* pour le problème des règles d'association a été déjà étudiée dans [2] par exemple. Nous pensons que l'exécution sur les données distribuées dans le Cloud d'une variante parallèle de l'algorithme de

clustering h-means en phase de présélection du processus PMML [18] (*Predictive Model Markup Language*) permettrait d'assurer un partitionnement optimal des données et de déterminer un nombre de VM optimal. Nous présentons dans ce papier notre approche qui consiste de partitionner les données en fonction de leurs similarités. Nous procédons par deux étapes : l'exécution de l'algorithme de partitionnement puis distribuer les données partitionnées sur le nombre de machines virtuelles trouvé.

Dans notre environnement d'exécution, les applications de fouille de données sont représentées sous forme d'un *workflow* conformément au processus PMML. Toute activité du *workflow* impliquant la manipulation de données génère des requêtes sur les données distribuées dans le Cloud. On trouve dans [3] une présentation de plusieurs techniques spécifiques utilisées par les systèmes de *workflows* pour exécuter les applications *workflows data intensive* en utilisant des ressources globalement distribuées. La plupart des systèmes utilisent une combinaison des techniques pour fournir les performances élevées et la haute disponibilité à faibles coûts. Une seule technique n'est pas suffisante pour minimiser les effets sur la performance et les coûts et augmenter l'efficacité des opérations de transferts de données.

Le partitionnement de données, le placement de données ou la réplication peuvent être réalisés avant ou durant l'exécution de *workflows* dans un environnement *Cloud Data Intensive* pour améliorer les performances d'exécution de l'application. Kosar et al. [4] ont défini un *scheduler* pour les activités de placement de données dans le Grid. Ils proposent d'en faire un axe majeur des travaux de recherche à effectuer dans ce domaine. Les activités de placement de données peuvent être intégrées ou découplées des activités de *scheduling des tasks.* La réplication de données dans des ressources distribuées est le mécanisme commun pour augmenter la disponibilité de données. La décision de placement et de réplication de données sont basées sur les objectifs à optimiser, la localité de référence de données et la minimisation de la distance de répartition entre les données et les traitements associés.

La réponse apportée par h-*means* pour répondre à ces derniers problèmes se traduit en termes de traitement des requêtes par similarité sur de gros volumes de données, notamment à travers, par exemple, les calculs des *k* plus proches voisins (*k*NN). Au processus de *clustering*, s'ajoutent les problèmes de parallélisme de traitements et de la distribution de données. On doit se donner les moyens d'étudier les performances de cette approche de partitionnement appliquée sur une grosse base de données pour obtenir et exploiter des classes et déterminer une taille et un nombre optimal de ces classes pour que les requêtes issues du *workflow* puissent être traitées en temps optimal et avec une haute précision. La mise en œuvre de h-*means parallèle* introduit des méthodes de traitement de requêtes parallèles sur une grille de machines pour réaliser l'allocation optimale des données grâce à la recherche efficace des *k*NN en parallèle. On vise l'obtention d'un nombre réduit de VM distribués à travers les Clouds permettant néanmoins des temps de recherche sous-linéaires et optimaux vis-à-vis des classes déterminées précédemment.

Dans un premier temps, une application structurée en *workflow* et utilisant les techniques de fouille de données est exécutée pour des besoins de tests dans un environnement Cloud hybride comportant l'infrastructure Cloud *OpenNebula* [5] interconnectée à *Amazon EC2* [6].

Les tests s'effectuent sur les instances *EC2 Amazon* en connexion avec un moteur d'orchestration de *workflow* implanté côté *OpenNebula*.

La suite de ce papier est structurée comme suit. Dans la section 2, nous présentons notre algorithme CloudMeans variante de h-*means parallèle* que nous utilisons. Dans la section 3, nous décrivons la configuration de l'environnement d'exécution de l'application de tests basée sur l'utilisation du moteur de *workflow* KNIME [17] et de l'infrastructure Cloud OpenNebula interconectée à Amazon via les services AWS (*Amazon Web Services*). A titre de conclusion, dans la section 4, nous présentons le plan des premières actions entreprises et les principaux axes de nos futurs travaux.

2. Présentation de points essentiels de l'algorithme

CloudMeans est une variante de *h-means* [9] lui même variante de l'algorithme *k-means* [8]. K-means est un des algorithmes de segmentation les plus connus. Il est souvent utilisé quand il y a un besoin d'extraire des regroupements (appelés encore catégories, groupes ou "clusters") à partir d'un ensemble de données d'une façon non supervisée. Chaque catégorie est représentée par son barycentre.

Entrées L'algorithme prend comme entrées un ensemble de n exemples, un entier k qui est le nombre de catégories à trouver, ainsi qu'une mesure de distance d() (comme la distance euclidienne) qui permet de calculer la distante entre deux exemples. Plus cette distance entre deux exemples est petite, plus les deux exemples sont similaires.

Déroulement L'algorithme assigne à chaque itération un exemple donné au barycentre le plus proche. Des mesures d'inerties inter-catégories et intra catégories sont calculées. Ces mesures d'inerties sont basées sur les densités des groupes. L'algorithme converge vers une situation de stabilisation de barycentres quand l'inertie inter-catégories est minimisée, autrement dit quand les catégories trouvées sont les plus denses possibles (ou les mieux séparés).

Sortie La sortie de l'algorithme consiste en une partition des n données en k groupes (ou catégories) chacun est représenté par son barycentre. Une variante très connue de k-means est l'algorithme h-means [9] dont le but est d'optimiser le temps de convergence vers les barycentres. La différence se fait au moment du calcul du barycentre qui se fait chaque fois qu'une catégorie est traitée complètement et non pas après le traitement de chaque exemple.

Plusieurs algorithmes de parallélisation de k-means ont été proposés [10,11,12,13]. Dans [10], une parallélisation de l'algorithme h-means est présentée. Nous présentons ici une parallélisation possible (sur plusieurs processeurs) de l'algorithme h-means. Cette parallélisation constitue le point de départ pour notre algorithme.

Version parallèle de l'algorithme h-means

Répartir en bloc de manière égale les exemples sur *p* processeurs

Pour chaque processeur k_p dans {0,1,..,*p*-1} (**en parallèle**)
Attribuer arbitrairement chaque exemple dans un processeur k_p à une catégorie

/* Calcul de affectations des exemples aux catégories en parallèle */

Tant que les barycentres ne sont pas stables
Pour tous les exemples Exemples(i) dans processeur k_p
Calculer la distance entre Exemple(i) et tous les barycentres
Trouver k' tel que k'ème barycentre soit le plus proche de Exemple(i)
Attribuer Exemple(i) au barycentre k'

/* Recalcule des barycentres des catégories en parallèle */

Calculer la somme partielle des exemples appartenant à la même catégorie.
Envoyer la somme partielle aux autres processeurs.

Recevoir les sommes partielles.

Pour chaque catégorie **faire**

Effectuer la somme des sommes partielles.
Diviser la somme totale par le nombre d'exemples de la catégorie courante.

Notre adaptation des ces algorithmes étend l'algorithme de parallélisation *h-means* en l'englobant dans d'autres taches plus complexes. Le déroulement de l'algorithme, illustré par la figure 1, peut être résumé ainsi : on commence par indiquer un nombre choisi aléatoirement (p) que nous introduisons comme paramètre en entrée à l'algorithme. Ce nombre servira comme base de répartition des blocs de données. Nous étudions actuellement l'impact des variations de ce nombre sur l'efficacité de notre algorithme et travaillons actuellement pour déterminer ce nombre de façon optimale. Une fois le nombre de catégories nous a été retourné par cette partie de l'algorithme nous l'utiliserons comme paramètre pour la création d'un nombre égale de machines virtuelles sur le cloud local. Les données actuellement partitionnées par catégorie seront redistribuées sur ces machines. Dans le cas où les ressources du cloud local sont insuffisantes pour absorber cette demande, nous faisons appel au cloud public afin qu'il nous fournisse les ressources manquantes.

Actuellement nous travaillons sur cette partie afin de définir de façon optimale la stratégie à adopter afin de maximiser l'efficacité des traitements tout en tenant compte du volume de données à traiter, du nombre de machines virtuelles à créer. Finalement, le traitement des données est lancé et le résultat est récupéré et mis à disposition de la prochaine étape du workflow global du traitement.

Les données internes nécessaires au fonctionnement de l'algorithme comme par exemple les tables de hachages communes sont stockées en local. Nous étudions également la possibilité d'utiliser une base NoSQL telle que Cassandra [21] pour améliorer la disponibilité et les performances de ces données. Une base de données NoSQL (*Not only SQL*) désigne une catégorie de base de données apparue en 2009 qui se différencie du modèle relationnel que l'on trouve dans des bases de données connues comme MySQL ou PostgreSQL. Ceci permet d'offrir une solution alternative aux

problèmes récurrents des bases de données relationnelles de perte de performance lorsque l'on doit traiter un très gros volume de données.

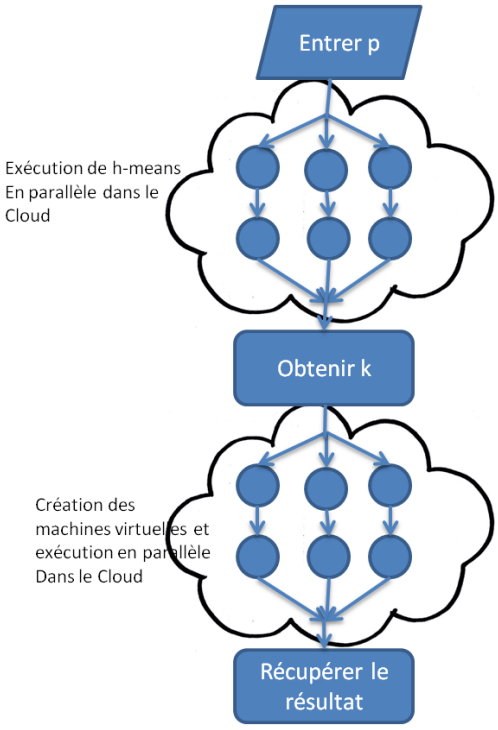


Figure 1 -Les points essentiels de l'algorithme

3. Un environnement distribué piloté par les workflows applicatifs

Dans notre environnement d'exécution, les applications de fouille de données sont représentées sous forme d'un *workflow* conformément au processus PMML [18] (*Predictive Model Markup Language*). Toute activité du *workflow* impliquant la manipulation de données génère des requêtes sur les données distribuées dans le Cloud.

PMML est un langage basé sur XML définissant une manière standard de décrire les modèles statistiques et de traitement de données au sein des applications décisionnelles. Ce langage est poussé par un consortium d'éditeurs, le *Data Mining Group* (DMG). Il a pour but de permettre le partage de modèles analytiques entre outils de « Business Intelligence » supportant ce standard. L'idée est bien de dépasser l'obstacle des technologies propriétaires en fournissant la possibilité de définir un modèle de traitement de données dans une application, qui puisse être consulté et réexploité dans un autre outil. Le format de description proposé prend la forme d'un XML Schema.

PMML intègre l'ensemble des informations nécessaires à l'exploitation d'un modèle d'analyse : le type de données en entrée et en sortie, le mode d'exécution du modèle, et la manière de générer et d'interpréter les résultats en s'appuyant sur ce standard. Egalement, il a pour vocation de couvrir l'ensemble des grandes méthodes de datamining.

Le moteur de workflow utilisé dans notre environnement est KNIME [16]. Il s'agit d'un environnement logiciel d'exécution de workflow (appelé aussi pipeline) libre de droit. Cet environnement est basé sur un principe modulaire, c'est à dire que les différentes étapes du pipeline sont représentées par des nœuds. Chaque nœud a une fonction précise et possède aucune ou plusieurs entrées et/ou aucune ou plusieurs sorties. De plus chaque nœud peut prendre en entrée la ou les sorties d'un ou plusieurs nœuds. KNIME est fourni avec un certain nombre de nœuds spécialisés pour la visualisation de données, l'analyse statistique et la fouille de données et offre la possibilité d'en créer des nouveaux pour interfacer par exemple des applications existantes. KNIME inclut des fonctionnalités de lecture / écriture au format PMML. Il est donc capable d'importer un fichier respectant le standard PMML et de le traduire en son propre format de pipeline comme d'exporter un pipeline KNIME au format PMML.

Notre environnement de test est basé sur un Cloud hybride. Ce type de Cloud est connu aussi sous l'appellation « Cloudbursting ». Il s'agit d'une extension des ressources du Cloud privé local avec celles d'un Cloud public. Ceci dans le but de combiner les ressources locales avec celles fournies par le Cloud public. Dans notre cas le fournisseur public est Amazon. L'intérêt d'une architecture de cloudbursting c'est qu'elle permet une haute évolutivité des environnements et améliore l'aspect « élasticité » des ressources.

Le choix du Cloud OpenNebula [6, 16] est motivé par le fait qu'il est l'un des plus riches en fonctionnalités open source des gestionnaires d'infrastructures virtuelles (*Virtual Infrastructure*). Il est basé sur le système d'exploitation Linux et a été initialement conçu pour gérer des infrastructures virtuelles locales tout en incluant des interfaces d'accès distants permettant la création de clouds publics.

OpenNebula met à disposition quatre API (*Application Programmable Interface*) au total :

- XML-RPC [19] pour les interactions locales.
- Libvirt [15] pour les interactions locales.
- un sous-ensemble de l'API Amazon EC2 (Query).
- l'API du cloud OpenNebula lui-même pour un accès public [14, 16].

L'architecture de OpenNebula est une architecture modulaire. Elle englobe plusieurs composants enfichables spécialisés. Le module principal orchestre les serveurs physiques avec leurs contrôleurs haut niveau (*hypervisors*), les nœuds de stockage et le réseau. Les opérations d'administration sont effectuées à travers des pilotes enfichables (*pluggable drivers*), qui interagissent avec les API des hyperviseurs, du stockage, du réseau et des clouds publics.

OpenNebula contient un module ordonnanceur (*Scheduler*) qui offre des fonctionnalités d'allocations de ressources dynamiques. Il est chargé d'assigner les requêtes de machines virtuelles (VM) en attente aux hôtes physiques. Le Scheduler donne la possibilité aux administrateurs de choisir différents types de gestion de ressources tels que le déploiement des VM sur un nombre réduit de hôtes physiques ou de maintenir une configuration d'équilibrage de charge. Egalement, il prend en charge plusieurs approches de virtualisations telles que Xen Hypervisor, KVM, et VMware vSphere. Ceci tout en permettant de déplacer des instances en cours d'exécution entre les différents nœuds connectés. Cependant, OpenNebula supporte uniquement les fonctionnalités de base de l'API de EC2 SOAP et de EC2 Query du cloud Amazon.

Notre plateforme vise à supporter l'exécution de notre algorithme CloudMeans lequel, comme décrit dans la section précédente, englobe une variante de l'algorithme k-means. La parallélisation implique de rendre asynchrone la suite des traitements en les décomposant de manière efficace comme une suite de processus connectés les uns aux autres avec des files d'attente de messages selon le pattern *pipe and filter*. Les unités de traitement successives attendent l'arrivée d'une tâche, la traitent et passent le résultat à l'étape suivante.

Une requête de parallélisation implique une phase de mappage des données à traiter, de lancer le traitement approprié sur ces données puis de procéder à la collecte du résultat du traitement effectué. Pour cela nous avons découpé le traitement en trois phases principales : la phase d'initialisation, la phase de mappage et la phase de collecte.

Pour implémenter le mécanisme d'échange des données et la collecte du résultat final entre les différentes images du cloud public Amazon EC2, nous avons utilisé Amazon SQS [20] (*Simple Queue Service*) qui permet la création de *pipelines* (files de traitements) hautement adaptables avec des unités de traitement interconnectées de façon souple. Ces files d'attentes permettent la flexibilité, l'asynchronisme et la tolérance aux pannes. Chaque étape du pipeline récupère des unités de travail d'une instance du service de file d'attente, traite l'unité de travail de façon appropriée et écrit les résultats dans une autre file d'attente pour les traitements ultérieurs. Il faut noter que les files d'attente fonctionnent bien lorsque les ressources nécessaires (temps, capacité processeur, vitesse d'entrées-sorties) de chaque étape pour une unité de travail particulière varient beaucoup.

Une file SQS contient des indicateurs sur l'emplacement des données à traiter. Dans la version actuelle de notre environnement de test nous avons définit manuellement le nombre des AMI (Amazon Machine Image) qui seront instanciées dans l'environnement EC2. Ce nombre sera défini par l'algorithme CloudMeans une fois finalisé en tenant compte des paramètres et contraintes citées dans la section précédente.

4. Conclusion et suite de travaux

Dans ce papier nous avons présenté les premiers éléments de notre algorithme ayant pour objectif de définir le nombre optimal de machines virtuelles à créer dans un environnement de cloud en mettant à profit une variante parallélisée de l'algorithme *k-means.* Nous avons également présenté notre environnement cloud de test lequel est un cloud hybride construit avec OpenNebula et Amazon EC2. Cet environnement constitue la base de nos futurs travaux sur la parallélisation et l'amélioration des performances des algorithmes de clustering dans le Cloud pour les applications de fouille de données.

Nous avons également présenté les principales technologies et les composants constituant notre environnement de tests. Notre environnement exploite efficacement l'ensemble des avantages attendus d'un environnement cloud à savoir l'élasticité, la scalabilité, la flexibilité et la simplicité d'utilisation.

Les différents tests ont démontré que l'algorithme *h-means* constituait un bon point de départ de notre algorithme qui vise à optimiser le nombre de machines virtuelles dans le but d'utiliser les ressources plus efficacement. La prochaine étape consiste à mettre à profit CloudMean dans le but de segmenter les données dans la phase du pré-traitement du "data mining".

Nous appliquons ce modèle sur l'algorithme *Apriori* qui a comme but la découverte des règles d'association à partir d'un ensemble de données appelé transactions [7]. Les règles d'association, étant une méthode d'apprentissage non supervisé, elles permettront de découvrir à partir d'un ensemble de transactions, un ensemble de règles qui exprime une possibilité d'association entre différents items. Une transaction est une succession d'items exprimée selon un ordre donné ; de même, l'ensemble de transactions contient des transactions de longueurs différentes.

Nous travaillons actuellement pour proposer un nouvel algorithme dont le but est de chercher plus efficacement les sous-ensembles fréquents ; l'idée étant d'appliquer auparavant notre version actuelle de l'algorithme dans le but d'améliorer le "point de départ" de la recherche des sous ensembles fréquents (au lieu de partir des candidats de longueurs 1). Nous sommes en train de réaliser des tests sur ce nouvel algorithme ainsi que l'influence des différents paramètres d'entrée sur les résultats ; notre but, dans ce cas, étant de le compléter pour assurer la couverture des sous-ensembles fréquents.

5. Références bibliographiques

[1] R. Moore, T.A Prince and M. Ellisman "Data-intensive computing and digital libraries" Commun. ACM, vol. 41, nà 11, pp 56-62, 1998.

[2] V. Fiolet, B. Toursel, Distributed data mining, Scalable Computing: Practice and Experiences 6 (1) (2005) 99–109.

[3] Suraj Pandey «Scheduling and Management of Data Intensive Application Workflows in Grid and Cloud Computing Environments» – PhD University of Melbourne – December 2010.

[4] T.Kosar and M. Livny "Stork : Making Data Placement a First Class Citizen in the Grid" in ICDCS'04 Proceedings of the 24th International Conference on Distributed Computing Systems. Washington, DC, USA IEEE 2004 pp. 342-349

[5] OpenNebula. http://opennebula.org

[6] Amazon Elastic Compute Cloud (Amazon EC2), http://aws.amazon.com/ec2.

[7] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In ACM SIGMOD Conference on Management of Data, 1993.

[8] J. Hartigan. Clustering Algorithm. John Wiles and Sons, NewYork,1975.

[9] R. Howard. Classifying a population into homogeneous groups. J.R. Lawerence (Ed.), Operational Research in the Social Sciences, 1966.

[10] K.Stoffel and A.Belkoniene. Parallel k/h-means clustering for large data sets. In Lecture Notes in Computer Science 1685, Proceedings of the 5th International Euro-Par Conference on Parallel Processing. Springer, 2004.

[11] S. Rahimi, M. Zargham, A. Thakre, and D. Chhillar. A parallel fuzzy c-mean algorithm for image segmentation. In IEEE Annual Meeting of the Fuzzy Information, Processing NAFIPS 4, volume 1, pages 234–237,2004.

[12] T.Jinlan, Z.Lin, Z.Suqin, and L.Lu. Improvement and parallelism of k-meansclustering algorithm. Tsinghua Science and Technology, 3(10) :277–281, 2005.

[13] Y.Zhang, Z.Xiong, J.Mao, and L.Ou. The study of paralle lk-means algorithm. In 6th World Congress on Intelligent Control and Automation 2, pages 5868–5871, 2006.

[14] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, Virtual infrastructure management in private and hybrid clouds, IEEE Internet Computing, 13(5):1422, September/October, 2009.

[15] Libvirt: The Virtualization API, Terminology and Goals, http://libvirt.org/goals. html, 22/4/2010.

[16] Distributed Systems Architecture Group, OpenNebula: The open source toolkit for cloud computing, http://www.opennebula.org, 22/4/2010.

[17] KNIME http://www.knime.com/

[18] PMML http://www.dmg.org/v4-0-1/GeneralStructure.html

[19] XML-RPC Specification http://xmlrpc.scripting.com/spec.html#update3

[20] Amazon Simple Queue Service (Amazon SQS) http://aws.amazon.com/fr/sqs/

[21] Avinash Lakshman and Prashant Malik. Cassandra - A Decentralized Structured Storage System.

Licorn*: construction de réseaux de régulation chez l'homme

Ines Chebil*, Mohamed Elati**, Rémy Nicolle**, Christophe Rodrigues*, Céline Rouveirol*

*LIPN — CNRS UMR 7030, Université Paris 13
99, av. J-B Clément, F-93430 Villetaneuse
prenom.nom@lipn.univ-paris13.fr
**iSSB, Institute of Systems and Synthetic Biology, University of Evry,
Genopole Campus 1, Genavenir 6, 5 rue Henri Desbruères, 91030 EVRY Cedex, France
prenom.nom@issb.genopole.fr

Résumé. Nous avons proposé un algorithme original de Fouille de Données, LICORN, afin d'inférer des relations de régulation coopérative à partir de données d'expression. LICORN donne de bons résultats s'il est appliqué à des données de levure, mais le passage à l'échelle sur des données plus complexes (e.g., humaines) est difficile. Dans cet article, nous proposons une extension de LICORN afin qu'il infère de manière efficace des ensembles de réseaux de régulation candidat pour un gène cible. D'autre part, nous avons proposé une autre extension de LICORN : sélectionner plusieurs réseaux parmi ces réseaux candidats (au lieu d'un seul dans LICORN). Ces réseaux sont combinés grace un mécanisme de vote pour prédire l'état de leur gène cible. Une évaluation préliminaire sur des données de transcriptome de tumeurs de vessie montre que ces deux améliorations permettent effectivement de passer à l'échelle sur des données de transcriptome humaines, tout en améliorant significativement les performances de prédiction par rapport à LICORN.

1 Introduction

Un des principaux objectifs de la biologie moléculaire consiste à comprendre la régulation des gènes d'un organisme vivant dans des contextes biologiques spécifiques. Les facteurs de transcription sont les régulateurs de la transcription qui vont réagir avec les promoteurs de la transcription des gènes cibles. Les techniques récentes d'analyse du transcriptome, telles que les puces à ADN, permettent de mesurer simultanément les niveaux d'expression de plusieurs milliers de gènes. Nous avons déjà décrit le système LICORN (Elati et al., 2007) qui se fonde sur un modèle de régulation locale coopérative : chaque gène peut être régulé par un ensemble des co-activateurs et/ou un ensemble de co-inhibiteurs ; ces corégulateurs agissent collectivement pour influencer leur(s) gène(s) cible(s). LICORN met en oeuvre une approche originale de Fouille de Données afin d'inférer des relations de régulation coopérative à partir de données d'expression. Cet algorithme a été évalué avec succès sur des données publiques de transcriptome de levure. L'application de LICORN sur des données de transcriptome humaines est plus complexe (Elati et Rouveirol, 2008), car le nombre de régulateurs connus est plus important, et nécessite un temps de calcul considérable. En effet, les gènes de faible support vont avoir

un nombre très élevé de régulateurs candidats. Nous proposons dans ce travail d'étendre LICORN pour qu'il puisse i) rechercher les n-meilleurs co-régulateurs candidats afin de borner la complexité de la recherche des réseaux candidats ii) combiner plusieurs réseaux candidats pour prédire l'état de leur gène cible.

La suite de cet article est organisée comme suit. Dans la section 2, nous introduisons brièvement le principe de LICORN. Dans la section 3, nous détaillons l'extension de LICORN à la recherche adaptative. Nous exposons des résultats préliminaires concernant l'application de cette approche à des données de cancer de la vessie dans la section 5 et enfin, nous concluons par les perspectives ouvertes par ce travail.

2 LICORN : Algorithme d'apprentissage

Étant donnés :
- un ensemble de régulateurs $\mathcal{R}$ et un autre de gènes cibles $\mathcal{G}$
- deux matrices d'expression discrétisées (MF, MG) à valeurs dans $\varepsilon = \{-1, 0, 1\}$, codant respectivement les états de l'ensemble des régulateurs de $\mathcal{R}$ et des gènes cibles de $\mathcal{G}$ pour un échantillon S de n conditions : $S = s_1, \ldots, s_n$
- un programme de régulation RP : $\varepsilon \times \varepsilon \rightarrow \varepsilon$, qui associe à un état d'un ensemble d'activateurs $\subseteq \mathcal{R}$ et à un état d'un ensemble d'inhibiteurs $\subseteq \mathcal{R}$ un état du gène cible [1]
- une fonction de score local $h : \varepsilon^n \times \varepsilon^n \rightarrow \mathbb{R}$, permettant d'évaluer un réseau local de régulation,

notre but est de trouver, pour chaque gène cible, l'ensemble de régulateurs qui explique au mieux son niveau d'expression, i.e. le graphe qui minimise la différence entre la valeur d'expression prédite et la valeur d'expression observée pour un gène cible.

Pour résoudre cette tâche, nous avons formalisé ce problème d'apprentissage de réseaux de régulation comme un problème de recherche dans un espace d'états, les états étant les motifs satisfaisant un ensemble de contraintes anti-monotones (Mannila et Toivonen, 1997). L'algorithme de LICORN (Elati et al., 2007) décompose la tâche d'apprentissage de structures en trois étapes indépendantes que nous rappelons brièvement ici. Par la suite, chaque gène ou régulateur g est représenté par deux ensembles supports $\subseteq S$, son 1-support $\mathcal{S}_1(g)$ et son -1-support $\mathcal{S}_{-1}(g)$. LICORN construit d'abord, en utilisant une extension de l'algorithme *Apriori* (Agrawal et al., 1993) qui manipule en parallèle les deux supports (1 et -1-supports), le treillis CL des ensembles de corégulateurs fréquents. Un corégulateur est fréquent s'il est fréquent dans MF pour au moins une des valeurs d'intérêt, ici 1 ou -1.

Ensuite LICORN calcule pour chaque gène cible g un ensemble de co-régulateurs candidats, de taille réduite par rapport à celle de tous les co-régulateurs fréquents CL. Le critère pour qu'un corégulateur puisse participer au programme de régulation d'un gène cible est d'observer dans les données d'expression une covariation significative de leurs niveaux d'expression. La contrainte de corégulation teste la taille de l'intersection entre les supports du gène et ceux du corégulateur.

Definition 1 (Contrainte de corégulation) *Soit une combinaison de co-régulateurs C, un gène cible g, et leurs supports respectifs $\mathcal{S}_x(C)$ et $\mathcal{S}_y(g)$ pour les états $x, y \in \{-1, 1\}$. C co-régule*

1. Ce programme de régulation permet de calculer l'expression prédite pour un gène dans un échantillon, étant donnés les états de ses activateurs et de ses inhibiteurs.

le gène g, noté **cov**(C, g, x, y) *si et seulement si* $\frac{|\mathcal{S}_y(g) \cap \mathcal{S}_x(C)|}{|\mathcal{S}_y(g)|} \geq T_o$*, où* T_o *est un seuil fixé par l'utilisateur.*

Nous distinguons les complexes de régulateurs satisfaisant **cov** pour chaque gène cible selon leur mode de régulation : les complexes d'activateurs candidats $\mathcal{A}(g)$ co-varient positivement avec le gène g, en revanche, les complexes d'inhibiteurs candidats $\mathcal{I}(g)$ co-varient négativement avec le gène cible :

$$\mathcal{A}(g) = \{A \in CL \mid \textbf{cov}(A, g, x, x); x \in \{-1, 1\}\}$$

$$\mathcal{I}(g) = \{I \in CL \mid \textbf{cov}(I, g, x, -x); x \in \{-1, 1\}\}$$

Nous avons décrit dans (Elati et al., 2007) comment nous exploitons l'anti-monotonie de **cov** pour effectuer une recherche efficace dans le treillis des régulateurs candidats CL. Ainsi, nous pouvons calculer l'ensemble de tous les graphes de régulation candidats pour chaque gène cible g, à partir des complexes d'activateurs et d'inhibiteurs extraits comme suit :

$$\mathcal{C}(g) = \{(A, I) | A \in \mathcal{A}(g), I \in \mathcal{I}(g) \text{ and } A \cap I = \emptyset\}$$

Un GRN candidat pour un gène cible g, ou tout simplement un GRN, est un élement de $\mathcal{C}(g)$. Enfin, à partir de chaque $\mathcal{C}(g)$, LICORN extrait le réseau de régulation (GRN) pour chaque gène cible qui explique au mieux les données d'expression, à l'aide d'une fonction de score (l'erreur absolue moyenne) qui mesure la capacité des régulateurs à prédire le niveau d'expression du gène cible g. Ensuite, LICORN sélectionne les GRNs statistiquement significatifs en se fondant sur une méthode (non décrite ici) à base de permutations.

3 Le passage à l'échelle de l'algorithme de LICORN

LICORN a été conçu et appliqué à des données de levure et nous avons décrit une première extension de LICORN à des données d'expression de gènes humains (Elati et Rouveirol, 2008). Le passage à des données humaines pose un sérieux problème de passage à l'échelle, en particulier, concernant l'étape de calcul des co-régulateurs candidats pour chaque gène cible.

Cette étape est contrôlée jusqu'à présent par la contrainte de co-régulation. La définition d'un seuil minimal pour la contrainte de corégulation, bien que très utile pour parcourir l'espace de recherche, est critique. En effet, les gènes peu exprimés vont avoir un nombre très élevé de corégulateurs candidats non pertinents (i.e. des corégulateurs très fréquemment exprimés, mais néanmoins non corrélés avec l'expression du gène cible (voir figure 1) entraînant un temps de calcul considérable.

Afin d'améliorer notre algorithme en vue de l'appliquer sur des données humaines de cancer, nous l'avons étendu pour qu'il puisse calculer pour chaque gène les seuls k-meilleurs régulateurs. Cette nouvelle stratégie ne nécessite pas de définir a priori un seuil de corégulation minimal, mais nécessite de fixer une valeur pour k. En fait, il ne s'agit plus de vérifier la contrainte de co-régulation isolément pour chaque régulateur mais de comparer les corégulateurs entre eux et d'extraire un nombre approprié de meilleurs corégulateurs candidats.

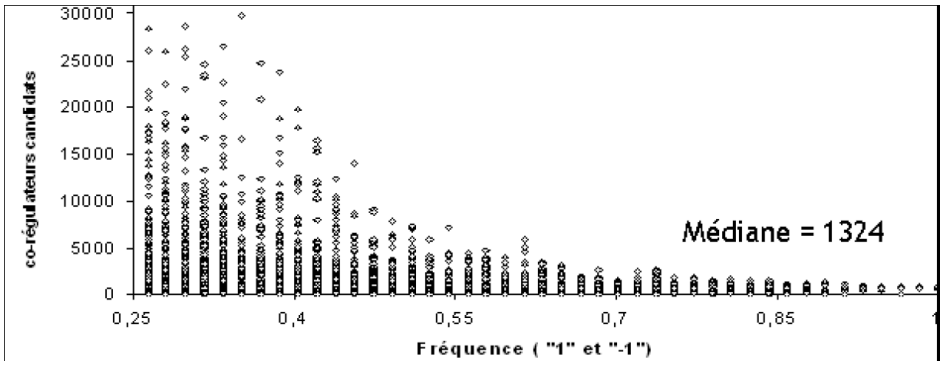


FIG. 1 – *Chaque point dans le graphique est un gène cible avec sa fréquence d'expression et le nombre de ses co-régulateurs candidats générés par* LICORN *($T_o = 0,3$).*

Definition 2 **k-meilleur motifs** *Soient $\mathcal{L}$ un langage de motifs, k un entier strictement positif, soit f une mesure d'intérêt, extraire les k-meilleurs motifs de $\mathcal{L}$ pour f revient à calculer un plus grand ensemble de taille inférieure ou égale à k :*

$$MM_{f,n} = \{Y \in \mathcal{L} | \forall X \in \mathcal{L}\, t.q.\, X \notin MM_{f,n}, f(Y) > f(X)\}$$

Cette contrainte compare les motifs de $\mathcal{L}$ les uns aux autres afin de conserver uniquement les n-meilleurs.

De telles contraintes existent dans la littérature (Fu et al., 2000). Récemment, Soulet et Crémilleux (2007) ont regroupé ces contraintes sous la notion de *contrainte globale*.

Fu et al. (Fu et al., 2000) étaient les premiers à introduire la problématique d'extraction de N-meilleurs motifs fréquents. Les auteurs adaptent APRIORI pour ajuster le seuil de fréquence minimale au fur et à mesure de l'extraction. Depuis, d'autres variantes ont été proposées utilisant des structures particulières d'arbre pour optimiser la recherche (e.g. , FP-tree (Han et al., 2002), COFI-tree (Ngan et al., 2005)). Tous ces travaux sont restreints à la mesure de fréquence comme mesure d'intérêt.

3.1 Algorithme de recherche des k-meilleurs co-régulateurs

Soit CL le inf-demi-treillis des co-régulateurs fréquents de MF, soit un gène cible g et soit un entier k. Le but est de calculer de la manière la plus efficace les k meilleurs corégulateurs de CL qui co-varient fréquemment avec le gène g.

Les paramètres de cet algorithme de recherche des k-meilleurs co-régulateurs sont i) le nombre k des solutions recherchées ii) T_o, le seuil minimal sur la contrainte de co-régulation, en dessous duquel on ne veut plus développer ce co-régulateur. $Ouvert$ est trié par ordre de score de corégulation décroissant. L'algorithme 1 effectue une recherche de type meilleur d'abord dans le inf-demi-treillis des co-régulateurs CL. L'ensemble des noeuds en attente de développement est tout d'abord initialisé à $CL(1)$ ($CL(1)$ est l'ensemble des noeuds de niveau 1 de CL). Le noeud de meilleur score de $Ouvert$ ($Node$) est tout d'abord développé en recherchant ses successeurs dans CL. Les successeurs de $Node$ sont ajoutés à $Ouvert$ en fonction de

leur score de co-régulation. Comme la contrainte de corégulation est anti-monotone, le score des successeurs de $Node$ est nécessairement inférieur ou égal au score de $Node$, assurant que tout nouveau noeud développé a un score inférieur ou égal à $Node$. Quand la taille de Sol atteint k, le processus d'ajout à Sol continue tant que le score du meilleur noeud de $Ouvert$ est le même que celui du dernier noeud de Sol.

Algorithm 1 Ad-GenCoreg : recherche meilleur d'abord des k-meilleurs co-régulateurs pour un gène g.

Require: g, CL, x, y, n tel que $x = y = 1$ pour le calcul de $\mathcal{A}(g)$, oo $x = -y = 1$ pour le calcul de $\mathcal{I}(g)$
Ensure: Sol, les k-meilleurs régulateurs pour g
1: **Begin**
2: $Ouvert := \emptyset$; $Sol := \emptyset$;
3: **for all** $c \in CL(1)$ **do**
4: $Ouvert := mise\_a\_jour(Ouvert, c)$
5: **end for**
6: $BN := noeud\_meilleur\_score(Ouvert); Ouvert := Ouvert - BN$;
7: $dernier\_score := score(BN)$;
8: **while** $((cov(BN, g, x, y) > T_o)\ AND\ ((|Sol| < k)\ OR\ (Score(N) = last\_score)))$ **do**
9: $Sol := Sol \cup \{BN\}$
10: **for all** $s \in successeurs(BN)$ **do**
11: $Ouvert := mise\_a\_jour(Ouvert, s)$
12: **end for**
13: $BN := noeud\_meilleur\_score(Ouvert)$
14: $dernier\_score := score(Node)$;
15: **end while**
16: **Return** Sol
17: **End**

4 Méthodes d'ensemble et LICORN*

Afin d'améliorer la performance de prédiction de LICORN* (i.e. la capacité des régulateurs appris à prédire le niveau d'expression de leur gène cible), nous proposons de sélectionner pour chaque gène cible g une collection de réseaux qui ensemble forment un classifieur plus performant en terme de prédiction de l'état de g.

4.1 Etat de l'art

Un grand nombre d'études (parmi lesquelles (Dietterich, 2000)) confirment que l'utilisation d'un ensemble de classifieurs donne généralement de meilleurs résultats qu'un seul classifieur. Par ailleurs, Dietterich démontre qu'une condition nécessaire et suffisante pour qu'un ensemble de classifieurs soit performant est que quelque soit le classifieur élément de l'ensemble, il doit être précis et différent des autres membres de l'ensemble. Ainsi, toute amé-

lioration des performances repose sur la précision et la diversité des classifieurs sélectionnés. Plusieurs approches permettent de sélectionner des classifieurs pour former un ensemble, celles qui sélectionnent les classifieurs les plus précis tels que *choose best* (Partridge et Yates, 1996) et celles qui sélectionnent les couples de classifieurs les plus divers tel que *Kappa Prunning* (Margineantu et Dietterich, 1997). D'autre part, Roli et Kittler (2002) ont proposé les approches séquentielles *Forward* et *Backward*. Pour l'approche *Forward*, chaque classifieur n'est ajouté à l'ensemble sélectionné que s'il optimise la précision de l'ensemble. De la même manière, pour l'approche *Backward*, chaque classifieur de l'ensemble n'est écarté définitivement de l'ensemble que si sa suppression permet d'avoir une meilleure précision dans l'ensemble des modèles restants. Cependant, ces dernières méthodes de sélection risque le sur-apprentissage. C'est pour cette raison que certains travaux (Niculescu-Mizil et al., 2009) sur la sélection de classifieurs de type *Forward* dans une bibliothèque de modèles ont essayé d'éviter ce problème par l'ajout de nouveaux procédés comme la sélection avec remise, l'initialisation par ensemble performant et le *Bagging* comme technique de construction d'ensemble.

Nous constatons que toutes ces approches ne tiennent pas compte du compromis entre la précision et la diversité. Margineantu et Dietterich (1997) suggèrent une nouvelle heuristique *Kappa-Error Convex Hull* qui permet de sélectionner simultanément les classifieurs les plus divers (moins précis) et les plus précis (moins divers). Cette dernière considère à la fois la diversité et la précision, cependant, elle reste une stratégie gloutonne qui ne permet pas de contrôler le nombre de classifieurs sélectionnés. D'autre part, Gacquer et al. (2009) ont proposé récemment un algorithme génétique appelé *DevGen* qui permet un contrôle direct de l'importance accordée à la diversité et à la précision de l'ensemble des classifieurs. $DevGen$ réalise une sélection des classifieurs à partir d'une première série de classifieurs. Cette sélection est basée sur la fonction de *Fitness*, initialement proposée par (Opitz et Maclin, 1999) pour la sélection des meilleurs sous ensembles de variables dans l'apprentissage d'un ensemble de classifieurs. La fonction de $Fitness$ est comme suit :

$$Fitness(E) = \alpha \times Acc(E) + (1 - \alpha) \times Div(E)$$

- $Acc(E)$ est la valeur de la précision de l'ensemble de classifieurs E. Elle est calculée en comparant les prédictions obtenues pour les individus d'un ensemble de validation par vote majoritaire pondéré de l'algorithme Adaboost à leurs classes réelles.
- $Div(E)$ est la valeur de la diversité d'un ensemble$E$ de classifieurs. Elle est calculée par le test de Kappa qui mesure l'accord entre observateurs lors d'un codage qualitatif en catégories (c'est une mesure d'accord entre deux codages seulement).
- α est le paramètre qui favorise l'importance soit de la précision soit de la diversité.

4.2 Contribution

Suite à l'analyse précédente, nous estimons qu'un compromis entre la précision et la diversité est nécessaire pour la sélection des réseaux. Nous proposons dans la suite VOTE-LICORN* , une nouvelle approche de sélection de réseaux parmi les réseaux clasiffieurs de LICORN en se basant sur la fonction de *Fitness*. Cette mesure al'avantage de coupler les deux critères de diversité et de précision dans une même fonction. Cependant, contrairement à l'approche $DevGen$, notre mesure de *Fitness* sera calculée pour chaque classifieur proposé pour un gène au lieu de tout l'ensemble et notre mesure de diversité sera basée sur la structure des réseaux de régulation (Birmele et al., 2008) et non sur leur diversité en terme de prédiction. Cette mesure est

ainsi calculée, pour chaque réseau de régulation r_i d'un gène dans $\{r_1..,r_E\}$, comme suit :

$$Fitness(r_i) = \alpha \times Acc(r_i) + (1-\alpha) \times Div(r_i)$$

Dans notre cas, la formule prend en paramètre :

- $Acc(r_i)$ qui représente la précision totale $(1 - erreur)$ sur les données d'apprentissage d'un réseau r_i
- $Div(r_i)$ qui représente la diversité de réseau r_i correspondant dans notre cas à la moyenne de l'index de $Jaccard$ entre le réseau r_i et tous les autres réseaux d'un gène sur les données d'apprentissage. Notons que l'index de $Jaccard$ [2] est une mesure rapide à calculer et qui permet d'évaluer la similarité entre deux ensembles.

Nous expliquons dans ce qui suit la mise en place de notre approche. Soient

- E : un ensemble de réseaux crées par Licorn* $\{r_1, .., r_E\}$ pour un gène g_k de MG
- $Fitness(r_i)$: l'évaluation calculée pour le réseau r_i

Nous commençons par calculer la $Fitness$ de chaque réseau r_i ; $i \in [1..E]$. Les réseaux sont ensuite triés suivant l'ordre décroissant de leurs valeurs de $Fitness$. On sélectionne à chaque étape le réseau de meilleure *Fitness* encore non exploré. On l'ajoutera à l'ensemble des réseaux sélectionnés E_s si l'ajout de ce réseau à E_s améliore la précision du nouvel ensemble. On note que dans notre cas, la précision de l'ensemble de classifieurs E est estimée par la précision du vote majoritaire de E sur les données d'apprentissage.

Algorithm 2 VOTE-LICORN* : Algorithme glouton pour la sélection d'un ensemble de réseaux

Require: $E = \{r_1, \ldots, r_{|E|}\}$ un ensemble de réseaux candidats inférés pour un gène donné et pour chaque réseau r_i, $Acc(r_i)$, $Div(r_i)$ et $Fitness(r_i)$
Ensure: E_s, un ensemble de réseaux sélectionnés

1: **Begin**
2: $R = E$ trié par ordre décroissant des $Fitness(r_i)$;
3: $Acc = 0$;$E_s = \emptyset$;
4: $BN = select\_best\_fitness\_node(R)$; $R = R - BN$;
5: **while** $(R \neq \emptyset)$ AND $(Acc \leq Acc(vote(E\_Select \cup Node)))$ **do**
6: $Acc := Acc(vote(E\_Select \cup BN))$;
7: $E\_Select := E\_Select \cup Node$;
8: $BN = select\_best\_fitness\_node(R)$; $R = R - BN$;
9: **end while**
10: **Return** E_Select = $\{r'_i, ..., r'_n\}$
11: **End**

5 Cadre applicatif et expérimentations : Cancer de la vessie

Dans le domaine du cancer, la technologie de puces à ADN est considérée comme un outil de diagnostic prometteur, car de nombreux gènes sont exprimés différentiellement entre les échantillons tumoraux et normaux. Mais, les causes de l'expression différentielle des gènes

2. L'indice de $Jaccard$ est le rapport entre la cardinalité de l'intersection des ensembles considérés et la cardinalité de l'union des ensembles. Soit deux ensembles A et B, l'indice est : $J(A,B) = \frac{A \cap B}{A \cup B}$

dans le cancer ne sont pas encore bien comprises. Beaucoup d'oncogènes agissent en tant que facteurs de transcription eux-mêmes ou ont un effet direct sur des facteurs de transcription. Quand ces oncogènes (par exemple, *MYC et RAS*) sont activés par un mécanisme génétique (mutation, amplification ou translocation), ils activent ou désactivent, directement ou indirectement, plusieurs facteurs de transcription qui changent le niveau d'expression de leurs gènes cibles par des mécanismes complexes. Cependant, le niveau d'expression de chaque gène cible est directement relié au niveau d'expression de ses régulateurs. Notamment, la capacité de notre méthode de détecter des régulations coopératives est cruciale pour son application aux eucaryotes supérieurs. Dans cette section, nous montrons les résultats de l'expérimentation avec des données humaines d'expression de cancer de la vessie.

5.1 Préparation des données

Dans cette étude, nous utilisons des données humaines de 57 échantillons (Stransky et al., 2006). Les puces à ADN utilisées sont des puces Affymetrix de type HG-U95A. Ces échantillons se divisent en cinq échantillons normaux (sujets dépourvus de cancer), et 52 échantillons de carcinomes de la vessie à différents stades de la maladie. Nous nous intéressons à trois tendances de variation du niveau d'expression d'un gène : dans un échantillon tumoral donné, le niveau d'expression d'un gène peut *augmenter* par rapport à son niveau *normal* (noté par 1 dans la matrice discrétisée) ; il peut *diminuer* par rapport à son niveau normal (noté par -1) ; ou encore être *stable* (noté par 0). Nous définissons alors la technique suivante de discrétisation afin de transformer la matrice originale O en une matrice discrète M qui code, pour chaque gène g, la tendance de variation du niveau d'expression dans les échantillons tumoraux par rapport au niveau normal, calculé comme la moyenne du niveau d'expression de g dans les échantillons normaux.

Soit $\mu_n(g)$ et $\sigma_n(g)$ la moyenne et l'écart type du niveau d'expression de g dans les échantillons normaux et ρ un paramètre fixé, entre autres, par la densité de la matrice souhaitée :

$$M(g,s_i) \quad = \quad \begin{cases} 1, & O(g,s_i) > (\mu_n(g) + \rho \cdot \sigma_n(g)) ; \\ \text{-1}, & O(g,s_i) < (\mu_n(g) - \rho \cdot \sigma_n(g)) ; \\ 0, & \text{sinon.} \end{cases} \tag{1}$$

Le paramètre ρ de discrétisation est fixé à 2, ce qui correspond à une densité de 30% de la matrice discrétisée. Afin de choisir une liste de facteurs de transcription, nous sélectionnons parmi les gènes représentés sur les puces ADN les facteurs de transcription dans la base TRANSFAC (Matys et al., 2003), en utilisant le symbole standard du gène ou les identificateurs SwissProt, si disponibles, ce qui nous amène à extraire un ensemble de 699 facteurs de transcription et 7224 gènes.

5.2 Protocole expérimental

Lors de la validation de LICORN* et VOTE-LICORN* , nous voulons mettre en évidence les points suivants : i) le volume des relations de régulation engendrées par LICORN* est nettement inférieur à celles engendrées par LICORN , ce qui conduit à un gain en temps considérable pour des performances en terme de prédiction similaires (section 5.2.1) ; ii) la sélection, pour un gène donné, d'un ensemble de réseaux de régulation (au lieu d'un seul réseau parmi ceux qui ont la meilleure erreur observée) améliore la prédiction avec VOTE-LICORN* (section 5.2.2).

Dans notre modèle, le niveau d'expression d'un gène cible g est une fonction discrète du niveau d'expression de ses régulateurs (A, I). Il est donc possible d'étudier la performance de notre algorithme en testant la capacité des régulateurs appris à prédire le niveau d'expression de leur gène cible sur un ensemble de test $\mathcal{T}$. Par la suite, nous utilisons l'erreur de prédiction comme mesure de distance entre le profil réel de l'expression du gène et celui estimé à partir de l'expression des ses régulateurs inférés par LICORN.

Pour qu'une mesure de performance soit objective, elle doit être évaluée sur un ensemble de validation qui n'a pas été utilisé pour apprendre le classifieur. La validation croisée consiste à diviser la population observée $\mathcal{S}$ en sous-groupes $\mathcal{S}_1, \ldots \mathcal{S}_K$. Pour $k = 1 \ldots K$, le classifieur est construit sur *l'ensemble d'apprentissage* $\mathcal{S} \setminus \mathcal{S}_k$, et sa performance est évaluée sur *l'ensemble de test* $\mathcal{S}_k$. Dans la pratique, *la 5 validation croisée* ($K = 5$) est souvent considérée, car elle permet une évaluation correcte de la performance avec un temps de calcul raisonnable. Nous proposons l'utilisation d'une 5x5-validation croisée afin d'avoir plus de valeurs d'erreurs de prédiction sur différents échantillons et pouvoir mieux évaluer nos approches (et la significativité d'une différence de performance entre ces approches). Nous utilisons pour cela un $t - test_{5\%}$ sur les erreurs de prédiction pour chaque gène pour pouvoir interpréter la significativité de la différence entre les résultats de LICORN et LICORN* ainsi qu'entre les résultats de LICORN* et VOTE-LICORN*.

Dans un premier temps, nous travaillons sur un échantillon de 10% de gènes tirés aléatoirement dans la population de gènes afin d'estimer la meilleure taille pour Sol (que l'on fixe égale à la taille de $Ouvert$ dans nos expériences). Nous faisons varier la taille de Sol et de $Ouvert$ de 1% à 10% de la taille totale de l'espace de recherche (CL), qui est de l'ordre de 16 500 corégulateurs fréquents pour un seuil de support minimum de 18%, pour un seuil de corégulation T_o fixé à 0,5 (Les parties gauches des tableaux Tab. 1 et Tab. 3).

Dans un deuxième temps, on va confronter les résultats des différentes approches pour tous les gènes avec différentes tailles de Sol et $Ouvert$. Le seuil de corégulation est fixé $T_o = 0,3$ pour LICORN* et VOTE-LICORN* afin d'augmenter la probabilité des co-régulations rares d'étre séléctionnés (les parties droites des tableaux Tab. 1 et Tab. 3).

Concernant le choix de la taille de E et de la valeur du paramètre α de notre approche VOTE-LICORN*, nous avons choisi de créer 100 réseaux pour chaque gène. On estime que $E = 100$ est un nombre assez grand pour faire apparaître des réseaux divers en structure et en qualité. Par rapport au paramètre α, nous avons choisi de ne pas pas favoriser l'un des critères de sélection (précision ou diversité) et nous avons donné la même importance à la qualité de prédiction qu'à la diversité entre les réseaux en posant $\alpha = 0,5$.

5.2.1 LICORN* vs. LICORN

Le tableau Tab. 1, permet de conclure que c'est une bonne méthodologie d'estimer la meilleure taille pour $Ouvert$ et SOl en sélectionnnant un échantillon de 10% des gènes, car cette estimation des performances relatives de LICORN et LICORN* se révèle stable et robuste sur la totalité des gènes. LICORN* prédit mieux ou aussi bien que LICORN pour toutes les tailles testées de Sol. Ceci nous amène à choisir pour Sol et $Ouvert$ une taille de 1% de $|CL|$, ce qui est suffisant pour obtenir un résultat équivalent avec un temps de calcul largement inférieur (voir Tab 2) et une meilleure performance (voir figure 2).

$n = k * \|CL\|$	1%	5%	10%	1%	5%	10%
LICORN* ≡ LICORN	259	252	250	4749	4695	4572
LICORN > LICORN*	193	198	199	389	370	374
LICORN* > LICORN	189	200	201	1138	1211	1330

TAB. 1 – *Performances relatives de* LICORN* *et* LICORN *pour 10% des gènes (partie gauche du tableau) et pour la totalité des gènes (6276) (partie droite du tableau)*

	LICORN	LICORN* (1%)
temps	19m18.748s	2m26.429s

TAB. 2 – *Estimation du temps de calcul pour la totalité des gènes*

5.2.2 VOTE-LICORN* **vs.** LICORN*

$n = k * \|CL\|$	1%	5%	10%	1%	5%	10%
LICORN* ≡ VOTE-LICORN*	107	106	108	1194	1208	1224
LICORN* > VOTE-LICORN*	71	67	64	906	932	929
VOTE-LICORN* > LICORN*	466	471	472	4738	4698	4685

TAB. 3 – *Performances relatives de* LICORN* *et* VOTE-LICORN* *pour 10% des gènes (partie gauche du tableau) et pour la totalité des gènes (6276) (partie droite du tableau)*

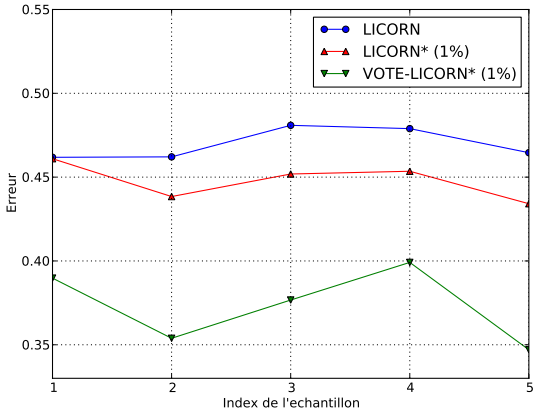


FIG. 2 – *Erreur de prédiction de 5-VC des différentes approches pour la totalité des gènes*

Le tableau Tab 3 montre que la sélection d'un ensemble de réseaux régulant un gène avec VOTE-LICORN* permet de prédire nettement mieux qu'avec un seul réseau de régulation avec LICORN*. Ceci est observé pour une échantillon aléatoire de 10% gènes comme pour la totalité. Le nombre k de réseaux sélectionné dépend du gène traité, il varie de 1 à 7 réseaux par gène. Ainsi, VOTE-LICORN* garantit une prédiction largement meilleure (voir figure 2) en un temps d'exécution très proche de LICORN* (2m56.446s pour 1% des corrégulateurs fréquents).

5.3 Conclusion

Licorn est un système de fouille de données qui calcule des réseaux de régulation complexes à l'aide d'outils de fouille de motifs contraints, combinés à une procédure de validation statistique des motifs appris. Licorn a été appliqué à des données d'expression de gènes chez la levure, mais il souffre de problèmes de passage à l'échelle lorsqu'il s'agit de l'appliquer à de données plus complexes, e.g. des données d'expression de gènes chez l'humain. Nous avons présenté dans cet article deux extensions de Licorn, une tout d'abord incluant une recherche de type n-meilleurs motifs. Ce type de recherche a permis de fixer un nombre maximal de co-regulateurs candidats par gène cible et de réduire considérablement le temps de calcul, tout en engendrant des réseaux dont les performances sont, dans une grande majorité des cas, équivalentes voire meilleures à celles obetnues par les classifieurs construits par Licorn. Dans un deuxième temps, nous avons étendu l'algorithme de prédiction de l'état d'un gène cible par une méthode d'ensemble. Les classifieurs de l'ensemble sont sélectionnés par un algorithme glouton guidé par un score original prenant en compte la qualité de l'ensemble en terme de précision ainsi que la diversité structurelle de l'ensemble des réseaux. Ce classifieur par ensemble se révèle plus performant que la méthode originale qui utilise uniquement le meilleur réseau calculé par Licorn ou Licorn* sur un jeu de données de cancer de la vessie. Ceci valide nos choix algorithmiques et nous autorise à penser que notre système, bien que la tâche soit complexe, passe à l'échelle pour l'inférence de réseaux de régulation pour des organismes plus complexes.

Références

Agrawal, R., T. Imielinski, et A. Swami (1993). Mining association rules between sets of items in large databases. In *Proceedings of the International Conference on Management of Data*, pp. 207–216.

Birmele, E., M. Elati, C. Rouveirol, et C. Ambroise (2008). Identification of functional modules based on transcriptional regulation structure. *BMC Proceedings 2*(Suppl 4), S4.

Dietterich, T. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems*, Volume 1857 of *Lecture Notes in Computer Science*, pp. 1–15. Springer Berlin / Heidelberg.

Elati, M., P. Neuvial, M. Bolotin-Fukuhara, E. Barillot, F. Radvanyi, et C. Rouveirol (2007). Licorn : learning cooperative regulation networks from gene expression data. *Bioinformatics 23*, 2407–2414.

Elati, M. et C. Rouveirol (2008). Recherche adaptative de structures de régulation génétique. In *Extraction et gestion des connaissances (EGC'2008), Actes des 8èmes journées Extraction*

et Gestion des Connaissances, Volume RNTI-E-11 of *Revue des Nouvelles Technologies de l'Information*, pp. 427–432. Cépaduès-Éditions.

Fu, A. W. C., R. Kwong, et J. Tang (2000). Mining n-most interesting itemsets. In *ISMIS*, pp. 59–67.

Gacquer, D., V. Delcroix, F. Delmotte, et S. Piechowiak (2009). On the effectiveness of diversity when training multiple classifier systems. In *ECSQARU*, pp. 493–504.

Han, J., J. Wang, Y. Lu, et P. Tzvetkov (2002). Mining top-k frequent closed patterns without minimum support. In *ICDM*, pp. 211–218.

Mannila, H. et H. Toivonen (1997). Levelwise search and borders of theories in knowledge discovery. *Data Min. Knowl. Discov 1*, 241–258.

Margineantu, D. D. et T. G. Dietterich (1997). Pruning adaptive boosting.

Matys, V., E. Fricke, R. Geffers, E. Gossling, M. Haubrock, R. Hehl, K. Hornischer, D. Karas, A. E. Kel, O. V. Kel-Margoulis, et D. U. Kloos (2003). Transfac : transcriptional regulation, from patterns to profiles. *Nucleic Acids Research 31*, 374–378.

Ngan, S. C., T. Lam, R. C.-W. Wong, et A. W. C. Fu (2005). Mining n-most interesting itemsets without support threshold by the cofi-tree. *Int. J. Business Intelligence and Data Mining 1*, 88–106.

Niculescu-Mizil, A., C. Perlich, G. Swirszcz, V. Sindhwani, Y. Liu, P. Melville, D. Wang, J. Xiao, J. Hu, M. Singh, W. X. Shang, et Y. F. Zhu (2009). Winning the kdd cup orange challenge with ensemble selection. *Journal of Machine Learning Research - Proceedings Track 7*, 23–34.

Opitz, D. W. et R. Maclin (1999). Popular ensemble methods : An empirical study. *J. Artif. Intell. Res. (JAIR) 11*, 169–198.

Partridge, D. et W. B. Yates (1996). Engineering multiversion neural-net systems. *Neural Comput. 8*, 869–893.

Roli, F. et J. Kittler (Eds.) (2002). *Multiple Classifier Systems, Third International Workshop, MCS 2002, Cagliari, Italy, June 24-26, 2002, Proceedings*, Volume 2364 of *Lecture Notes in Computer Science*. Springer.

Soulet, A. et B. Crémilleux (2007). Extraction des Top-k Motifs par Approximer-et-Pousser. In *actes des 7èmes journÈes Extraction et Gestion des Connaissances (EGC'07)*, pp. 271–282. Cépaduès.

Stransky, N. et al. (2006). Regional copy number-independent deregulation of transcription in cancer. *Nature Genetics 38*, 1386–1396.

Apport des relations spatiales dans l'extraction automatique d'informations à partir d'image

Bruno Belarte*, Cédric Wemmert*

*Université de Strasbourg, LSIIT, UMR CNRS 7005, Strasbourg, France
{belarte,wemmert}@unistra.fr

Résumé. L'évolution des technologies nous donne accès à des images d'un très haut niveau de détail. Les algorithmes de segmentations permettent d'extraire des objets d'intérêt de ces images. L'utilisation d'une base de connaissances experte permet de classifier ces objets. À l'heure actuelle, très peu de méthodes utilisent les connaissances spatiales pour analyser le résultat d'une segmentation. Nous proposons dans cet article une méthode pour formaliser et utiliser cette connaissance experte dans un processus d'analyse d'images de télédétection. Une analyse des limites de cette méthode permettra de faire émerger plusieurs perspectives de recherche.

1 Introduction

L'évolution croissante de la résolution d'acquisition des images dans tous les domaines (imagerie médicale, télédétection, photographie, etc.) permet d'obtenir des images d'un très haut niveau de détail. Ce bond technologique entraine une complication majeure dans l'analyse de ces images. En effet, l'augmentation de la finesse de la résolution induit une augmentation considérable de la masse d'information ainsi que du bruit, qui va venir perturber l'analyse. Les méthodes d'analyse reposant sur une approche basée pixel montrent dans ce cas clairement leurs limites car les objets recherchés dans l'image sont constitués d'un ensemble hétérogène de pixels.

C'est dans ce contexte qu'ont été développés les modèles d'analyse reposant sur une approche dite "basée région". Ces approches fonctionnent en deux phases et commencent par segmenter l'image en régions (agrégats de pixels homogènes). Ensuite, les régions obtenues forment l'ensemble d'objets à analyser. Cette approche offre de nouvelles possibilités. Il est maintenant possible de s'intéresser à des caractéristiques autres que la signature spectrale pour analyser ces images, telles que la taille ou l'élongation des objets. Une fois les objets caractérisés, il est possible d'en faire une analyse sémantique à l'aide d'une base de connaissances, qui va permettre de les classer. Plusieurs méthodes reposant sur l'utilisation de connaissances expertes existent. Cependant, un type de connaissances est encore très peu utilisée dans ce type d'approche, les relations spatiales entre les objets.

C'est cette connaissance en particulier, d'un haut niveau sémantique, que nous proposons d'intégrer dans un processus d'analyse d'image. Nous proposons dans cet article

une méthodologie visant à caractériser les objets extraits par la phase de segmentation à l'aide d'une base de connaissances construite avec l'expert et revenons sur les préconditions nécessaires à l'utilisation de ce genre de connaissances. Cette base intègre pour chaque objet recherché des caractéristiques spectrales (radiométrie), structurelles (forme, taille, etc.) et spatiales (relation de voisinage, proximité, etc.). L'approche est testée sur des images de télédétection à très haute résolution spatiale (THRS).

Le reste de l'article est organisé de manière suivante : la section 2 présente dans un état de l'art non exhaustif de différentes utilisations de connaissances dans la classification et l'analyse d'images, ainsi que sur le raisonnement spatial. La section 3 décrit la méthodologie proposée. Les différentes expérimentations réalisées ainsi que les résultats obtenus sur des images de télédétection sont exposés en section 4. Enfin, la section 5 conclut et évoque des perspectives pour ce travail.

2 Connaissances expertes et relations spatiales

Cette section présente un état de l'art de travaux récents d'une part sur la modélisation et l'utilisation de connaissances expertes dans le processus d'extraction d'informations en général ou dans des images, et d'autre part, sur la représentation et la formalisation des relations spatiales dans des images, avec un paragraphe dédié à son utilisation dans le cadre de l'analyse d'images de télédétection, qui nous intéresse dans cet article.

2.1 Utilisation de la connaissance experte en extraction d'informations

Classification guidée par la connaissance. Dans la classification non supervisée par contraintes, la connaissance est exprimée sous la forme de deux relations signifiant que deux objets *doivent* ou *ne peuvent pas* appartenir à la même classe. Wagstaff et al. (2001) proposent une version contrainte de l'algorithme K-Means utilisant de telles contraintes pour biaiser l'affectation des objets aux classes. À chaque étape, l'algorithme essaie de s'accorder avec les contraintes données. Bilenko et al. (2004) utilisent ces contraintes pour apprendre une fonction de distance biaisée par les connaissances.

Kumar et Kummamuru (2008) introduisent un autre type de connaissances, les comparaisons relatives, à travers un algorithme de classification non supervisée. Leur étude expérimentale démontre que l'algorithme proposé est plus précis et robuste avec ces relations qu'avec les contraintes présentées plus haut.

Pedrycz (2004) introduit plusieurs sortes de connaissances du domaine telles que la supervision partielle, les conseils basés sur la proximité ou sur l'incertitude. L'auteur propose plusieurs solutions pour exploiter et intégrer ces connaissances dans un algorithme Fuzzy C-Means.

Analyse d'images guidée par la connaissance. De nombreuses méthodes spécifiques existent pour détecter des objets particuliers dans une image Zhao et al. (2002); Peteri et al. (2003); Jin et Davis (2005). Ces méthodes intègrent les connaissances im-

plicites qu'ont leurs concepteurs sur les types d'objets recherchés. Ces connaissances sont ensuite difficiles à décrire si l'on souhaite améliorer ou adapter l'algorithme.

Afin de modéliser ces connaissances implicites, l'utilisation d'ontologie devient de plus en plus courante. Une ontologie Gruber (1995) est une spécification abstraite, une vue simplifiée du monde représentée dans un but précis. Une ontologie définit un ensemble de concepts, leurs caractéristiques et les relations entre ces concepts. L'utilisation d'ontologies est devenue de plus en plus courante dans les systèmes d'informations géographiques Fonseca et al. (2002). De plus, un intérêt croissant a été porté aux ontologies spécialisées pour l'analyse d'image Bittner et Winter (1999) depuis ces dix dernières années. La plupart des méthodes Forestier et al. (2008); Maillot et Thonnat (2008); Mezaris et al. (2004); Durand et al. (2007) formalisent les concepts pouvant être présents dans une image puis proposent une analyse sémantique de celle-ci en cherchant à identifier des représentants de ces concepts dans l'image.

2.2 Raisonnement spatial

Description de la connaissance spatiale. Freksa (1992) s'intéresse à la position d'un point par rapport à deux autres. Il définit huit orientations par rapport au segment caractérisé par ces deux points (*en haut, en haut à droite, à droite, en bas à droite, ...*). En combinant ces huit directions avec celles obtenues par rapport au segment, il obtient un ensemble de quinze directions. Freska propose une méthode pour inférer la position d'un point par rapport à un segment.

Une autre méthode est celle proposée par Moratz et al. (2000). Cette méthode s'intéresse à la position relative de deux segments. Chaque segment est orienté, et défini par son point de départ et son point d'arrivée. Un système de règles permet, à partir de la position d'un point par rapport à un segment, d'inférer la position relative des deux segments. Pour cela, est utilisé la position des deux extrémités d'un segment par rapport à l'autre segment.

Formalisation de la connaissance spatiale. La connaissance peut être formalisée à l'aide de la logique formelle. Renz et Nebel (1999) représentent les relations RCC-8 sous forme de logique modale, à l'aide de contraintes de modèle et de contraintes d'implications.

Kutz et al. (2004) formalisent le même problème sous forme de logique de description. En plus des opérateurs logiques usuels sont utilisés des quantificateurs universels et existentiels. Ce formalisme a l'avantage de permettre l'expression de contraintes quantitatives.

Papadias et Egenhofer (1997) utilisent un réseau de contraintes pour calculer la position relative de points. Les sommets du graphe représentent les points, les arêtes orientées représentent la contrainte spatiale entre les deux points. Un mécanisme d'inférence permet de calculer la position de deux points liés par un chemin dans le graphe.

Utilisation en télédétection. Les relations spatiales sont actuellement peu utilisées dans l'analyse d'image de télédétection. Inglada et Michel (2009) utilisent les relations RCC-8 pour détecter des avions dans une image de télédétection. Pour ce

faire, ils utilisent une segmentation multi-échelle, basée sur une image panchromatique sur laquelle sont appliquées deux fonctions conçues à partir d'opérateurs morphologiques par reconstruction sur une métrique géodésique. Le résultat obtenu permet de construire un graphe qui met en relation les éléments structurants des différents niveaux (chevauchement, ...).

Vanegas et al. (2009, 2010) détectent des alignements d'objets appartenant au concept maison en comparant les orientations de couples d'objets. Les résultats sont ensuite raffinés en calculant le parallélisme de ces alignements à des segments de route.

Stoica et al. (2001) cherchent à extraire le réseau routier d'une image de télédétection. Ils utilisent pour cela un processus d'inférence bayésienne basé sur les méthodes MCMC (Monte Carlo Markov Chain). Ce sont des méthodes d'échantillonnage à partir d'une distribution de probabilité basées sur le parcours de chaînes de Markov qui ont pour lois stationnaires les distributions à échantillonner. La méthode utilisée par Stoica est l'algorithme de Metropolis-Hastings, qui utilise des marches aléatoires entre les chaînes de Markov.

3 Méthodologie

La méthode proposée repose sur la segmentation d'une image et est une extension des travaux de Forestier et al. (2008). Nous nous intéressons aux relations spatiales entre les objets d'intérêt et voulons vérifier la pertinence de leur utilisation dans un processus d'analyse d'images. À chaque objet est associé un ensemble d'attributs spatiaux, structurels et spectraux. Ces attributs vont permettre de raffiner les concepts attribués aux objets. Dans la section 3.1, nous présentons la méthode d'extraction des objets d'intérêt. La section 3.2 décrit la base de connaissances. L'algorithme de raffinement est présenté dans la section 3.3.

3.1 Extraction des objets d'intérêt

Dans cette section nous présentons la méthode mise en place pour extraire les objets d'intérêt. Pour ce faire, nous segmentons l'image en entrée. Pour vérifier l'intérêt et l'efficacité de l'utilisation de relations spatiales, nous considérons qu'un segment correspond à un objet d'intérêt. Chaque objet est caractérisé par des attributs spectraux, structurels et spatiaux. Ces objets sont placés dans un graphe d'adjacence. Soit Ω l'ensemble des objets d'intérêt.

Définition 1 : Un objet d'intérêt est un objet ayant une réalité sémantique (dans notre cas, dans un paysage urbain).

Définition 2 : Un attribut spectral est une valeur calculée à l'aide de masques binaires représentant une valeur seuillée pour des filtres spectraux donnés (par exemple, dans le cas d'images de télédétection : NDVI, NDWI2, CI, SBI, intensité). Un attribut spectral est calculé pour chaque masque en donnant le pourcentage (exprimé entre 0 et 1) de pixel de l'objet courant appartenant au masque.

Définition 3 : Un attribut structurel est une valeur caractéristique de la forme de l'objet (taille, élongation, ...).

Définition 4 : Un attribut spatial est une valeur quantifiant une relation spatiale donnée entre deux objets d'intérêt (distance euclidienne, ...). Ces attributs sont calculés en parcourant le graphe. La position d'un objet dans le graphe donne des informations de voisinage sur ces objets.

À la fin de ce processus, les objets extraits sont étiquetés avec le label *objet*. L'image n'est maintenant plus utilisée, la méthode proposée ne s'intéresse qu'au graphe d'objets d'intérêt.

3.2 Base de connaissances

Dans cette section nous définissons la base de connaissances ainsi que les différentes notions associées. La base de connaissances est définie par un dictionnaire de concepts ainsi qu'une hiérarchie qui ordonne ces concepts. Soient Θ l'ensemble des concepts, Γ l'ensemble des contraintes et $\Re$ l'ensemble des relations spatiales. Les concepts sont définis comme suit.

Définition 5 : Un concept est défini par un ensemble de contraintes, quantifiables ou non.

Définition 6 : Une contrainte quantifiable $C_{a,min,max} \in \Gamma$ est définie comme une condition sur un attribut d'un objet. La contrainte est satisfaite si l'attribut est dans l'intervalle $[min, max]$ définie par la contrainte.

Définition 7 : Une contrainte non quantifiable $C_{r,c} \in \Gamma$ est définie comme une contrainte relationnelle. L'objet courant doit avoir au moins une relation r avec un objet de concept c dans le graphe.

Définition 8 : L'opérateur de négation, noté $\neg$, est utilisé lorsqu'une contrainte *ne* doit *pas* être satisfaite.

Définition 9 : Un concept $\mathcal{C} \in \Theta$ est défini par un ensemble de contraintes.

Définition 10 : La relation de sous-concept est une relation d'ordre partiel, notée $\preceq_\Theta$, définie par :

$$\forall \mathcal{C}_i, \mathcal{C}_j \in \Theta^2, \mathcal{C}_i \preceq_\Theta \mathcal{C}_j \text{ signifie que } \mathcal{C}_i \text{ est un sous-concept de } \mathcal{C}_j.$$

On note $\preceq^i_\Theta$ la relation d'ordre i. Ainsi, $\mathcal{C}_i \preceq^1_\Theta \mathcal{C}_j$ signifie que $\mathcal{C}_i$ est un sous-concept descendant directement de $\mathcal{C}_j$.

Ces concepts sont ordonnés hiérarchiquement. Un sous-concept hérite des contraintes du concept parent et en ajoute d'autres qui permettent de le différencier de ses concepts frères. Un extrait de la hiérarchie utilisée pour notre exemple en télédétection est donné dans la figure 1. Deux fonctions sont définies pour calculer la correspondance entre un objet et un concept.

Définition 11 : Le prédicat booléen $satisfy(C, O), C \in \Gamma, O \in \Omega$ définit si oui ou non la contrainte C est satisfaite par l'objet O. Ce prédicat est défini comme suit :

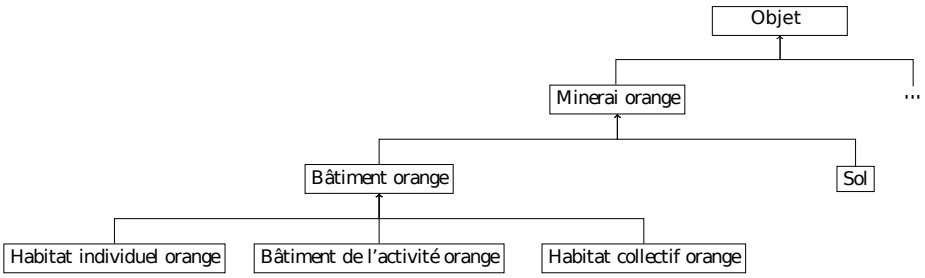


FIG. 1 – *Extrait de la hiérarchie correspondant aux concepts raffinant le concept* minerai orange. *Cette hiérarchie est utilisée pour valider le modèle sur des images de télédétection.*

$$\begin{array}{rll} satisfy(C_{a,min,max}, O) = & \text{vrai} & \text{si } min \leq O(a) \leq max \\ & \text{faux} & \text{sinon} \\ satisfy(C_{r,concept}, O) = & \text{vrai} & \text{si } \exists r \in \Re, \exists O' \in \Omega \text{ tels que } OrO' \\ & \text{faux} & \text{sinon} \\ satisfy(\neg C, O) = & \text{vrai} & \text{si } satisfy(C, O) = \text{faux} \\ & \text{faux} & \text{sinon} \end{array}$$

Définition 12 : La fonction $match(\mathcal{C}, O)$ calcule la correspondance entre un objet O et un concept $\mathcal{C}$. Cette fonction est définie comme suit :

$$match(\mathcal{C}_i, O) = \frac{m}{n}.$$

avec n le nombre de contraintes de $\mathcal{C}$ et m le nombre de contraintes de $\mathcal{C}$ telles que $satisfy(C, O) =$ vrai.

3.3 Raffinement de concept attribué

Dans cette section nous présentons la méthode mise en place pour raffiner les concepts attribues aux objets d'intérêt. Soit O un objet auquel est attribué un concept $\mathcal{C}_j$ de la hiérarchie. Nous cherchons à savoir s'il existe un concept descendant du concept $\mathcal{C}_j$ qui correspond plus à cet objet. Nous utilisons pour cela la fonction $match(C, O)$, définie précédemment, sur l'ensemble des descendants directs du concept attribué et leurs notes sont comparées. Si une note dépasse celle du concept attribué, celui-ci est changé, sinon le concept attribué reste le même et l'algorithme termine. L'algorithme se termine une fois que le concept attribué n'a plus de descendant (le concept attribué est le plus spécialisé) ou que la note n'est maximisée par aucun des concepts fils (les contraintes des concepts fils ne permettent pas de les différencier par rapport au concept attribué).

Pour favoriser les concepts plus en profondeur dans la hiérarchie, donc plus raffinés, la note retournée est pondérée par la fonction $match(C, O)$ par la profondeur du concept C dans la hiérarchie. La profondeur d'un concept dans la hiérarchie est retournée par la fonction $depth(\mathcal{C}_i)$.

L'algorithme de raffinement est défini précisément dans l'algorithme 1. La complexité de l'algorithme est linéaire en nombre de nœuds dans le graphe.

Algorithme 1: Algorithme de raffinement : $refine(O)$

soit $\mathcal{C}_i \in \Theta$ le concept attribué à O, **tant que** $\exists \mathcal{C}_j$ *tels que* $\mathcal{C}_j \preceq^1_\Theta \mathcal{C}_i$ **faire**
  pour tous les $\mathcal{C}_j$ **faire**
    si $depth(\mathcal{C}_j) \times match(\mathcal{C}_j, O) > depth(\mathcal{C}_i) \times match(\mathcal{C}_i, O)$ **alors**
      $\mathcal{C}_i := \mathcal{C}_j$
  si $\mathcal{C}_i$ *n'a pas changé* **alors**
    Terminer

4 Expérimentations

Le modèle présenté précédemment a été testé sur des images de télédétection à très haute résolution spatiale : 2.4m par pixel pour l'image multi-spectrale, 60cm par pixel pour l'image panchromatique. Ces deux images sont fusionnées pour obtenir une image couleur d'une résolution de 60cm par pixel. Nous nous intéressons plus particulièrement à des images de paysages urbains.

4.1 Description du jeu de données

Nous choisissons de nous concentrer sur trois zones caractéristiques des paysages urbains pour tester l'intérêt de l'utilisation des relations spatiales sur la détection de pavillons. La première est une image de 500x500 pixels prise à Saint-Orens, à Toulouse, caractéristique d'une zone résidentielle pavillonnaire. La seconde est une image de 400x400 pixels prise près de Rangueil, à Toulouse, représentant une zone résidentielle rurale. La troisième est une image du quartier des Quinze, à Strasbourg, de 900x900 pixels, représentant une zone résidentielle mixte. Ces trois images ont été acquises par le satellite QuickBird.

4.2 Protocole de tests

Les images sont segmentées à l'aide d'approches simples. Le choix des paramètres de segmentation est laissé à l'expertise du géographe. Nous souhaitons vérifier que l'utilisation de connaissances spatiales permet l'obtention de meilleurs résultats qu'avec les simples connaissances spectrales et structurelles. Pour cela, l'expert fournit deux bases de connaissances. La première définit les concepts à partir de contraintes structurelles et spectrales. La seconde ajoute des contraintes spatiales (adjacence et distance) à l'ensemble des contraintes précédentes. Chaque image est traitée avec chacune de ces bases de connaissances. Les résultats sont comparés à une vérité terrain obtenue sur une base de données géographique. Un exemple d'image et de vérité terrain associée sont donnés dans les figures 2(a) et 2(b).

4.3 Résultats et discussion

Pour exprimer les résultats de façon chiffré, nous utilisons la précision, le rappel et la F-mesure. Les résultats pour l'image de Saint-Orens (caractéristique d'une zone urbaine pavillonnaire) sont récapitulés dans le tableau 1 et dans les figures 3, 4, 5 et 6.

Image brute	Précision	Rappel	F-Mesure
Sans relations spatiales	0.73	0.55	0.63
Avec relations spatiales	0.69	0.59	0.64
Avec relations spatiales sur la segmentation guidée	0.73	0.60	0.66

TAB. 1 – *Résultats pour l'image de Saint-Orens, exprimés par rapport à la vérité terrain (image originale et vérité terrain données dans la figure figure 2, extraits de résultats dans les figures 3, 4, 5 et 6). La segmentation guidée fait référence à une image dont les éléments caractéristiques ont été segmentés manuellement par l'expert.*

(a) Image originale (b) Vérité terrain bâtiment

FIG. 2 – *Extrait du jeu de données utilisé. Image originale de Saint-Orens (2(a)) et vérité terrain associée aux bâtiments (2(b)).*

Les figures 3(c), 4(c), 5(c) et 6(c) montrent les résultats de l'algorithme sans utilisation de relations spatiales. La comparaison avec la vérité terrain montre que les pavillons sont correctement extraits dans de nombreux cas, mais le rappel n'est pas élevé. De plus, les contours des pavillons ne sont pas réguliers, ce qui induit une erreur supplémentaire.

Les figures 3(d), 4(d), 5(d) et 6(d) montrent les résultats de l'algorithme avec utilisation de relations spatiales. À nouveau, les contours ne sont pas réguliers. Ceci est dû à l'imprécision de l'algorithme de segmentation face à la résolution d'une image THRS. Pour certains cas, figure 5(d), l'utilisation des relations spatiales introduit du bruit qui

dégrade les résultats en y ajoutant des faux positifs, des objets sont détectés comme des pavillons alors qu'ils n'en sont pas. Cependant, l'utilisation des relations spatiales permet de détecter certains pavillons qui ne l'étaient pas sans. Deux exemples sont donnés dans les figures 3 et 4.

(a) Image originale (b) Vérité terrain (c) Sans utilisation de relations spatiales (d) Avec utilisation de relations spatiales

FIG. 3 – *Extrait des résultats. Dans l'image 3(c) la troisième maison alignée n'est pas correctement détectée. Avec l'utilisation des relations spatiales, cette maison est bien identifiée.*

(a) Image originale (b) Vérité terrain (c) Sans utilisation de relations spatiales (d) Avec utilisation de relations spatiales

FIG. 4 – *Extrait des résultats. Dans l'image 4(c) la maison n'est pas correctement détectée. Avec l'utilisation des relations spatiales, cette maison est mieux identifiée.*

Ces résultats montrent l'importance d'une bonne détection des objets d'intérêt. Les connaissances utilisées, particulièrement les connaissances spatiales et structurelles, décrivent des concepts ayant une réalité sémantique. Or les algorithmes de segmentations usuels ne permettent pas une détection efficace de l'ensemble des concepts. L'image est soit sur-segmentée, soit sous-segmentée, les objets extraits correspondent à un sous-ensemble (respectivement, un sur-ensemble) d'un concept donné, ce qui fausse le calcul de correspondance entre un objet et ce concept.

L'apparition de faux-positifs montre que le calcul de correspondance doit être tolérant à l'imprécision, pour pouvoir gérer le bruit inhérent à la résolution des images THRS. En se basant sur des notations ensemblistes strictes, cette imprécision ne peut être traitée et vient parasiter les résultats.

(a) Image originale (b) Vérité terrain (c) Sans utilisation de relations spatiales (d) Avec utilisation de relations spatiales

FIG. 5 – *Extrait des résultats. Dans l'image 5(c) la maison est correctement détectée. Avec l'utilisation des relations spatiales, nous pouvons noter l'apparition de faux positifs (ici le jardin). Dans ce cas, les relations spatiales apportent du bruit à l'analyse.*

(a) Image originale (b) Vérité terrain (c) Sans utilisation de relations spatiales (d) Avec utilisation de relations spatiales

FIG. 6 – *Extrait des résultats. Dans l'image 6(c) le pavillon n'est pas détecté. Dans l'image 6(d) le pavillon est détecté mais est aggloméré avec un autre pavillon.*

5 Conclusion et perspectives

Cet article présente une méthodologie pour l'analyse d'image à l'aide de connaissances expertes, et plus particulièrement des relations spatiales entre objets d'intérêt. Un certain nombre de limites ont été mises en avant au cours de ces travaux. Premièrement, le processus de segmentation doit permettre l'extraction d'objets ayant une réalité sémantique forte. En effet, la base de connaissances décrit des concepts réels. Pour pouvoir être utilisés efficacement, les objets extraits de l'image doivent correspondre à ces concepts. Or les algorithmes de segmentations usuels ne permettent pas un découpage correct de l'image. Deuxièmement, sachant qu'une image contient une part non négligeable de bruit, l'algorithme d'inférence de concept doit être tolérant à l'imprécision. Cette imprécision peut être modélisée à l'aide de la théorie des ensembles flous (Zadeh (1965)). Ces problèmes doivent être corrigés avant de pouvoir utiliser la méthode proposée sur des jeux de données plus conséquents. Ceci est particulièrement vrai pour les images de télédétection.

Dans le cadre de la télédétection, une perspective envisageable pour pallier au problème de segmentation est l'utilisation de détecteurs éprouvés pour l'extraction de

concepts caractéristiques du paysage urbain (bâtiments, routes, ...). De plus, nous envisageons d'utiliser les connaissances sur les relations spatiales entre objets d'intérêt pour construire automatiquement des objets composites (îlots, quartiers, ...). Une fois construits, ces objets permettraient l'ajout d'informations spatiales de haut niveau (appartenance, composition, ...).

Références

Bilenko, M., S. Basu, et R. J. Mooney (2004). Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of 21st International Conference on Machine Learning (ICML-2004)*, Banff, Canada, pp. 81–88.

Bittner, T. et S. Winter (1999). On ontology in image analysis. *International Workshop On Integrated Spatial Databases ISD'99 1737*, pp. 168–191.

Durand, N., S. Derivaux, G. Forestier, C. Wemmert, P. Gancarski, O. Boussaid D, et A. Puissant (2007). Ontology-based object recognition for remote sensing image interpretation. In *IEEE International Conference on Tools with Artificial Intelligence*, Patras, Greece, pp. 472–479.

Fonseca, F., M. Egenhofer, P. Agouris, et G. Camara (2002). Using ontologies for integrated geographic information systems. *Transactions in Geographic Information Systems 6*(3), pp. 231–257.

Forestier, G., C. Wemmert, et P. Gancarski (2008). On combining unsupervised classification and ontology knowledge. In *IEEE Geoscience and Remote Sensing Symposium*, Volume 4, Boston, Massachusetts, pp. 395 – 398.

Freksa, C. (1992). Using orientation information for qualitative spatial reasoning. In A. Frank, I. Campari, et U. Formentini (Eds.), *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, Volume 639 of *Lecture Notes in Computer Science*, pp. 162–178. Springer Berlin / Heidelberg.

Gruber, T. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies 43*(5/6), pp. 907–928.

Inglada, J. et J. Michel (2009). Qualitative spatial reasoning for high-resolution remote sensing image analysis. *IEEE Transaction on Geoscience and remote sensing 47*, pp. 599–612.

Jin, X. et C. H. Davis (2005). Automated building extraction from high-resolution satellite imagery in urban areas using structural, contextual, and spectral information. *EURASIP Journal on Applied Signal Processing* (14), pp. 2196–2206.

Kumar, N. et K. Kummamuru (2008). Semisupervised clustering with metric learning using relative comparisons. *Knowledge and Data Engineering, IEEE Transactions on 20*(4), pp. 496–503.

Kutz, O., C. Lutz, F. Wolter, et M. Zakharyaschev (2004). E-connections of abstract description systems. *Artificial Intelligence 156*(1), pp. 1–73.

Maillot, N. E. et M. Thonnat (2008). Ontology based complex object recognition. *Image Vision Computing 26*(1), pp. 102–113.

Mezaris, V., I. Kompatsiaris, et M. G. Strintzis (2004). Region-based image retrieval using an object ontology and relevance feedback. *EURASIP Journal on Advances in Signal Processing 2004*(1), pp. 886–901.

Moratz, R., J. Renz, et D. Wolter (2000). Qualitative spatial reasoning about line segments. In *ECAI 2000. Proceedings of the 14th European Conference on Artifical Intelligence*, pp. 234–238. IOS Press.

Papadias, D. et M. J. Egenhofer (1997). Algorithms for hierarchical spatial reasoning. *GeoInformatica 1*, pp. 251–273.

Pedrycz, W. (2004). Fuzzy clustering with a knowledge-based guidance. *Pattern Recognition Letters 25*(4), pp. 469–480.

Peteri, R., J. Celle, et T. Ranchin (2003). Detection and extraction of road networks from high resolution satellite mages. In *IEEE International Conference on Image Processing*, pp. 301–304.

Renz, J. et B. Nebel (1999). On the complexity of qualitative spatial reasoning : A maximal tractable fragment of the region connection calculus. *Artificial Intelligence 108*(1-2), pp. 69–123.

Stoica, R., X. Descombes, et J. Zerubia (2001). Road extraction in remote sensed images using stochastic geometry framework. *AIP Conference Proceedings 568*(1), pp. 531–542.

Vanegas, M., I. Bloch, et J. Inglada (2009). Fuzzy spatial relations for high resolution remote sensing image analysis : The case of "to go across". In *Geoscience and Remote Sensing Symposium,2009 IEEE International,IGARSS 2009*, Volume 4, pp. 773–776.

Vanegas, M., I. Bloch, et J. Inglada (2010). Detection of aligned objects for high resolution image understanding. In *Geoscience and Remote Sensing Symposium,2010 IEEE International,IGARSS 2010*.

Wagstaff, K., C. Cardie, S. Rogers, et S. Schroedl (2001). Constrained k-means clustering with background knowledge. In *Proceeding of the International Conference of Machine Learning*, pp. 577–584. Morgan Kaufmann.

Zadeh, L. (1965). Fuzzy sets. *Information and Control 8*(3), pp. 338–353.

Zhao, H., J. Kumagai, M. Nakagawa, et R. Shibasaki (2002). Semi-automatic road extraction from high-resolution satellite image. In *ISPRS Symposium on Photogrammetry and Computer Vision*, pp. 406.

Summary

The technological evolution gives us access to images with a high level of detail. Segmentation algorithms allow us to extract objects of interest from these images. Using an expert knowledge formalization helps us to classify these objects. Nowadays, only a few methods are taking advantage of the spatial knowledge to analyze remote sensing images. In this article we propose a method for formalizing this kind of expert knowledge and using it for analysis of remote sensing images. An analysis of the limits of this method will open the way for new research perspectives.

Vers davantage de flexibilité et d'expressivité dans les hiérarchies contextuelles des entrepôts de données

Yoann Pitarch*, Cécile Favre**
Anne Laurent***, Pascal Poncelet***

* Dept. d'Informatique, Aalborg, Danemark
ypitarch@cs.aau.dk
**ERIC, Lyon, France
cecile.favre@univ-lyon2.fr
***LIRMM, Montpellier, France
{laurent,poncelet}@lirmm

Résumé. Les entrepôts de données sont aujourd'hui couramment utilisés pour analyser efficacement des volumes importants de données hétérogènes. Une des clés de ce succès est l'utilisation des hiérarchies qui permettent de considérer les données à différents niveaux de granularité. Au cours de travaux précédents, nous avons montré que ce processus de généralisation n'introduisait que très peu de connaissances expertes conduisant ainsi à des analyses erronées. Nous avons alors proposé une nouvelle catégorie de hiérarchie, les hiérarchies contextuelles, pour répondre à cette limite. Malheureusement, les connaissances expertes introduites devaient être spécifiées de façon rigide et ne pouvaient donc refléter toute leur réelle complexité. Dans cet article, nous étendons ces hiérarchies et les mécanismes associés pour accroître grandement leur flexibilité et leur expressivité et utilisons pour cela une approche basée sur la logique floue. Ce qui est présenté ici constitue un travail préliminaire visant à montrer l'intérêt et la faisabilité de l'idée.

1 Introduction

Les entrepôts de données (Inmon, 1996) visent à consolider des données à des fins d'analyse, en adoptant une modélisation dite *multidimensionnelle* au travers de laquelle des faits sont analysés au moyen d'indicateurs (les mesures) selon différents axes d'analyse (les dimensions). En s'appuyant sur des mécanismes d'agrégation, les outils OLAP (On Line Analytical Process) permettent de naviguer aisément le long des hiérarchies des dimensions (Malinowski et Zimányi, 2004). L'intérêt des décideurs pour ce type d'application s'avère être croissant, et ce quel que soit leur domaine d'application : marketing, suivi de production, médical... Dans cet article, nous nous proposons d'illustrer nos propos par rapport à ce dernier domaine puisque ce travail avait débuté dans le cadre d'un projet ANR [1].

A l'heure où l'accroissement du volume de données incite à proposer de nouvelles solutions de gérance de l'informatique telles que le *cloud computing* où les données s'éloignent des utilisateurs et induisent de nombreuses problématiques parmi lesquelles la sécurité, le souci des utilisateurs s'est parallèlement renforcé avec l'intensification de travaux, dans le contexte

1. Projet ANR MIDAS (ANR-07-MDCO-008)

des entrepôts de données et de l'analyse OLAP entre autres, sur la personnalisation, la recommandation et l'intégration de connaissances utilisateurs. Ainsi, dans ce travail, nous mettons en avant l'importance des connaissances et de leur expression dans une phase de structuration des données qui sont massives et complexes. Cette phase de structuration vient en amont de la phase de fouille de données qui pourra tirer partie de deux constats : 1) l'entrepôt intègre par définition des sources de données variées et constitue donc une source *propre* de données de qualité ; 2) l'intégration de connaissances que nous proposons permet une réutilisation de ces connaissances lors du processus de fouille.

Considérons le cas réel d'un entrepôt de données médicales rassemblant les paramètres vitaux (e.g., la tension artérielle...) des patients d'un service de réanimation. Afin de réaliser un suivi efficace des patients, un médecin souhaiterait par exemple connaître ceux qui ont eu une tension artérielle basse au cours de la nuit. Pouvoir formuler ce type de requête suppose l'existence d'une hiérarchie sur la tension artérielle dont le premier niveau d'agrégation serait une catégorisation de la tension artérielle (e.g., basse, normale, élevée), hiérarchie qui pourra être exploitée durant un processus de navigation dans les données. Toutefois, cette catégorisation est délicate car elle dépend à la fois de la tension artérielle mesurée mais aussi de certaines caractéristiques physiologiques (âge du patient, fumeur ou non, ...). Dès lors, une même tension peut être généralisée différemment selon le contexte d'analyse considéré. Par exemple, 13 est une tension élevée chez un nourrisson alors qu'il s'agit d'une tension normale chez un adulte. Ainsi, pour permettre cette généralisation, nous avons été amenés dans des travaux précédents (Pitarch et al., 2010a,b) à proposer un nouveau type de hiérarchies qualifiées de *contextuelles*.

La définition de ces hiérarchies permet donc d'exprimer la connaissance métier des utilisateurs pour ensuite les exploiter lors du processus de navigation et d'analyse. Il s'avère aujourd'hui que nous avons besoin d'aller au-delà en matière d'expressivité de ces connaissances en intégrant le fait que ces connaissances peuvent être plus ou moins exactes et que le contenu même de ces connaissances peut être plus ou moins précis.

En matière d'expressivité de connaissances, l'approche floue trouve un réel intérêt (Dubois, 2011). Différents travaux se sont d'ailleurs intéressés à l'intégration de la logique floue dans les entrepôts de données pour augmenter la richesse des hiérarchies entre autres (Fasel et Shahzad, 2010; Perez et al., 2007). Ainsi, dans ce travail, nous nous proposons de nous intéresser à l'intégration de la logique floue dans le cadre des hiérarchies contextuelles, franchissant un pas supplémentaire dans l'expressivité des hiérarchies, assurant une analyse encore plus pertinente lors du processus de navigation. Les résultats présentés ici constituent un travail préliminaire visant à montrer l'intérêt et la faisabilité de cette idée.

La suite de ce papier est organisée de la façon suivante. La section 2 expose notre cas d'étude sur des données médicales qui permettra d'illustrer par la suite le problème posé et la solution apportée. Dans la section 3, nous revenons sur le principe et la formalisation des hiérarchies contextuelles. Nous proposons alors dans la section 4 l'adaptation de notre solution au contexte du flou. Puis nous discutons de l'implémentation de cette solution dans la section 5. Enfin, nous concluons et indiquons les perspectives de ce travail dans la section 6.

2 Cas d'étude

Afin d'illustrer les limites des hiérarchies contextuelles actuelles et montrer comment nous les pallions dans cet article, considérons le scénario médical suivant. Pour chaque patient, le

système enregistre son âge, sa consommation moyenne quotidienne de tabac, son traitement médical éventuel et sa tension à un instant donné. Nous souhaitons déterminer si la tension doit être généralisée en *faible*, *normale* ou *élevée*. En outre, l'attribut associé à l'âge (resp. à la consommation de tabac) peut être considéré à deux niveaux de granularité différents : l'âge (resp. la quantité exacte de cigarettes fumées) et la catégorie de l'âge (resp. la catégorie de consommation de tabac). Le tableau 1 présente les informations associées au patient $P1$ et la figure 1(a) indique les extraits des hiérarchies utiles dans ce cas d'étude. Comme argumenté dans Pitarch et al. (2010b), la généralisation d'une tension artérielle est contextuelle par nature puisque de nombreux paramètres peuvent influer sur le résultat de cette généralisation. La figure 1(b) répertorie quelques connaissances expertes pour guider le processus de généralisation. Par exemple, $R4$ indique qu'une tension supérieure à 14 est élevée pour un fumeur régulier.

En analysant conjointement le tableau 1 et les figures 1(a)-1(b), plusieurs observations peuvent être faites :

1. Les connaissances ne sont pas nécessairement exprimées sur l'ensemble des attributs impactant la généralisation d'une tension. Par exemple, $R4$ spécifie des conditions sur les attributs $CatConsoTabac$ et $Tension$ (TA) alors que $R5$ spécifie des conditions sur les attributs $Traitement$ et $Tension$ (TA).
2. Les connaissances $R5$ et $R6$ sont toutes deux adaptées donc applicables à $P1$ mais diffèrent quant au résultat de la généralisation de la tension.
3. La connaissance $R5$ est plus précise que la connaissance $R6$, *i.e.*, elle définit un plus grand nombre de conditions. Il apparaît donc plus raisonnable de généraliser en prenant en compte $R5$.
4. Bien que ne s'appliquant pas strictement à $P1$, les conditions des connaissances $R2$ et $R3$ sont *presque* réunies pour le patient $P1$. Typiquement, il aurait fallu que $P1$ consomme seulement une seule cigarette supplémentaire par jour pour que la connaissance $R3$ puisse être appliquée.

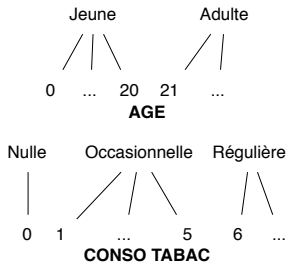
IdP	Age	CatAge	ConsoTabac	CatConsoTabac	Traitement	Tension (TA)	CatTension
P1	22	Adulte	5	Occasionnelle	Hypotenseur	13	?
...	...	...	...	...	...	...	...

TAB. 1 – *Patient exemple*

Nous montrons dans la prochaine section que les hiérarchies contextuelles, telles que nous les avons définies précédemment, n'intègrent que partiellement ces remarques et proposons ensuite une solution à ce problème.

3 Les hiérarchies contextuelles

Nous rappelons d'abord les définitions associées aux hiérarchies contextuelles classiques puis les solutions proposées pour les mettre en œuvre. Cette section s'achève par une discussion autour des limites de ce modèle en matière de flexibilité et d'expressivité.



(a) Hiérarchies

IdC	Connaissance
R1	Jeune + Nulle ou occasionnelle + TA $\in [8;12] \rightarrow$ Normale
R2	Jeune + Nulle ou occasionnelle + $TA > 12 \rightarrow$ Elevée
R3	Régulière + $TA \in [9;14] \rightarrow$ Normale
R4	Régulière + $TA > 14 \rightarrow$ Elevée
R5	Hypotenseur + $TA > 12 \rightarrow$ Elevée
R6	$TA \in [8;13] \rightarrow$ Normale
...	...

(b) Connaissances expertes

FIG. 1 – *Connaissances utiles pour la généralisation contextuelle*

3.1 Définitions

Les hiérarchies contextuelles ont été proposées dans le cadre des entrepôts de données. Le formalisme introduit redéfinit donc les concepts bien connus de dimensions, tables de faits, attributs de dimensions et mesures. Par souci de simplicité et sans perte de généralité, les définitions présentées dans cette section s'abstraient de ce cadre formel et supposent simplement l'existence d'un ensemble d'identifiants déterminant fonctionnellement un ensemble d'attributs.

Nous notons $Id = \{id_1, \ldots, id_n\}$ l'ensemble des n identifiants et $\mathcal{A} = \{A_1, \ldots, A_t\}$ l'ensemble des t attributs. Nous supposons que A_t est l'attribut à généraliser contextuellement. Pour chaque attribut A_i, $i = 1 \ldots t$, son domaine de définition est noté $Dom(A_i)$. Chaque attribut A_i est équipé d'une hiérarchie et peut donc être observé selon plusieurs granularités $A_i = A_i^0, \ldots, A_i^{M_i}$. De telles hiérarchies sont dites *simples*. Par convention, A_i^0 désigne la granularité la plus fine et $A_i^{M_i}$ désigne la granularité la plus élevée. Nous notons $a_i^j \in Dom(A_i^j)$ si la valeur a_i^j représente une instance du niveau A_i^j. Nous notons $\mathcal{A}^*$ l'ensemble $\mathcal{A}^* = \{A_1^0, A_1^1, \ldots, A_t^{M_t}\}$. Le schéma de T, la table relationnelle manipulée, est donc $T = (Id, A_1^0, A_1^1, \ldots, A_t^{M_t})$ tel que $Id \rightarrow A_1^0 \times A_1^1 \times \ldots \times, A_t^{M_t}$. Un n-uplet t de T est de la forme $t = (id, a_1^0, a_1^1, \ldots, a_t^0, \ldots, a_t^{M_t})$ tel que a_t^0 est la valeur à généraliser contextuellement et $a_t^1, \ldots, a_t^{M_t}$ en sont les généralisations contextuelles. Le tableau 1 présente un extrait de la table relationnelle utilisée dans notre cas d'étude.

Pour un attribut donné, le passage entre un niveau et sa granularité supérieure directe est permis par les *chemins d'agrégation* définis comme suit.

Définition 1 *(Chemin d'agrégation) Soient A_i^j et A_i^{j+1}, $j = 0, \ldots, M_i - 1$, deux niveaux de granularité de l'attributs A_i. Un* chemin d'agrégation*, noté G, entre A_i^j et A_i^{j+1}, est défini par $G = A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$ avec $\mathrm{C} = \{A_i^j, \ldots\}$ tel que (1) A_i^j est appelé* attribut source*, (2) $A_i^{j+1} \notin \mathrm{C}$ et (3) $\mathrm{C} \subseteq \mathcal{A}$ et A_i^{j+1} dépend fonctionnellement de* C.

Définition 2 *(Chemin d'agrégation contextuel et classique) Soit* $G = A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$ *un chemin d'agrégation.* G *est appelé un* chemin d'agrégation contextuel *si* $|\mathrm{C}| > 1$. *Sinon* G *est appelé un* chemin d'agrégation classique *(ou simplement un chemin d'agrégation quand cela est sans ambiguïté).*

Définition 3 *(Attribut contextualisant et contextualisé) Soit* $G = A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$ *un chemin d'agrégation contextuel. L'attribut* A_i^{j+1} *est dit* contextualisé *par les attributs de* C. *Les attributs de* C *sont appelés attributs* contextualisants.

Définition 4 *(Contexte) Soit* $G = A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$ *un chemin d'agrégation contextuel. Le triplet* $c = (A_i^j, \mathrm{C}, A_i^{j+1})$ *est appelé le* contexte *de l'attribut* A_i^{j+1} *et nous notons* $\mathcal{C} = \{c_1, \ldots, c_k\}$ *l'ensemble des contextes.*

Définition 5 *(Instance d'un chemin d'agrégation) Soit* $G = A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$ *un chemin d'agrégation. Une instance de* G *, notée* $g = a \overset{\mathrm{IC}}{\longmapsto} a'$*, est telle que :*

1. $a \in Dom(A_i^j)$ *et* $a' \in Dom(A_i^{j+1})$
2. $\mathrm{IC} = \{(A_l^m, \alpha) \mid A_l^m \in \mathrm{C}$ *et* $\alpha \subseteq Dom(A_l^m)\}$
3. $\exists (A_l^m, \alpha)$ *tel que* $A_i^j = A_l^m$ *et* $a \in \alpha$
4. $\nexists A_l^m$ *tel que* $A_l^m \in \mathrm{C}$ *et* $(A_l^m, \alpha) \notin \mathrm{IC}$

Définition 6 *(Instance de contexte) Soit* $g = a \overset{\mathrm{IC}}{\longmapsto} a'$ *une instance du chemin d'agrégation* $A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$*. La paire* $c^l = (\mathrm{IC}, a')$ *est appelée une* instance *du contexte* c *de* A_i^{j+1} *et nous notons* $\mathcal{IC} = \{c_1^1, \ldots, c_k^{l_k}\}$ *l'ensemble des instances de contextes.*

Exemple 1 *Considérons que la généralisation du niveau* Tension *au niveau* CatTension *dépend uniquement des attributs* CatConsoTabac *et* Tension*. Dès lors le chemin d'agrégation contextuel associé est noté* $G = Tension \overset{\mathrm{C}}{\mapsto} CatTension$ *avec* $\mathrm{C} = \{CatConsoTabac, Tension\}$ *et le contexte correspondant est noté* $c = (Tension, \mathrm{C}, CatTension)$*. Dans ce contexte,* $CatConsoTabac$ *et* $Tension$ *sont les attributs contextualisants et* $CatTension$ *est l'attribut contextualisé. La connaissance associée à* $R4$ *est représentée par l'instance de contexte* $c^l = (\mathrm{IC}, \mathrm{Elevée})$ *avec* $\mathrm{IC} = \{(CatConsoTabac, \{\mathrm{Régulière}\}), (Tension, \{x > 14\})\}$.

3.2 Mise en œuvre

3.2.1 Stockage de la connaissance

Dans Pitarch et al. (2010b), nous proposons de représenter les connaissances expertes (structures et instances de contexte) via une base de données relationnelles externe, notée $\mathcal{B}_{Know.}$ afin de garantir la généricité du stockage des différents contextes présents dans l'entrepôt. Cette base de données est seulement composée de deux relations que nous décrivons brièvement.

La relation `MTC` (`M`éta-`T`able des `C`onnaissances) permet de stocker les contextes, *i.e.*, quels sont les attributs contextualisants et contextualisés pour chaque contexte, et possède la structure suivante :

- **Contexte** désigne l'identifiant du contexte ;
- **Attribut** désigne un attribut intervenant dans le contexte ;
- **Type** définit le rôle joué par *Attribut* dans le contexte. *Type* vaut "*Contexte*" si *Attribut* est un attribut *contextualisant* dans ce contexte et vaut "*Résultat*" si *Attribut* est l'attribut contextualisé.

La relation `TC` (`T`able des `C`onnaissances) permet de stocker quelles sont les instances des différents contextes, *i.e.*, la connaissance experte, et possède la structure suivante :

- **Contexte** désigne l'identifiant du contexte concerné ;
- **Instance_Contexte** identifie l'instance du contexte ;
- **Attribut** désigne un attribut intervenant dans le contexte ;
- **Valeur** représente l'ensemble de valeurs d'*Attribut* concerné dans l'instance en question.

3.2.2 Généralisation contextuelle

Par manque de place, le fonctionnement de `ROLL_UP_CTX`, l'opérateur de généralisation contextuelle, est présenté succinctement [2]. D'abord, il faut identifier dans la table `MTC` quel est le contexte en jeu et quels sont les attributs contextualisants. Une recherche dans la table `TC` permet ensuite d'identifier l'unique instance concernée dans cette généralisation. Dès lors, il suffit de retourner la valeur de l'attribut contextualisé correspondant à cette instance.

3.3 Discussion

Les hiérarchies contextuelles permettent la prise en compte de connaissances expertes dans le processus de généralisation et représentent donc une solution originale et intéressante dans de nombreux domaines d'applications où l'on doit donner du sens à des attributs numériques. Malgré tout, ces hiérarchies contextuelles souffrent encore de quelques limites. Précisément, dans le modèle actuel, il ne doit exister qu'une (contrainte d'existence) et une seule (contrainte d'unicité) instance de contexte pour une généralisation contextuelle. Nous détaillons ci-dessous les arguments qui motivent de relâcher quelque peu ces contraintes.

Tout d'abord, la contrainte d'existence implique que l'ensemble des connaissances soit complet au sens des instances de T. Sans cette garantie, la tension artérielle d'un patient pourrait ne pas être généralisée par exemple. Pourtant, malgré l'existence d'outils pour vérifier la complétion d'un ensemble de règles, ceux-ci montrent leur limite quand il s'agit de traiter de nombreux attributs (Ligęza (2006)). Ce point pose problème dans la mesure où il est impossible de contrôler le nombre d'attributs pouvant impacter sur une généralisation.

Ensuite, la contrainte d'unicité garantit certes la consistance de la connaissance experte mais est en pratique très difficile à obtenir. Plusieurs arguments étayent cette affirmation. D'abord, bien que les attributs contextualisants représentent l'ensemble des facteurs pouvant potentiellement influer sur une généralisation, rien ne garantit que l'impact conjoint de ces attributs ait été étudié. Par exemple, il n'existe pas de connaissances portant sur tous les attributs contextualisants dans notre cas d'étude. Ensuite, l'appartenance d'un n-uplet à une instance de contexte est parfois délicate à définir strictement. Par exemple, comme nous l'avons remar-

2. Les lecteurs intéressés sont invités à se référer à Pitarch (2011) pour plus de détails.

qué plus tôt, les conditions de $R3$ sont presque réunies pour le patient $P1$. Pour pallier cette rigidité, l'utilisation des hiérarchies floues apparaît très prometteuse.

Relâcher les contraintes d'existence et d'unicité augmenterait considérablement la flexibilité et l'expressivité des hiérarchies contextuelles mais soulève de nouvelles problématiques quant au stockage de la connaissance et au processus de généralisation. La prochaine section détaille ces problématiques et décrit les modifications apportées au modèle.

4 Hiérarchies contextuelles généralisées

Pour pallier les limites précédemment mentionnées, nous étendons le modèle actuel de deux manières. D'abord, nous introduisons le concept d'*instance de contexte généralisée* qui relâche la contrainte d'unicité, autorise les instances de contexte à ne pas inclure systématiquement l'ensemble des attributs contextualisants et permet la définition d'un ordre partiel au sein des instances d'un contexte donné. Nous étudions ensuite les conséquences de l'*utilisation des hiérarchies floues* pour les attributs contextualisants et définissons alors un degré d'appartenance d'un n-uplet à une instance de contexte.

4.1 Instance de contexte généralisée

Nous l'avons vu, garantir la complétude au sens des instances de la base peut se révéler délicat. En outre, cette complétude peut très bien être assurée à un certains temps t et ne plus l'être au temps $t+1$ lorsque de nouvelles instances sont insérées dans la table. Garantir une complétude universelle est alors préférable. Une solution pour garantir cette complétude universelle est d'autoriser plusieurs degrés de précision dans les instances de contexte. Par exemple, l'insertion de connaissances très générales, telles que $R6$, garantirait qu'une tension sera toujours généralisée. Ces connaissances pourraient être complétées par des connaissances bien plus précises telles que $R1$ et $R2$. Intuitivement, le processus de généralisation favoriserait alors l'instance adéquate la plus précise. Nous formalisons ci-dessous ce concept d'instance de contexte généralisée et introduisons une relation d'ordre partiel entre les instances.

Définition 7 *(Instance généralisée d'un chemin d'agrégation contextuel) Soit* $G = A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$ *un chemin d'agrégation contextuel. Une instance généralisée de* G *, notée* $g = a \overset{\mathrm{IC}}{\longmapsto} a'$*, est telle que :*

1. $a \in Dom(A_i^j)$ *et* $a' \in Dom(A_i^{j+1})$
2. $\mathrm{IC} = \{(A_l^m, \alpha) \mid A_l^m \in \mathrm{C} \wedge \alpha \subseteq Dom(A_l^m)\}$
3. $\exists (A_l^m, \alpha) | A_i^j = A_l^m \wedge a \in \alpha$

Définition 8 *(Instance généralisée de contexte) Soit* $g = a \overset{\mathrm{IC}}{\longmapsto} a'$ *une instance généralisée du chemin d'agrégation* $A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$*. La paire* $c^l = (\mathrm{IC}, a')$ *est appelée une* instance *généralisée du contexte* c *de* A_i^{j+1} *et nous notons* $\mathcal{IC} = \{c_1^1, \ldots, c_k^{l_k}\}$ *l'ensemble des instances généralisées de contextes.*

Définition 9 *(Précision d'une instance généralisée de contexte) Soit* $c^l = (\mathrm{IC}, a)$ *une instance généralisée du contexte* $c = (A_i^j, \mathrm{C}, A_i^{j+1})$. *La précision de* c^l, *notée* $Prec(c^l)$, *est définie comme* $Prec(c^l) = |\mathrm{IC}|$.

Nous notons $Attrib(g)$ les attributs contextualisants présents dans l'instance g.

Exemple 2 $c^l = \big(\{(\textit{Tension}, [8; 13])\}, \textit{Normale}\big)$ *est l'instance généralisée du contexte* $c = (Tension, \{Tension, CatAge, CatConsoTabac, Traitement\}, CatTension)$ *qui représente la connaissance* $R6$. *De plus, nous avons* $Attrib(c^l) = \{Tension\}$ *et* $Prec(c^l) = 1$.

Remarquons que, dans le cadre formel initial, la précision d'une instance de contexte est toujours le nombre d'attributs contextualisants. Désormais, des instances pouvant être plus ou moins générales, il est naturel de définir un ordre partiel pour comparer le niveau de précision de deux instances.

Définition 10 *(Ordre partiel sur les instances d'un contexte) Soient* $c^l = (\mathrm{IC}, a)$ *et* $c^{l'} = (\mathrm{IC}', a')$ *deux instances du contexte* $c = (A_i^j, \mathrm{C}, A_i^{j+1})$. c^l *spécialise* $c^{l'}$ *(resp.* $c^{l'}$ *généralise* c^l*), noté* $c^l \prec c^{l'}$ *(resp.* $c^{l'} \succ c^l$*) , si :*

(1) $Attr(c^{l'}) \subseteq Attr(c^l)$ *;*

(2) $\alpha \subseteq \alpha'$ *avec* $(A_k^m, \alpha) \in \mathrm{IC}$, $(A_k^m, \alpha') \in \mathrm{IC}'$ *et* $A_k^m \in (Attr(c^{l'}) \cap Attr(c^l)) - \{A_i^j\}$.

Exemple 3 *Soient* $c^1 = \big(\{(\textit{Tension}, [8; 13])\}, \textit{Normale}\big)$, $c^2 = \big(\{(\textit{Tension}, [8; 13]), (\textit{CatAge}, \{Jeune, Adulte\})\}, \textit{Normale}\big)$ *et* $c^3 = \big(\{(\textit{Tension}, [9; 12]), (\textit{CatAge}, \{Jeune\})\}, \textit{Normale}\big)$ *trois instances de contexte. Nous avons* $c^1 \prec c^2$, $c^2 \prec c^3$ *et* $c^1 \prec c^3$ [3].

4.2 Prise en compte de hiérarchies floues

Introduits par Zadeh (1965), les sous-ensembles flous permettent de modéliser la représentation humaine des connaissances et améliorent ainsi les performances des systèmes de décision qui utilisent cette modélisation. Plus particulièrement, les hiérarchies floues présentées par Laurent (2002) permettent de modéliser l'appartenance d'un élément à plusieurs généralisations avec des degrés de confiance propres à chacune d'entre elles. L'utilisation de ces hiérarchies apparaît alors particulièrement adaptée pour représenter le fait qu'une instance *correspond presque* à un n-uplet.

Etant donné un ensemble de référence $Dom(A_i^j)$, avec $j = 1, \ldots, M_i$, un sous-ensemble flou A de $Dom(A_i^j)$ est alors défini par une fonction d'appartenance f_A qui associe à chaque élément a du niveau $Dom(A_i^0)$ le degré $f_A(a)$, compris entre 0 et 1, reflétant une gradualité dans son appartenance à A. Par exemple, $f_{CatAge=Adulte}(22) = 0,7$ indique que 22 appartient, à un degré de $0,7$ au sous-ensemble $\{Adulte\}$.

L'utilisation des hiérarchies floues nous contraint à redéfinir le contenu de la table T pour stocker les divers degrés de confiance. Un n-uplet $t = (id, a_1^0, a_1^1, \ldots, a_t^0, \ldots, a_t^{M_t})$ de T sera désormais stocké sous la forme $t' = (id', e_1^0, e_1^1, \ldots, e_t^0, \ldots, e_t^{M_t})$ tels que $id = id'$, $e_i^0 = a_i^0$ pour $i = 1, \ldots, t$ et $e_i^j = \{(\alpha_i^j, f_{\alpha_i^j}(e_i^0)) \mid \alpha_i^j \in Dom(A_i^j)$ *et* $f_{\alpha_i^j}(e_i^0) \neq 0\}$ pour $i = 1, \ldots, t-1$ et $j = 1, \ldots, M_i$. Le tableau 2 illustre comment les hiérarchies floues sur les

IdP	Age	CatAge	ConsoTabac	CatConsoTabac	Traitement	Tension	CatTension
P1	22	$f_{CatAge=Jeune}(22) = 0,25$ $f_{CatAge=Adulte}(22) = 0,7$	5	$f_{CCT=occ.}(5) = 0,3$ $f_{CCT=reg.}(5) = 0,6$	Hypotenseur	13	?

TAB. 2 – *Patient exemple avec intégration des hiérarchies floues*

attributs relatifs à l'âge et à la consommation de tabac sont intégrées dans la table relationnelle de notre cas d'étude.

Dès lors, un n-uplet flou de T peut être associé à plusieurs instances de contexte avec des degrés d'adéquation distincts. Nous nous intéressons maintenant à la définition de ce degré d'adéquation entre une instance et un n-uplet. Intuitivement, le principe est identique au cas strict : il s'agit de vérifier séparément l'adéquation du n-uplet à chaque condition d'une instance puis de les combiner.

Etudions d'abord comment calculer le degré d'adéquation d'un n-uplet à une condition d'une instance. Supposons que l'on souhaite calculer à quel point le patient $P1$ valide le critère $(SubCatAge, \{Jeune, Adulte\})$. Ici, il s'agit de mesurer le degré d'appartenance de l'élément 22 au sous-ensemble flou $\{Jeune, Adulte\}$ ou, autrement dit, à l'union des sous-ensembles flous $\{Jeune\}$ et $\{Adulte\}$. L'opérateur de t-conorme, noté $\perp$, est l'extension floue de l'opérateur d'union. Nous l'utilisons pour définir l'adéquation d'un n-uplet à un critère.

Définition 11 *(Adéquation d'une instance à une condition) Soient* $cond = (A_i^j, \alpha)$ *une condition (avec* $j = 0, \ldots, M_i$ *et* $\alpha = \{\alpha_1, \ldots, \alpha_k\}$*) et* $t' = (id', e_1^0, e_1^1, \ldots, e_t^0, \ldots, e_t^{M_t})$ *un n-uplet de* T*. L'adéquation de* t' *à* $cond$*, notée* $\mu_{unit}(t', cond)$*, est définie comme :*

$$\mu_{unit}(t', cond) = \perp\Big(f_{\alpha_1}(e_i^0), \ldots, f_{\alpha_k}(e_i^0)\Big)$$

Remarquons qu'il existe là aussi dans la littérature plusieurs fonctions pour définir la t-conorme. Une étude approfondie de la fonction la plus appropriée serait pertinente. Néanmoins, nous utilisons à nouveau la fonction proposée par Zadeh, à savoir la fonction max.

Exemple 4 *Soient* t *le n-uplet correspondant au patient* $P1$ *et* $cond = (CatAge, \{Jeune, Adulte\})$*, un critère. Nous avons* $\mu_{unit}(t, cond) = max(f_{CatAge=Jeune}; f_{CatAge=Adulte}) = 0,7$.

Etudions maintenant comment combiner ces adéquations locales pour calculer le degré d'adéquation globale d'un n-uplet à une instance de contexte, *i.e.*, à une conjonction de conditions. Supposons que l'on souhaite déterminer à quel point le patient $P1$ valide l'instance globale. Ici, il s'agit de mesurer le degré d'appartenance de $P1$ à toutes les conditions de $R1$ ou, autrement dit, à l'intersection des sous-ensembles flous représentés que sont chaque condition. L'opérateur de t-norme, noté $\top$, est l'extension floue de l'opérateur d'intersection. Nous l'utilisons pour définir l'adéquation d'un n-uplet à une instance.

Définition 12 *(Adéquation d'une instance à un n-uplet) Soient* $c^l = (\mathrm{IC}, a)$*, une instance du contexte* $c = (A_i^j, \mathrm{C}, A_i^{j+1})$ *et* $t' = (Id, e_1^0, e_1^1, \ldots, e_t^0, \ldots, e_t^{M_t})$*, un n-uplet de* T*. L'adéquation de* t' *à* c^l*, notée* $\mu(t', c^l)$*, est définie comme :*

$$\mu(t', c^l) = \top\Big(\mu_{unit}\big(t', (A_k^m, \alpha)\big) \text{ tel que } A_k^m \in Attrib(c^l)\Big)$$

3. Des investigations supplémentaires sont nécessaires pour prouver la transitivité de $\prec$ dans le cadre général.

Remarquons qu'il existe dans la littérature plusieurs fonctions pour définir la t-norme. Une étude approfondie de la fonction la plus appropriée serait pertinente. Néanmoins, nous utilisons ici la fonction proposée par Zadeh, la fonction min.

Exemple 5 *Soit t le n-uplet correspondant au patient* $P1$ *et* $c^l = \big(\{(\textit{Tension}, [8; 13]), (\textit{CatAge}, \{Jeune, Adulte\})\}, \textit{Normale}\big)$ *une instance de contexte. Nous avons :*

$$\begin{aligned}\mu(t, c^l) &= min\Big(\mu_{unit}\big(t, (\textit{Tension}, [8; 13])\big); \\ &\qquad \mu_{unit}\big(t, (\textit{CatAge}, \{Jeune, Adulte\})\big)\Big) \\ &= min(0,7\ ;\ 1) \\ &= 0,7\end{aligned}$$

5 Discussion autour de la mise en œuvre

Le cadre formel étant étendu, nous discutons maintenant ses implications sur deux aspects essentiels : la représentation des connaissances et la généralisation contextuelle.

5.1 Stockage

Un premier point très intéressant à soulever ici est que l'extension des hiérarchies contextuelles telle que définie dans la section précédente n'impacte pas sur le mode de stockage de la connaissance. En effet, dans la mesure où la définition d'un contexte reste inchangée, *i.e.*, l'existence d'attributs contextualisants et d'un attribut contextualisé est toujours valide, la structure de la relation `MTC` ne doit subir aucune modification. En outre, la solution proposée précédemment pour stocker les instances de contexte ne contraint pas à spécifier un ensemble de valeurs pour chaque attribut contextualisant. Dès lors, la relation `TC` peut stocker des instances généralisées de contexte sans aucune modification structurelle. Enfin, la présence de hiérarchies floues est propre à la table T considérée et n'a aucune incidence sur la base de données externe.

Cependant, il est important de garder en tête que, dans un contexte analytique, le nombre de généralisations à réaliser peut être très important. Il est donc nécessaire de stocker les connaissances de sorte à ce qu'elles soient efficacement exploitables. Dans un travail précédent, nous avons choisi de stocker la connaissance dans une base externe. Cette solution offre l'avantage de la généricité et nous permet de bénéficier de techniques avancées d'indexation et d'accès aux données. Cependant, un autre mode de stockage peut faire sens : le stockage de la connaissance sous forme arborescente via un fichier XML qui serait chargé en fonction des besoins. Cette solution entraînerait certes une perte de généricité dans le stockage, *i.e.*, un arbre de connaissance par contexte, mais possède des propriétés intéressantes. D'abord, d'un point de vue algorithmique, la recherche d'information dans un arbre peut être exécutée très efficacement. De plus, cette solution minimiserait grandement le nombre d'accès à la base externe qui est souvent prohibitif dans les applications nécessitant des accès fréquents à une base. Aussi, à court terme, il s'agira de réaliser une étude comparative de ces deux solutions afin d'élire celle offrant les meilleures performances.

5.2 Extension de l'algorithme de généralisation contextuelle

L'algorithme de généralisation présenté précédemment ne peut plus être appliqué car il supposait l'existence d'une unique instance adéquate. Cette contrainte relâchée, il est désormais nécessaire d'introduire un mécanisme pour élire l'instance qui sera utilisée pour généraliser. Nous avons introduit deux méthodes pour relâcher cette contrainte : la prise en compte d'instances généralisées et la possibilité d'utiliser des hiérarchies floues pour les attributs contextualisants. Dès lors, l'ensemble des instances candidates à la généralisation peut contenir des instances plus ou moins générales et/ou plus ou moins adéquates. Une fonction de score agrégeant ces deux mesures de qualité, *i.e.*, généralité et adéquation, doit donc être définie.

Définition 13 *(Score) Soient t un n-uplet de T et c^l une instance du contexte c. Le score quantifiant à quel point c^l représente t s'écrit :*

$$Score(c^l, t) = Agr\big(Prec(c^l),\ \mu(t, c^l)\big)$$

Des investigations supplémentaires sont nécessaires pour définir précisément cette fonction de score. En effet, la généralisation contextuelle choisie sera celle associée à l'instance maximisant ce score. Cette fonction occupe donc une place cruciale dans le processus de généralisation. Nous pensons qu'elle doit être définie en accord avec les utilisateurs du système. Typiquement, certains pourraient vouloir favoriser les instances plus générales plutôt que l'adéquation ou bien des instances faisant intervenir certains attributs, *e.g.*, les instances précisant le traitement médical.

Cette fonction supposée définie, nous nous intéressons maintenant aux modifications à apporter nécessairement pour définir `ROLL_UP_CTX_GEN`, l'opérateur étendu de généralisation contextuelle. La première étape de sélection du contexte est commune aux deux opérateurs. Ensuite, il s'agit de rechercher dans `TC` les instances de contexte pouvant être appliquées au n-uplet considéré. Afin de ne pas considérer des instances redondantes, nous exploitons l'ordre partiel défini dans la section 4.1 pour éliminer les instances dont il existe déjà une spécialisation dans l'ensemble des candidats. Ensuite, la fonction de score est appliquée aux instances restantes. La généralisation retournée sera finalement celle associée à l'instance maximisant le score.

L'algorithme de généralisation introduit deux nouvelles étapes qui peuvent se révéler coûteuses. L'implantation efficace de l'opérateur `ROLL_UP_CTX_GEN` passe alors par une optimisation poussée de chaque étape de l'algorithme.

6 Conclusion

Dans cet article, nous avons montré l'intérêt d'étendre le concept de hiérarchie contextuelle dans les entrepôts de données avec la logique foue. Il s'agit de pouvoir intégrer des connaissances utilisateurs avec davantage d'expressivité et de flexibilité au sein des entrepôts de données, en vue d'améliorer le processus d'analyse.

Cette étude sur l'intégration de la logique floue dans le cadre des hiérarchies contextuelles constitue un début prometteur. Le développement du système correspondant avec une interface permettra de tester l'approche auprès d'utilisateurs décideurs. Pour parvenir à ce résultat, différentes perspectives à moyen terme devront être étudiées, dont un travail poussé sur l'intégration

des connaissances floues et leur exploitation pour la généralisation, avec le souci permanent de la performance compte-tenu du contexte d'analyse en ligne sur des données volumineuses. La richesse d'expressivité a mis en avant dans la partie discussion le rôle que pouvait avoir l'utilisateur décideur dans le choix de la fonction de scoring pour découvrir la meilleure généralisation possible. Ainsi, un travail sur l'exploitation personnalisée de l'entrepôt dans le cadre du recours à des hiérarchies contextuelles floues pourrait être intéressant.

Références

Dubois, D. (2011). The role of fuzzy sets in decision sciences : Old techniques and new directions. *Fuzzy Sets and Systems 184*(1), 3–28.

Fasel, D. et K. Shahzad (2010). A data warehouse model for integrating fuzzy concepts in meta table structures. In *ECBS'10*, pp. 100–109.

Inmon, W. H. (1996). *Building the Data Warehouse, 2nd Edition* (2 ed.). Wiley.

Laurent, A. (2002). *Bases de données multidimensionnelles floues et leur utilisation pour la fouille de données*. Ph. D. thesis, Université Paris 6.

Ligęza, A. (2006). *Logical foundations for rule-based systems*, Volume 11. Springer-Verlag New York Inc.

Malinowski, E. et E. Zimányi (2004). OLAP Hierarchies : A Conceptual Perspective. In *CAiSE'04*, Volume 3084 of *LNCS*, pp. 477–491. Springer.

Perez, D., M. J. Somodevilla, et I. H. Pineda (2007). *Fuzzy Spatial Data Warehouse : A Multidimensional Model*, pp. 3–9. IEEE Computer Society.

Pitarch, Y. (2011). *Résumé de Flots de Données : Motifs, Cubes et Hiérarchies*. Ph. D. thesis, Université Montpellier 2.

Pitarch, Y., C. Favre, A. Laurent, et P. Poncelet (2010a). Analyse flexible dans les entrepôts de données : quand les contextes s'en mêlent. In *EDA'10*, Volume B-6 of *RNTI*, pp. 191–205. Cépaduès.

Pitarch, Y., C. Favre, A. Laurent, et P. Poncelet (2010b). Context-aware generalization for cube measures. In *DOLAP'10*, pp. 99–104.

Zadeh, L. (1965). Fuzzy sets*. *Information and control 8*(3), 338–353.

Summary

Data warehouses are nowadays extensively used to perform analysis on huge volume of heterogeneous data. This success is partly due to the capacity of considering data at several granularity levels thanks to the use of hierarchies. Nevertheless, in previous work, we showed that very few amount of knowledge expert was consider in the generalization process leading to inaccurate analysis. To overcome this drawback, we introduced a new category of hierarchies, namely the contextual hierarchies. Unfortunately, in contrast to the complexity of expert knowledge that should be considered, the knowledge definition process was too rigid. In this paper, we extend these hierarchies and their related techniques to drastically increase their flexibility and expressivity. To this purpose, we present a preliminary work that consists in a fuzzy-based approach.

Weighted Hierarchical Mixed Topological Map: une méthode de classification hiérarchique à deux niveaux basée sur les cartes topologiques mixtes pondérées

Mory Ouattara*,** Ndèye Niang**
Sylvie Thiria *** Corinne Mandin*
Fouad Badran **

*Centre Scientifique et Techniques du Bâtiment 84 Avenue Jean Jaurès-Champs sur Marne
77445 Marne la vallée Cedex 2
http://www.cstb.fr
mory.ouattara@cstb.fr, corinne.mandin@cstb.fr
**CNAM CEDRIC 292 rue St Martin 75141 France-Paris
http://www.cedric.cnam.fr
n-deye.niang_keita@cnam.fr
fouad.badran@cnam.fr
***LOCEAN Université Paris VI 4 place Jussieu, Tour 45-55 75252 PARIS Cedex05, FRANCE
thiria@locean-ipsl.upmc.fr

Résumé. Nous proposons une méthode de classification automatique d'individus décrits par des variables mixtes structurées en blocs. C'est une méthode hiérarchique à deux niveaux, semblable à l'Analyse en Composantes Principales Hiérarchique de Wold, basée sur les cartes topologiques mixtes (MTM). La première étape consiste à établir pour chaque bloc de variables mixtes une classification dite locale représentant une synthèse des individus au niveau du bloc. La deuxième consiste à appliquer MTM sur les résultats du premier niveau pondérés par des indices de qualité des cartes tels que l'indice de Davie-Bouldin. On obtient alors une unique classification globale des individus, consensus entre les partitions issues du niveau 1. La méthode proposée est illustrée sur les données de la campagne nationale «logements» de l'Observatoire de la Qualité de l'Air Intérieur ($OQAI$).

1 Introduction

Dans le domaine environnemental, les phénomènes entraînant la dégradation de l'environnement sont liés à diverses sources de pollution. La qualité de l'air peut par exemple être liée à l'action directe de l'homme sur son environnement ou à des phénomènes naturels. La diversité des sources et causes de pollution de l'air intérieur nécessite des données de très grande dimension qui engendrent en cas de modélisations mathématique et statistique la création de plusieurs blocs de variables. Ces blocs décrivent des phénomènes différents qui peuvent être

des sources de pollutions de l'air intérieur parfois identiques. Ces blocs sont généralement constitués de variables qualitatives et quantitatives, dites mixtes.

L'objectif est alors de quantifier dans un premier temps l'information à l'intérieur de chaque bloc. Dans un deuxième temps, il s'agira d'établir, grâce aux quantifications vectorielles obtenues au niveau de chaque bloc, un résumé final de l'information contenue dans l'ensemble des blocs par consensus entre les blocs.
La problématique de gestion des données mixtes est assez classique et de nombreuses méthodes ont été développées par différents auteurs pour y répondre.

Dans le cadre de l'analyse factorielle, l'étude de variables mixtes repose généralement soit sur un codagel, souvent disjonctif, des variables qualitatives pour ensuite les traiter comme des variables numériques à travers une analyse en composantes principales (ACP), soit sur une transformation adéquate des variables quantitatives en variables qualitatives (Escofier (1979)) en préalable à une analyse des correspondances multiples (ACM). Saporta (1990) propose de réaliser une ACP, avec une métrique judicieusement choisie, sur le tableau résultant de la juxtaposition des variables quantitatives réduites et des variables qualitatives codées sous forme disjonctive complète. Pagès (2004) regroupe les points de vue d'Escofier et de Saporta dans l'analyse factorielle de données mixtes ($AFDM$).

Dans le cadre de la classification automatique d'individus, lorsque ces derniers sont décrits par des données mixtes, une extension naturelle serait d'appliquer la classification sur les coordonnées factorielles aux composantes issues de l'$AFDM$.

Dans le cadre des méthodes neuronales, la classification des données mixtes par la méthode des cartes auto-organisées de Kohonen (1995) repose sur une adaptation de la fonction de coût classique initialement définie pour les variables quantitatives.

Lebbah et al. (2005) proposent la méthode MTM (Mixed Topological Map) pour les données mixtes. C'est une combinaison des algorithmes SOM (Kohonen (1995)) et binbatch (Lebbah et al. (2005)) qui est une adaptation de l'algorithme de Kohonen aux variables qualitatives. On peut aussi noter les travaux de Cottrell et al. (2003) qui proposent la méthode $KACM$ (Kohonen Analyse des Correspondances Multiples) qui consiste à appliquer l'algorithme de Kohonen sur les coordonnées factorielles issues de l'application d'une ACM sur les variables qualitatives.

Pour la classification d'un ensemble d'individus décrits par des variables mixtes structurées en blocs, Niang et al. (2011) proposent la méthode $HMTM$ (Hierarchical Mixed Topological Map) qui consiste à appliquer MTM d'abord sur chaque bloc de variables mixtes et ensuite sur la table composée des résultats des blocs pris séparément. Cette méthode permet de résumer l'information à la fois dans chaque bloc et l'information générale issue de l'ensemble des blocs.

Dans cette communication, nous proposons $Weighted - HMTM$ qui est une extension de $HMTM$ dans laquelle une étape supplémentaire de pondération est ajoutée comme dans l'approche d'ACP hiérarchique de Wold et Kettanah (1996).

Dans la section 2, après avoir rappelé le principe de MTM et la méthode $HMTM$, nous décrivons la méthode $Weighted - HMTM$. En section 3 seront présentés les résultats de l'application à une base de données réelles issue de la campagne nationale « logements » de l'Observatoire de la Qualité de l'Air Intérieur organisée entre 2003 et 2005.

2 Méthodes de classification automatique d'individus décrits par des variables mixtes structurées en blocs

2.1 HMTM (Hierarchical Mixed Topological Map)

$HMTM$ consiste en une double application de l'algorithme MTM que nous décrivons ci-dessous. MTM est une extension de SOM aux données mixtes. L'algorithme MTM est une combinaison de l'algorithme de Kohonen (1995) pour les variables quantitatives et de l'algorithme Binbatch, pour les variables qualitatives codées sous la forme disjonctive (Lebbah et al., 2005). Ainsi, chaque individu z_i de la base de données E=$\{z_i, i = 1, ...N\}$ de dimension n comporte une partie quantitative représentée par z_i^r=$(z_{i1}...z_{ip})(z_i^r \in R^p)$ et une partie qualitative z_i^b=$(z_{i(p+1)}...z_{i(n)})$ $(z_i^b \in \{0,1\}^{n-p})$.
Comme dans SOM, on associe alors à chaque cellule c de la carte C un vecteur référent w_c qui se décompose en $w_c = (w_c^r, w_c^b)$ où $w_c^r \in R^p$ désigne la partie réelle du référent et $w_c^b \in R^{n-p}$ la partie binaire. Ainsi, le vecteur référent a la même structure que les données initiales.

On note par W l'ensemble des vecteurs référents, W^r sa partie réelle et W^b sa partie binaire. Nous décrivons ci-dessous les spécificités de la méthode MTM, en particulier la fonction de coût. Cette dernière repose sur une mesure D de dissimilarité entre un individu z_i et un référent w_c, elle est composée de la distance euclidienne pour la partie quantitative et de la distance de Hamming H pour la partie qualitative :

$$D(z_i, w_c) = ||z_i - w_c||^2 = ||z_i^r - w_c^r||^2 + \beta H(z_i^b, w_c^b) \tag{1}$$

où $\beta = \frac{p}{n-p}$ est le poids relatif des variables qualitatives par rapport aux variables quantitatives. La fonction de coût $J_{MTM}^T(\chi, w)$ à minimiser est alors :

$$J_{MTM}^T(\chi, w) = \sum_{z_i \in E} \sum_{c \in C} k^T(\sigma(\chi(z_i), c)) D(z_i, w_c) \tag{2}$$

où $\chi(z_i) = argmin_c(||z_i - w_c||^2), k^T(k > 0 \; et \; lim_{|x| \to 0}|k(x)| = 0)$, T et σ désignent respectivement la fonction d'affectation d'un individu au référent le plus proche, le système de voisinage associé à chaque référent, le paramètre de voisinage et le paramètre évaluant la distance en deux neurones. L'optimisation de la fonction de coût est réalisée itérativement de manière locale. Elle consiste à choisir le système de référents $w \in W$ qui minimise la fonction $J_{MTM}^T(\chi, w)$. Cette optimisation conduit aux expressions suivantes de calcul des référents :

$$w_c^r = \frac{\sum_{z_i \in E} k(\sigma(\chi(z_i), r)) z_i^r}{\sum_{z_i \in E} k(\sigma(\chi(z_i), r))} \tag{3}$$

pour la partie réelle et à

$$w_c^{bk} = \{_{1 \quad sinon}^{0 \; si \sum_{z_i \in E} k(\sigma(\chi(z_i),r))(1-z_i^{bk}) > \sum_{z_i \in E} k(\sigma(\chi(z_i),r)) z_i^{bk}} \tag{4}$$

pour la partie binaire.

Le choix d'une carte topologique peut être effectué à l'aide de différents critères tels que l'erreur de quantification vectorielle, le taux d'erreur de classification topologique, la mesure de distorsion, les indices de Davies et Bouldin (1987) et la silhouette value de Rousseeuw (1979).
Dans le cas où la taille de la carte ou de manière équivalente le nombre de référents, est trop grand, il est possible d'utiliser une classification ascendante hiérarchique (CAH) pour regrouper les référents en un nombre restreint de classes (Yacoub et al., 2001).

Dans la méthode $HMTM$, Niang et al. (2011) ont appliqué dans une première étape MTM suivi d'une CAH sur les blocs de variables initiaux (Fig. 1) séparément. Cela fournit autant de partitions que de blocs. Ces partitions assimilables à des variables qualitatives sont regroupées dans une table (Fig. 2) sur laquelle on applique à nouveau MTM suivi d'une CAH. C'est donc une méthode à deux niveaux dans laquelle :

– Le niveau inférieur est constitué des blocs de variables initiaux (Fig. 1).

– Le niveau supérieur est constitué d'un bloc de variables catégorielles représentant la classe d'appartenance de chaque individu aux partitions correspondants aux blocs (table en bas à gauche de la Fig. 2).

La méthode $HMTM$ permet d'obtenir à la fois des typologies des individus propres aux variables de chaque bloc et une typologie globale des observations qui tient compte de l'information au niveau des blocs.

	Bloc 1		...	**Bloc P**	
1	$z_{1r_{11}} \cdots z_{1r_{1p1}}$	$z_{1b_{11}} \cdots z_{1b_{1q1}}$		$z_{1r_{p1}} \cdots z_{1r_{pp}}$	$z_{1b_p} \cdots z_{1b_{qp}}$
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.
n	$z_{nr_{11}} \cdots z_{nr_{1p1}}$	$z_{nb_{11}} \cdots z_{nb_{1q1}}$		$z_{nr_{p1}} \cdots z_{nr_{pp}}$	$z_{nb_1} \cdots z_{nb_{qp}}$

FIG. 1 – *Structure des tables au niveau inférieur*

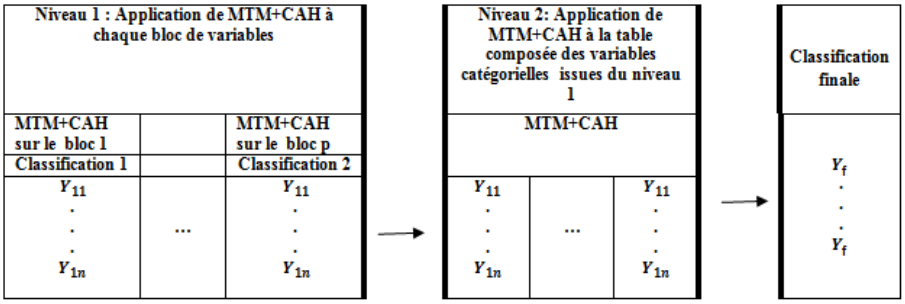


FIG. 2 – *Déroulement de l'algorithme HMTM.* Y_{ji} *est la modalité de la variable qualitative* Y_j *prise par l'individus i dans la classification obtenue au niveau du bloc j et* Y_f *la classification finale obtenue*

2.2 WEIGHTED-HMTM

Les différents indices d'appréciation de la qualité d'une classification montrent que les classifications obtenues au niveau de chaque bloc de variables mixtes sont de qualités différentes. Ils permettent par conséquent d'établir une hiérarchie de qualité entre les classifications. Nous proposons donc de tenir compte de cette hiérarchie en introduisant des coefficients de pondération dans la fonction de coût de $HMTM$. Cela consiste à remplaçer la distance euclidienne classique par une distance euclidienne pondérée sous la forme :

$$D^2(z_i, w_j) = \sum_{k=1}^{n} p_{ik}(z_{ik} - w_{jk})^2 \tag{5}$$

où w_j, z_i, p_{ik} et n correspondent respectivement au centre de classe ou référent, au i ième individu, au poids relatif des individus par rapport au référent w_j et à la dimension de l'espace. De plus, dans $weighted - HMTM$ ($WHMTM$) nous proposons de conserver l'information au niveau des référents dans la première étape.
Donc au niveau supérieur dans $WHMTM$ chaque individu i est représenté par un vecteur $V_i = (V_{i_{K_1}}, ..., V_{i_{K_b}})$ où $V_{i_{K_j}}$ est le vecteur référent ayant capté l'individu i dans le bloc j (cf. Fig 3). Les observations sont donc contenues dans un espace de dimension $(K_1 + ... + K_b)$ où b est le nombre de blocs et K_j la dimension du bloc j.

Ce choix permet de conserver plus d'information dans chaque bloc de variables.

Finalement, dans $WHMTM$, la fonction de coût associée aux données est :

$$J^T_{WHMTM}(\chi, \Omega) = \sum_{V_i \in E} \sum_{c \in C} k^T(\sigma(\chi(V_i), \Omega_c)) D^2(V_i, \Omega_c) \tag{6}$$

Où Ω_c est le référent au niveau supérieur. Formellement, p_{ik} est une fonction de la distance entre l'individu V_i et le référent de la carte C_l du bloc l auquel il appartient au niveau inférieur et de l'indice de Davies-Bouldin associé à cette carte.

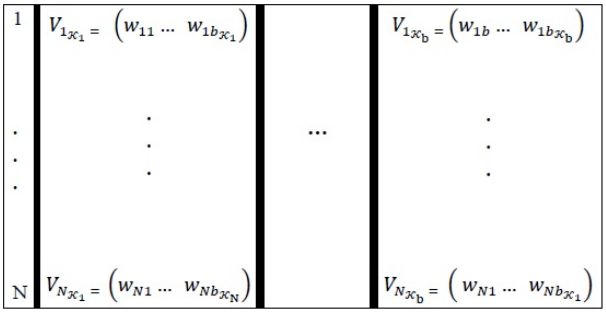


FIG. 3 – *Structure de la table au niveau supérieur dans WHMTM*

Les p_{ik} ne varient pas au cours de l'apprentissage au niveau supérieur et l'ensemble $\{p_{ik}, i \in 1, ..., N \; et \; k = 1, ..., b\}$ vérifie $\sum_i^n p_{ik} = 1 \; et \; 0 \leq p_{ik} \leq 1$. Sous ces hypothèses Huang et al. (2005) montrent que la fonction J_{WHMTM}^T converge. L'optimisation de cette fonction de coût conduit aux centres suivants pour un référent c :

la partie réelle est :

$$W_c^r = \frac{\sum_{V_i \in E} \sum_k p_{ik} k(\sigma(\chi(V_i), c)) V_i^r}{\sum_{V_i \in E} \sum_k p_{ik} k(\sigma(\chi(V_i), c))} \tag{7}$$

La composante k de la partie binaire est :

$$W_c^{bk} = \{^{0 \; si \sum_{V_i \in E} k(\sigma(\chi(V_i),c)) \sum_k p_{ik}(1-V_i^{bk}) > \sum_{V_i \in E} k(\sigma(\chi(V_i),c)) \sum_k p_{ik}(V_i^{bk})}_{1 sinon} \tag{8}$$

L'application de la CAH sur la carte finale fournit la partition d'individus consensus entre les blocs initiaux.

L'interprétation de cette partition consensus fait apparaître un ensemble de variables caractéristiques des classes. On aurait aussi pu considérer comme partition finale celle ayant le meilleur indice de Davies-Bouldin au niveau des blocs. Mais l'interprétation de cette partition en utilisant les variables du bloc associé et celles des autres blocs considérées comme variables illustratives engendrent une perte d'information quant aux variables effectives (variables de chaque bloc) qui caractérisent réellement les classifications dans ces blocs. Cela sera plus clairement illustrée dans l'application.

3 Application sur les données de la Campagne Nationale « Logements»

L'Observatoire de la Qualité de l'Air Intérieur (OQAI) a réalisé entre 2003 et 2005 une campagne nationale dans les logements sur un échantillon de 567 logements représentatif du parc des 24 millions de résidences principales de la France continentale métropolitaine. Cette

campagne vise à dresser un état de la pollution de l'air dans l'habitat afin de donner les éléments utiles pour l'estimation de l'exposition des populations, la quantification et la hiérarchisation des risques sanitaires associés, ainsi que l'identification des facteurs prédictifs de la qualité de l'air intérieur.
Des questions ont été posées sur : la structure du bâtiment (les matériaux utilisés pour la construction, les produits de décoration, le mobilier, ...), la structure des ménages et les habitudes des occupants.
Finalement une base de données composée de plus de 628 variables a été constituée. Cette base a été ensuite divisée en trois bases par regroupement et combinaison de variables suivant trois critères. Le premier critère décrit les logements, le deuxième critère décrit la structure des ménages et le troisième critère décrit les habitudes des occupants. Ces trois bases sont dans la suite appelées blocs de variables.

- Le bloc logements est composé de 71 variables dont 32 quantitatives.
- Le bloc ménages est composé de 12 variables dont 5 quantitatives.
- Le bloc habitudes est composé de 45 variables dont 21 quantitatives.

3.1 Apprentissage des cartes

Plusieurs apprentissages sont effectués au niveau des blocs de variables en faisant varier les paramètres de la fonction de coût, le nombre d'itérations n_i, les dimensions de la carte. La meilleure carte est alors choisie parmi l'ensemble des apprentissages, elle équivaut à la carte qui donne le meilleur indice de Davie Bouldin (Fig.4). Rappelons qu'il est possible d'utiliser d'autres indices.

3.2 Application de l'algorithme WHMTM

L'application de la méthode proposée sur chacun des blocs de variables de la campagne nationale « logements » montre à travers l'indice de Davie-Bouldin que la carte obtenue sur le bloc ménages est la meilleure au niveau inférieur. Dans la suite de cette section nous donnons d'abord une description des classes basée sur les variables les plus caratéristiques du bloc mais aussi à travers les variables des autres blocs considérées comme illustratives. Ensuite nous décrivons les classes obtenues après application de $WHMTM$ par rapport à l'ensemble des variables.

Au niveau inférieur :

Dans le bloc logement, on obtient une classification en 5 classes. Les variables les plus caractéristiques de ces classes sont : la surface du logement, l'année de construction du bâtiment, le type de logement (individuel ou collectif), l'ameublement intérieur des logements. Ainsi, les classes 1 et 2 contiennent les petits logements collectifs récents. La classe 3 regroupe les logements ou maisons récents de taille moyenne possédant beaucoup d'appareils à combustion raccordés à un conduit de fumée.
Les classes 4 et 5 regroupent les grandes maisons individuelles généralement tout en un raccordées à des conduits de fumée et possédant un fort taux de meubles en bois massifs (classe

4) et les petite maisons individuelles (classe 5).

En utilisant les variables des autres blocs pour enrichir l'interprétation on constate que seule la variable «revenu» du bloc ménage et certaines variables d'entretient du logement et de soins corporels du bloc habitudes permettent de caractériser les classes de logements. Par exemple les classes de logements 1 et 2 sont associées à des familles ayant de faibles revenus alors que les logements 3 et 4 sont associés à des familles ayant des revenus élevés. Notons que n'interviennent pas dans la description les variables décrivant le nombre d'enfants de moins ou de plus de 10 ans par foyer.

Dans le bloc ménage, on obtient une classification en 6 classes. Les variables les plus caractéristiques des classes sont : les revenus du ménage, le nombre de personnes, l'âge des enfants et le statut des ménages (personnes seules ou en couples).

En utilisant les variables des autres blocs on retrouve certaines variables significatives dans l'interprétation précédente comme la variable taux de meubles en bois du bloc logement et les variables d'entretient et de produits de soins corporels du bloc habitude qui permettent de caractériser les classes du bloc ménage. Par contre la variable âge du logement ne caractérise plus de manière significative les classes. De même l'âge des enfants significative ici ne l'était pas dans la partition du bloc logement. On constate alors une perte d'information dans les interprétations. Il en va de même pour l'interprétation de la partition associée au bloc habitude.

Au niveau supérieur, on obtient une classification en 8 classes après application d'une classification ascendante hiérarchique sur les neurones de ce niveau.

Les familles aisées, âgées, souvent nombreuses avec beaucoup d'enfants de plus ou de moins de 10 ans habitent généralement dans des maisons individuelles tout en un avec garage et cave attenants, ils entretiennent beaucoup leur maison. A contrario, dans les logements collectifs on retrouve les jeunes ou les retraités vivant seuls et ayant des revenus moyens, ils entretiennent peu leur logement. Le nombre d'enfants de plus ou de moins de 10 apparait important pour les classes de cette partition

En plus de cette confrontation au niveau de l'interprétation des classes, nous avons comparé les partitions en utilisant l'indice Rand qui met en évidence la similarité entre les partitions (TAB 1.), plus la valeur de cet indice est proche de 1 mieux est la similarité entre les deux classifications.

Les valeurs relativement élevées des indices de rand entre partitions du niveau inférieur montrent certes une similitude mais elle montre aussi le fait que chaque bloc de variables porte une information qui lui est spécifique. La méthode directe qui consiste à appliquer MTM sur la table juxtaposant les blocs initiaux (MTM_{gle}) fournit une partition globale moins semblable aux partitions locales à la différence des partitions obtenues avec $HMTM$ et $WHMTM$ plus consensuelles.

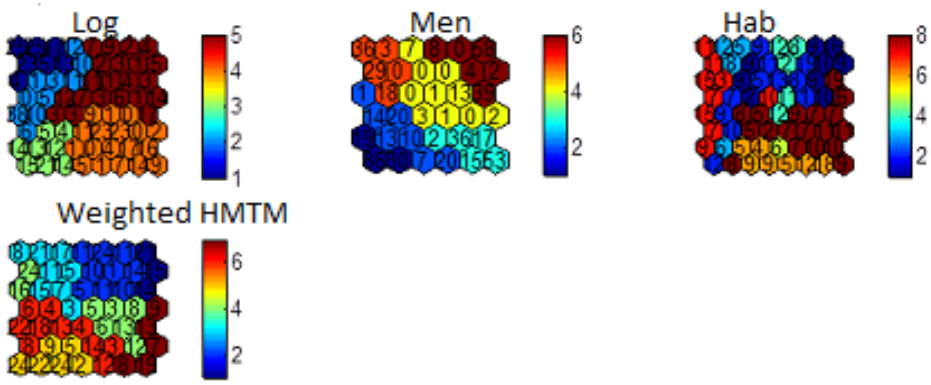


FIG. 4 – *Structure des cartes aux niveaux inférieur et supérieur : en haut les cartes du niveau inférieur et en bas la carte du niveau supérieur*

	Log	Men	Hab	weighted-HMTM$_{\Pi_k}$	MTM$_{gle}$
Log	1	0.69	0.70	0.82	0.73
Men		1	0.68	0.68	0.67
Hab			1	0.72	0.66
Weighted-HMTM$_{\Pi_k}$				1	70
MTM$_{gle}$					1

TAB. 1 – *Valeurs de l'indice de Rand de comparaison des classifications : Log, Men, Hab, weighted-HMTM$_{\Pi_k}$, MTM$_{gle}$ correspondent respectivement aux classifications obtenues sur les blocs logement, ménage, habitude, les individus sont affectés des poids P_{ik}, enfin la classification obtenue en concaténant les tables composées des données initiales dans chaque bloc*

4 Conclusion

Face à la problématique de classification d'individus décrits par des variables mixtes structurées en blocs, la méthode WHMTM crée un consensus entre l'information dans chaque bloc de variable mixtes et l'information globale issue de l'ensemble des variables. Les pondérations introduites au niveau supérieur permettent d'améliorer la structure de la carte finale. Cependant, le consensus peut être amélioré entre les blocs et nécessite des recherches supplémentaires. Notamment, en procédant à la sélection des variables les plus informatives dans le passage du niveau 1 au niveau 2, cela permettrait une réduction de la dimension des individus au niveau 2 et en rendant les poids adaptives.

Références

Cottrell, M., I. Smail, et P. Letrémy (2003). Cartes auto-organisés pour l'analyse exploratoire de données et la visualisation. *Journal de la Société Française de Statistique 144*, 67–106.

Davies, D. L. et D. W. Bouldin (1987). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence (2) 4*, 224–227.

Escofier, B. (1979). Traitement simultané de variables quantitatives et qualitatives en analyse factorielle. *RSA 4*, 137–146.

Huang, J. Z., M. K. N. H. Rong, et Z. LI (2005). Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 657–668.

Kohonen, T. (1995). the self organizing map.

Lebbah, M., A. Chazottes, F. Badran, et S. Thiria (2005). Cartes topologiques mixtes. *EGC 5*, 43–48.

Niang, N., M. Ouattara, J. Brajard, et C. Mandin (2011). classification d'individus decrits par des variables mixtes structurées en blocs. *SFDS 43*, 146–152.

Pagès (2004). *Analyse Factorielle de Données Mixtes*. ITALIA : RSA.

Rousseeuw, P. J. (1979). Silhouettes : a graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics, Volume 20, Page 65 20*, 53–65.

Saporta, G. (1990). Simultaneous analysis of qualitative and quantitative data atti della xxxv riunione scientifica. *Società itialana di statistica 35*, 57–64.

Wold, S. et N. Kettanah (1996). Hierarchical multiblock pls and pc models interpretation and as an aternative to variable selection. *J. Chemon 10*, 463–482.

Yacoub, M., N. Niang, F. Badran, et S. Thiria (2001). A new hierarchical clustering method using topological map. *ASMDA2001*.

Summary

We propose a method to solve the problem of clustering individuals described by mixed variables structured in blocks. It is a hierarchical method with two levels similar to Hierarchical Principal Component Analysis presented by Wold, based on the mixed topology maps (MTM). The first step consists in establishing, for each block of mixed variables, a clustering representing a synthesis of local individuals at the block level. The second step is to re-apply MTM on the weighted results of the first level. We obtain a global clustering of individuals, consensus among the partitions from level 1. The proposed method is illustrated on data from the national survey in the French housing stock carried out in 2003-2005 by the Observatory for Indoor Air Quality (OQAI).

Classification croisée par approche évolutionnaire itérative

Alexandre Blansché et Lydia Boudjeloud-Assala

Laboratoire d'Informatique Théorique et Appliquée (EA 3097)
Île du Saulcy, 57045 Metz Cedex 1
lydia.boudjeloud|alexandre.blansche@univ-metz.fr

Résumé. Nous proposons dans cet article une classification croisée par approche évolutionnaire itérative. L'approche évolutionnaire basée sur un algorithme génétique propose une alternative originale pour résoudre le problème de sélection de plusieurs sous-espaces décrivant le mieux les données. Ces sous-espaces mettent en évidence des structures particulières des données et sont sélectionnés itérativement. Cette étape itérative, nous permet donc de faire une classification croisée sur les données et sur les dimensions composant le sous-espace de façon simultanée. L'approche génétique évalue un sous-ensemble de dimensions avec des mesures permettant de trouver la meilleure topologie des données pour ce sous-ensemble de dimensions puis on réitère le processus pour plusieurs sous-ensembles de dimensions sélectionnées par l'algorithme génétique. Des expérimentations sur des ensembles de données ayant un très grand nombre de dimensions montrent l'efficacité de l'approche proposée.

1 Introduction

La plupart des méthodes classiques de classification non supervisée se basent sur des approches divisant un ensemble de données de N objets représentés par D dimensions formant une matrice de taille $N \times D$, en sous-ensembles tels que les données d'un même sous-ensemble soient similaires pour un certain critère fixé. Dans les ensembles de données à très grandes dimensions tels que ceux issus du domaine biomédical ou de la génétique, les techniques classiques de classification ne sont plus efficaces pour traiter ce type de données. Une des solutions proposée est d'utiliser des méthodes de classification croisée afin d'obtenir une structure des données en blocs homogènes. Les approches classiques utilisées (Hartigan, 1972) consistent à appliquer des algorithmes de classification simple sur l'ensemble des objets et sur l'ensemble des dimensions séparément, les blocs résultent du croisement des deux partitions obtenues. L'algorithme CLIQUE (**?**) se base sur une discrétisation des attributs numérique et construit des classes à la manière de l'algorithme Apriori. D'autres approches consistent à chercher simultanément une partition en lignes et une partition en colonnes, tel que l'algorithme Croeuc (Govaert et Nadif, 2003). Les centres des blocs, ainsi obtenus, constituent une matrice de taille réduite offrant un résumé des données. Les algorithmes existant génèrent ainsi des sous ensembles d'objets présentant un comportement similaire dans un sous-ensemble de dimensions et vise versa.

L'idée traitée dans cet article est de proposer une approche de classification croisée en sélectionnant plusieurs sous-espaces qui permettent de mettre en évidence des structures complexes et particulières des données ou de trouver des sous-espaces de données homogènes. Les données peuvent se chevaucher contrairement aux dimensions. L'idée est de voir comment les données peuvent se comporter sur les différents sous-espaces sélectionnés, parfois ils représentent des objets atypiques, parfois des classes homogènes ou présentent des relations d'homogénéité avec d'autres données. Nous vérifions ensuite, si il y a un recouvrement total de l'ensemble de données ou pas, ainsi qu'un recouvrement entre les classes.

Cette sélection se fait de façon itérative à partir d'un algorithme génétique et d'un critère d'évaluation d'une classe homogène par rapport à l'ensemble total des données. L'originalité de notre approche est qu'un objet (ou un groupe d'objets) peut être décrit par plusieurs sous-ensembles de dimensions. Cet objet peut se comporter comme un objet atypique (outlier) dans un sous-espace, et dans un autre sous-espace il se mêle complètement dans la masse ou pouvant faire partie d'un cluster homogène.

2 Approche génétique itérative

La méthode proposée consiste à découvrir, une à une, des groupes d'objets homogènes (classes) dans un sous-ensemble de dimensions. Pour cela, à chaque itération de la méthode, un algorithme évolutionnaire est utilisé pour découvrir une telle classe. L'évaluation des dimensions se base sur une mesure d'homogénéité d'une classe de données. Deux critères sont utilisés dans cet article : le critère de compacité de Wemmert et Gançarski (Wemmert et al., 2000) ainsi qu'un critère R_I fondé sur l'inertie.

L'algorithme génétique explore les sous-ensembles de dimensions. Pour chaque sous-ensemble étudié, nous recherchons un ensemble de classes (avec un algorithme simple tel que K-means. Puis, pour chaque classe trouvée nous calculons son indice de compacité associé. La meilleure classe selon l'indice choisi sera associée au sous-ensemble de dimensions ainsi que la valeur de l'indice. Le sous-ensemble de dimensions sera alors évalué selon l'évaluation de la classe associée. L'algorithme génétique va ainsi chercher le sous-ensemble de dimensions qui contient la classe la plus compacte possible.

2.1 Evaluation d'un sous-ensemble de dimensions

Le critère de compacité de Wemmert et Gançarski (Wemmert et al., 2000) s'appuie sur le rapport entre deux distances pour chaque objet o : la distance entre o et le centre C_k de sa classe d'appartenance et la distance entre o et le centre différent de C'_k le plus proche de o.

Ce critère tient compte à la fois de la compacité et de la séparabilité des classes, l'indice de compacité de Wemmert et Gançarski pour une classe C_k est défini par :

$$WG\left(C_k\right) = \begin{cases} 0 & \text{si } \frac{1}{Card(C_k)} \sum\limits_{o \in C_k} \frac{d\left(o, c_k\right)}{d\left(o, c_{k'}\right)} > 1 \\ 1 - \frac{1}{Card(C_k)} \sum\limits_{o \in C_k} \frac{d\left(o, c_k\right)}{d\left(o, c_{k'}\right)} & \text{sinon} \end{cases} \quad (1)$$

où $k' = Argmin_{h \neq k} \left(d\left(o, c_h\right) \right)$ et $d\left(o, o'\right)$ représente la dissimilarité entre deux objets o et o' dans le sous-ensemble d'attributs utilisé pour construire la classe.

Cet indice prends ses valeurs entre 0 et 1. Plus la classe est compacte plus l'indice est élevé.

Nous proposons d'utiliser également un critère fondé sur l'inertie intraclasse. Afin de s'adapter aux divers ensembles de dimensions, nous utiliserons le rapport entre l'inertie intraclasse et l'inertie totale des données. Ce critère est ainsi défini par :

$$R_I\left(C_k\right) = \frac{\sum\limits_{o \in C_k} d\left(o, c_k\right)^2}{\sum\limits_{o \in D} d\left(o, g\right)^2} \tag{2}$$

où D est l'ensemble de données complet et g l'isobarycentre de D et $d\left(o, o'\right)$ représente toujours la dissimilarité entre deux objets o et o' dans le sous-ensemble d'attributs utilisé pour construire la classe.

Cet indice prends ses valeurs entre 0 et $+\infty$. Plus la classe est compacte plus l'indice est faible.

2.2 Algorithme génétique

L'algorithme génétique utilisé a été proposé par Boudjeloud-Assala (2005). Le codage utilisé consiste à se fixer un nombre de dimensions à sélectionner ss. Un individu génétique représente donc une combinaison possible de ss dimensions. Ce type de codage, permet une meilleure interactivité avec les dimensions. La taille ss est un paramètre d'entrée de l'algorithme génétique. Les individus de la population de départ sont constitués à partir d'un tableau contenant toutes les dimensions disponibles qui décrivent l'ensemble des données. On associe un taux de fréquence à chaque dimension, un nouvel individu est constitué à partir des dimensions non utilisées (qui ont une fréquence minimale). Vu le nombre important de dimensions disponibles il est nécessaire, à notre avis, de balayer toutes les dimensions pour ne garder que les plus pertinentes. A ce niveau là, la population de départ est prête, elle ne contient pas deux individus identiques ni deux même dimensions dans le même individu génétique. C'est le rôle des deux fonctions $Noclone$ (pour éliminer deux individus identiques) et $EtudInd$ (pour modifier un gène s'il est identique à un autre gène du même individu) qui sont appelées à chaque création d'un nouvel individu génétique (Boudjeloud-Assala, 2005). Une fois la population évaluée par la mesure de compacité, elle est triée en ordre décroissant, nous opérons un croisement sur deux parents choisis aléatoirement. L'un des deux enfants est ensuite muté avec une probabilité de p_{mut} et remis dans la population en remplaçant un individu dans la deuxième partie de la population, sous la médiane. Le choix de l'individu remplacé se fait aléatoirement (Boudjeloud-Assala, 2005). L'opérateur de croisement favorise l'exploration de l'espace de recherche. Il permet de créer de nouvelles combinaisons de dimensions. Il existe des méthodes où il faut déterminer deux points de coupure ou bien un seul, appelées respectivement, "croisement à deux points" et "croisement à un point". Nous croisons deux parents avec un "croisement à un point", cette opération est appelé croisement simple (Boudjeloud-Assala, 2005), nous obtenons deux enfants. Pour s'assurer qu'il n'existe pas deux dimensions identique dans le même enfant (présentées dans un ordre différent ou pas) et que les mêmes

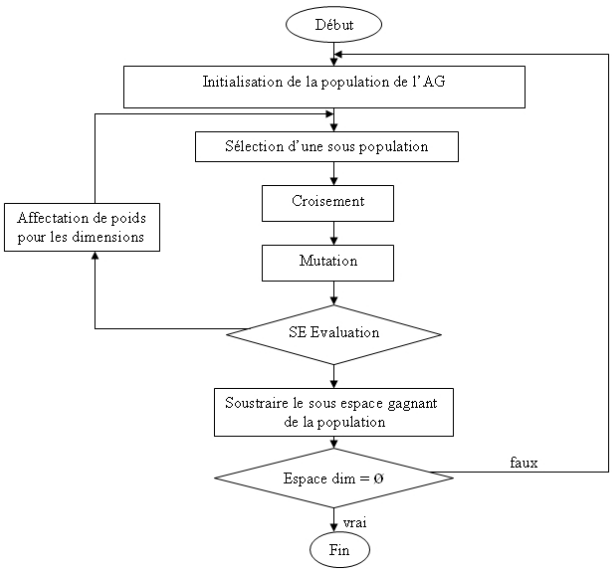


FIG. 1 – *Algorithme génétique itératif pour le biclustering.*

dimensions ne réapparaissent pas chez un même enfant nous appelons les fonctions $Noclone$ et $EtudInd$.

Nous avons, dans un premier temps, opté pour un point de coupe déterminé aléatoirement que nous avons modifié de façon à obtenir un point de coupe "optimisé" ce qui implique une évaluation de l'individu issu de chaque coupe possible. Nous opérons dans un premier temps un croisement créant deux enfants initiaux. Le premier enfant issu d'un processus de croisement répété meilleur que l'un des deux enfants initiaux détermine le meilleur point de coupe optimisé. Durant la phase de mutation, nous modifions un seul gène (représentant une dimension) en le remplaçant par une dimension différente des autres gènes composant l'individu en suivant la procédure $EtudInd$ car nous pourrions avoir un individu ayant deux fois la même dimension dans le sous-espace de dimensions traité, cas à éviter.

2.3 Approche itérative

Le processus itératif permet de faire une sélection itérative de sous-espaces de dimensions, l'algorithme génétique réduit l'espace de données en déterminant uniquement le sous-espace pertinent. Une fois le sous-ensemble de dimensions sélectionné, on élimine les dimensions qui le composent de l'ensemble de recherche, et on réitère le processus de l'algorithme génétique (Boudjeloud-Assala, 2005) jusqu'à ce que l'on élimine toutes les dimensions. Comme on peut le voir sur la figure 1. Pour l'exécution de l'AG, nous devons fixer au préalable plusieurs paramètres, tel que ss, la taille du sous-ensemble de dimensions, nb_{iter}, le nombre d'itérations de l'algorithme génétique, dont dépendent n_c, le nombre de croisements, et n_m, le nombre de mutations, enfin $npop$, la taille de population.

3 Expérimentation

Afin de tester la faisabilité de notre approche nous avons procédé à deux séries d'expérimentations. S'agissant d'un travail préliminaire, nous avons choisi des ensembles de données de taille modérée.

La première série d'expérimentations a pour but de vérifier que l'algorithme est capable d'extraire des classes variées et de bonne qualité selon l'indice utilisé. Pour cela, nous étudierons le recouvrement des données ainsi que les intersections des classes. La seconde série d'expérimentations a pour objectif de montrer, sur un exemple simple, quelles connaissances peuvent être extraites à partir des données.

3.1 Recouvrement des classes

Dans la première série d'expérimentations, nous avons appliqué notre méthode sur l'ensemble *Libras Movement*, provenant du dépôt de l'UCI (Blake et Merz, 1998), qui contient 360 objets décrits dans un espace à 90 dimensions numériques. Nous avons arbitrairement fixé la taille des sous-ensembles de dimensions à 10, le nombre de générations à 1000, la taille de la population à 50. Les classes ont été extraite par l'algorithme K-means (McQueen, 1967) configuré pour découvrir trois classes. La meilleure des trois classes selon l'indice WG est considérée comme étant la classe extraite par un individu de l'AG. Afin de s'assurer d'avoir des classes significatives plutôt que des petits ensembles d'objets isolés du reste des données, nous avons éliminé toutes les classes de taille inférieure à 10 % de la taille de l'ensemble des données.

Sur l'ensemble de données *Libras Movement*, il y aura donc neuf classes extraites (chaque classe étant représenté dans un sous-espace de 10 dimensions parmi les 90 d'origine). L'algorithme évolutionnaire n'est bien entendu pas utilisé pour la dernière classe extraite (il ne reste alors plus que les 10 dernières dimensions à choisir). Les résultats présentés sont des moyennes sur 10 exécutions de l'algorithme. En terme de temps de calcul, en programmation Java sur un processeur Intel à 1,3GHz, l'extraction d'une classe a pris entre huit et neuf minutes.

Nous avons tout d'abord étudié la qualité des classes extraites selon l'indice WG, selon l'ordre dans lequel elles ont été extraites (cf. fig. 2). Nous remarquons que les classes les plus compactes ont été extraites lors des premières itérations de l'algorithme proposé. La qualité a ensuite tendance à baisser dans la suite des itérations. En particulier, la dernière classe extraite est de très mauvaise qualité. Dans la suite, nous ne tiendrons plus compte de la dernière classe extraire, mais uniquement des 8 premières. Ce comportement était prévisible. En effet, les premières classes extraites (lors des premières itérations) sont censées être les plus compactes, donc avec un indice WG élevé. À mesure que nous retirons des dimensions, il est donc normale de trouver des classes de moindre qualité. Afin d'évaluer le recouvrement des classes et de s'assurer qu'un groupement d'objets identiques n'a pas été extrait à chaque itération de l'algorithme, nous avons évalué à combien de classe chaque objet appartient. Nous pouvons alors représenter graphiquement sous la forme d'un histogramme, le pourcentage d'objets appartenant à un nombre donné de classes (cf. fig. 3). On notera en particulier que plus de 95 % de l'ensemble des données appartient à au moins une classe, ce qui montre que l'ensemble des classes extraites forme un bon recouvrement de l'ensemble de données. On note également que le plupart des objets appartiennent à une, deux ou trois classes. Très peu d'objets appartiennent en même temps à cinq classes ou plus.

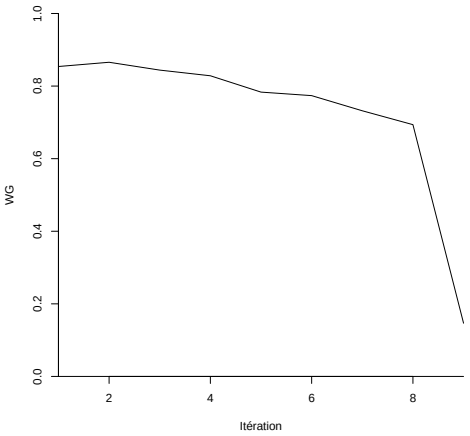


FIG. 2 – *Évaluation des classes extraites selon WG*

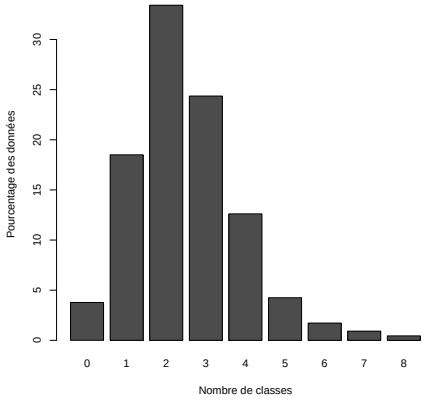


FIG. 3 – *Pourcentage d'objets appartenant à un nombre donné de classes*

3.2 Extraction de connaissances

Dans la seconde série d'expérimentations, nous avons appliqué l'algorithme proposé sur un ensemble de données artificiel très simple, généré de la façon suivante :

- il y a 400 objets et neuf dimensions ;
- objets 1 à 100 ont une distribution gaussienne centrée en 0 avec un écart-type de 1 sur les trois premières dimensions, et une distribution gaussienne centrée en 50 ou en -50 (aléatoirement) et un écart-type de 5 sur les dimensions 4 à 9 ;
- objets 101 à 200 ont une distribution gaussienne centrée en 0 avec un écart-type de 1 sur les dimensions 4, 5 et 6, et une distribution gaussienne centrée en 50 ou en -50 (aléatoirement) et un écart-type de 5 sur les autres dimensions ;
- objets 201 à 300 ont une distribution gaussienne centrée en 0 avec un écart-type de 1 sur les dimensions 7, 8 et 9, et une distribution gaussienne centrée en 50 ou en -50 (aléatoirement) et un écart-type de 5 sur les autres dimensions ;
- objets 301 à 400 ont une distribution gaussienne centrée en 0 avec un écart-type de 1 sur les dimensions 1 à 6, et une distribution gaussienne centrée en 50 ou en -50 (aléatoirement) et un écart-type de 5 sur les dimensions 7 à 9.

Nous avons fixé la taille des sous-ensembles de dimensions à 3, le nombre de générations à 1000, la taille de la population à 20. Les classes ont été extraite par l'algorithme K-means configuré pour découvrir trois classes. La meilleure des trois classes selon le critère R_I est considérée comme étant la classe extraite par un individu de l'AG. Afin de s'assurer d'avoir des classes significatives plutôt que des petits ensembles d'objets isolés du reste des données, nous avons éliminé toutes les classes de taille inférieure à 10 % de la taille de l'ensemble des données.

Trois classes homogènes ont été extraites par l'algorithme :

- C_1 contient les objets 201 à 300 sur les dimensions 7, 8 et 9 ;
- C_2 contient les objets 101 à 200 et 301 à 400 sur les dimensions 4, 5 et 6 ;

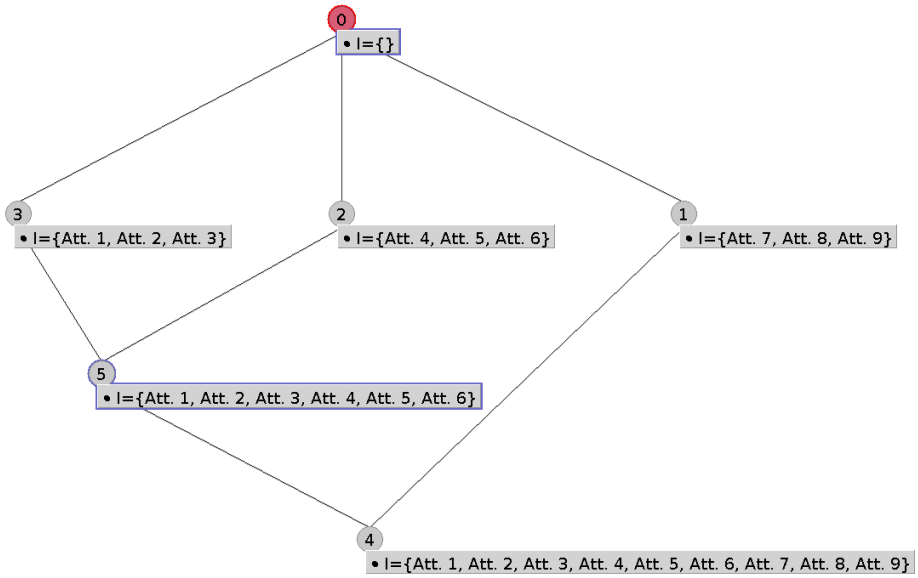


FIG. 4 – *Treillis de concept issu des classes extraites*

– C_3 contient 1 à 100 et 301 à 400 sur les dimensions 1, 2 et 3.

On remarque ainsi que tous les objets ont été affectés à au moins une classe et que les classes C_2 et C_3 se chevauchent (les objets 301 à 400 appartiennent aux deux classes).

Afin d'approfondir l'étude des classes extraite, nous avons décidé d'utiliser l'analyse de concepts formels (Ganter et Wille, 1999). Nous construisons un contexte à partir des classes extraites par notre algorithme. Les objets du contexte sont les objets de l'ensemble de données. L'attribut A_i correspond à la dimension i. Un objet o possède l'attribut A_i si $\exists C_k | o \in obj(C_k) et A_i \in dim(C_k)$ où $obj(C_k)$ représente l'ensemble des objets de C_k et $dim(C_k)$ l'ensemble des dimensions définissant C_k. Si un objet possède l'attribut A_i, cela signifie qu'il appartient à une classe qui est compacte sur la dimension i

À partir de ce contexte, un treillis a été construit à l'aide du logiciel Galicia [1] (cf. fig. 4).

On constate que quatre concepts ont été créés (en plus de $\top$ et $\bot$). Le concept 1 correspond à la classe 3, le concept 2 à la classe 2, le concept 3 à la classe 1 et le concept 4 à l'intersection des classes 2 et 3 (en terme d'objet, à leur union en terme d'attributs). On peut donc en déduire qu'il y a quatre classes dans les données compactes dans ensembles de dimensions de taille 3 ou 6.

4 Conclusion et travaux futurs

La classification croisée permet de mettre en évidence des structures complexes et particulières de données. Dans l'approche proposée, l'extraction de classes homogènes se fait

1. `http://www.iro.umontreal.ca/~galicia/`

de façon itérative à partir d'un algorithme évolutionnaire et d'un critère d'évaluation d'un sous-espace (un sous-ensemble de dimensions). Les travaux préliminaires sur notre approche itérative de classification croisée par algorithme évolutionnaire sont engageants. L'algorithme s'est montré capable d'extraire un ensemble de classe, chacune définie dans un sous-espace de l'espace de représentation des données. Les premières expériences ont permis de montrer que l'algorithme était capable de découvrir un ensemble de classes variées à partir de sous-ensembles de dimensions disjoints. Ces résultats nous encouragent à poursuive nos travaux, ouvrant la voie à de nombreuses perspectives.

Il sera, tout d'abord intéressant d'étudier le comportement de la méthode avec d'autres critères d'évaluation. L'algorithme d'extraction utilisé est l'algorithme K-means, cette approche présentant certains désavantages. En particulier, il est nécessaire de choisir le nombre de classes que découvrira K-means pour ensuite ne conserver qu'une seule de ces classes. De plus, l'utilisation de K-means est coûteuse pour l'extraction d'une unique classe. Nous comptons donc étudier d'autres méthodes pour extraire une classe homogène d'un ensemble de données plus rapidement. De plus, dans un souci de simplification, le nombre de dimensions dans les sous-ensembles est actuellement fixe. Il sera intéressant de faire varier le nombre de dimensions qui décrivent les classes extraites, en utilisant des mécanismes évolutionnaires permettant de traiter des gènes de taille variable. Enfin, nous approfondirons la possibilité de représenter l'ensemble des classes extraites par un treillis.

Les premières expérimentations que nous avons menées sont encourageantes et ont montrées la faisabilité de notre approche. Il reste cependant à mener des études comparatives afin de situer l'efficacité de notre proposition par rapport aux autres algorithmes de biclustering.

Références

Agrawal, R., J. Gehrke, D. Gunopulos, et P. Raghavan (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pp. 94–105.

Blake, C. et C. Merz (1998). Uci repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences. http ://www.ics.uci.edu/∼mlearn/MLRepository.html accédé en septembre 2005.

Boudjeloud-Assala, L. (2005). *Visualisation et Algorithmes Génétiques pour la Fouille de Grands Ensembles de Données*. Thèse de Doctorat de l'Ecole Polytechnique de l'Université de Nantes.

Ganter, B. et R. Wille (1999). *Formal Concept Analysis, Mathematical Foundation*. Springer.

Govaert, G. et M. Nadif (2003). Clustering with block mixture models. *Pattern Recognition 36*, 463–473.

Hartigan, J. (1972). Direct clustering of data matrix. In *journal of the American statistical association*, Volume 67, pp. 123–129.

McQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297.

Wemmert, C., P. Gançarski, et J. Korczak (2000). A collaborative approach to combine multiple learning methods. *International Journal on Artificial Intelligence Tools 9*(1), 59–78.

Summary

We propose in this paper an iterative genetic algorithm for biclustering. The genetic algorithm offers an original alternative to solve the problem of selecting subspace to deal with complex data structures in different subspaces. These subspaces that shows specific data structures are chosen iteratively. This step allows us to cluster data and subspace simultaneously. The genetic approach evaluates a subset of dimensions with a measure to find the best data cluster and then repeat the process for several subsets selected iteratively by the genetic algorithm. Experiments on data sets with a large number of measurements show the effectiveness of the proposed approach.