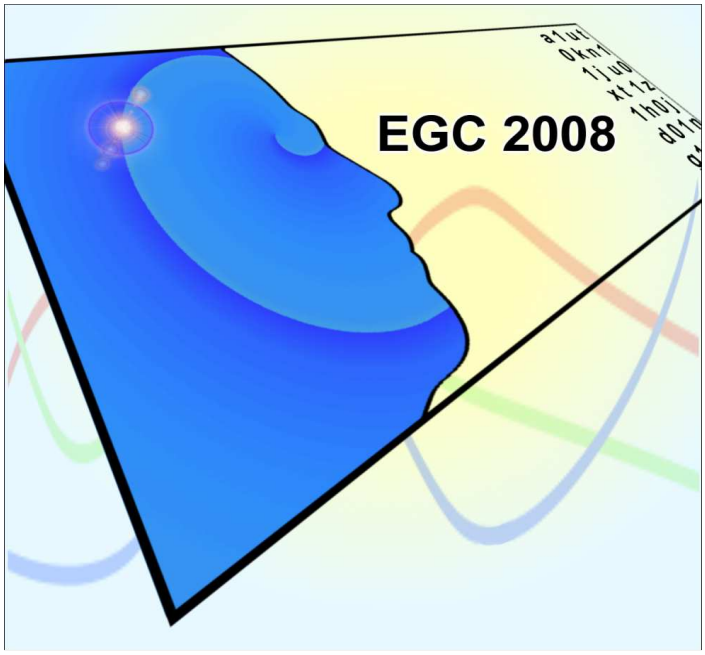
8èmes Journées Francophones
Extraction et Gestion des Connaissances
Sophia Antipolis 29 janvier 2008

Atelier



Fouille de données complexes dans un processus d'extraction des connaissances

Organisateurs :

- Pierre Gançarski (LSIIT, Strasbourg)
- Arnaud Martin (ENSIETA, Brest)

Responsables des Ateliers EGC :

Alzennyr Da Silva (INRIA, Rocquencourt)
Alice Marascu (INRIA, Sophia Antipolis)
Florent Masseglia (INRIA, Sophia Antipolis)

http://www-sop.inria.fr/axis/egc08

EGC

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche SOPHIA ANTIPOLIS - MÉDITERRANÉE

TABLE DES MATIÈRES

5ème atelier sur la “Fouille de données complexes dans un processus d’extraction des connaissances”

1 Présentation

L’atelier sur la fouille de données complexes dans un processus d’extraction de connaissances est organisé à l’instigation du groupe de travail “Fouilles de Données Complexes” GT FDC et s’inscrit dans le cadre de la conférence EGC. Cet atelier se veut être un lieu de rencontre annuel où chercheurs/industriels peuvent partager leurs expériences et expertises dans le domaine de la fouille de données. L’atelier se veut ouvert en terme de propositions. On pourra y présenter aussi bien un travail abouti, des réflexions sur la fouille de données complexes ou un travail préliminaire (qui présentera davantage un problème qu’une solution). Enfin, les discussions sur les liens entre différentes disciplines sont également bienvenues.

Les quatre premières éditions de cet atelier au sein d’EGC (2004 à Clermont-Ferrand, 2005 à Paris, 2006 à Lille et 2007 à Namur) furent une réelle réussite, accueillant des chercheurs/doctorants représentant plus de 30 laboratoires francophones différents. Ils auront permis d’avancer sur la compréhension de la complexité d’un processus d’extraction de connaissances à partir de bases de données et d’initier de nouveaux échanges scientifiques entre chercheurs. La cinquième édition de cet atelier a eu lieu dans le cadre de EGC 08 (Sophia Antipolis).

Thèmes

Dans tous les domaines tels que le multi-média, la télédétection, l’imagerie médicale, les bases de données, le web sémantique, la bio informatique et bien d’autres, les données à traiter pour y extraire de la connaissance utilisable sont de plus en plus complexes et volumineuses.

On est ainsi conduit à manipuler des données souvent non structurées.

Aussi la fouille de données complexes ne doit plus être considérée comme un processus isolé mais davantage comme une des étapes du processus plus général d’extraction de connaissances dans les bases de données (ECDB). En effet, avant d’appliquer des techniques de fouille de données, les données complexes ont besoin de structuration. De plus anticiper, dès la phase de pré-traitement des données, l’étape de fouille de données ainsi que la notion d’utilité des motifs extraits est également un sujet visé par cet atelier.

Afin de dresser un panorama des travaux récents dans le domaine de la fouille de données complexes, seront particulièrement appréciés (liste non exhaustive) des articles présentant un état de l’art et des perspectives ouvertes dans ce domaine ; des études comparatives de différentes approches de fouille dans des données complexes ou d’approches relatives aux différentes étapes du processus d’ECD dans ce contexte ; la présentation argumentée de nou-

velles approches d'ECD pour la fouille de données complexes ; des descriptions d'applications réelles mettant en jeu un processus de fouille de données complexes.

Une liste de thèmes, non exhaustive, est donnée ci-dessous à titre indicatif :

- Pré-traitement, structuration et organisation de données complexes ;
- Processus et méthodes de fouille de données complexes ;
- Post-traitement ;
- Rôle des Connaissances, Ontologies, Méta données en ECD complexe ;
- Retours d'expériences (Web, sciences du vivant)

2 Responsables

- Martin Arnaud (Laboratoire E3I2, ENSIETA, Brest)
  Email Arnaud.Martin@ensieta.fr
  Tel : 02 98 34 88 84
- Gançarski Pierre (Laboratoire LSIIT Equipe AFD, Strasbourg)
  Email Pierre.Gancarski@lsiit.u-strasbg.fr
  Tel : 03 90 24 45 76

3 Comité de lecture

- Aufaure Marie-Aude (SUPELEC)
- Boussaid Omar (ERIC)
- Desprès Sylvie (LIPN)
- Gançarski Pierre (LSIIT)
- Lefèvre Eric (LGI2A)
- Martin Arnaud (ENSIETA)
- Masseglia Florent (INRIA)
- Osswald Christophe (ENSIETA)
- Petit Jean-Marc (LIRIS)
- Elie Prudhomme (ERIC)
- Trousse Brigitte (INRIA)
- Wemmert Cedric (LSIIT)
- Zighed Djamel (ERIC)

4 Remerciements

Les responsables de l'atelier tiennent à remercier chaleureusement :

- les auteurs pour la qualité de leurs contributions,
- les membres du comité de lecture pour leur travail indispensable à la qualité de cet atelier,
- Florent Masseglia, Alice Marascu et Alzennyr Da Silva, responsables des ateliers pour EGC 2008 pour gentillesse,
- Brigitte Trousse, présidente du comité d'organisation d'EGC 2008 de nous accueillir à Sophia Antipolis.

PROGRAMME

9h30 Accueil et introduction

Session Base de données
- 9h45 Catégorisation des problèmes d'intégration des modèles des cubes de données spatiales, Tarek Sboui, Mehrdad Salehi, Yvan Bédard, Sonia Rivest
- 10h10 Data tube2 : exploration interactive de données temporelles en réalité virtuelle, Florian Sureau, Fatma Bouali, Gilles Venturini
- 10h35 Propriétés d'un modèle conceptuel multidimensionnel pour les données complexes, Nouria Harbi, Omar Boussaid, Fadila Bentayeb

11h-11h30 Pause

Session Classification
- 11h30 Post-Classification d'images texturées par fusion crédibiliste, Hicham Laanaya, Arnaud Martin, Ali Khenchaf, Driss Aboutajdine
- 11h55 Application du Modèle des Croyances Transférables dans le cadre d'expertises en Entomologie Médico-Légale, Gildas Morvan, Alexandre Veremme, David Mercier, Eric Lefèvre
- 12h20 Clustering de trajectoires contraintes par un réseau, Ahmed Kharrat, Karine Zeitouni, Sami Faiz

13h-14h30 Repas

Session Règles d'association
- 14h30 Découverte de relations par croisement d'analyses, Lobna Karoui, Marie-Aude Aufaure
- 14h55 SWAR : Modèle de génération des règles d'associations sémantiques à partir d'une base d'association, Thabet Slimani, Boutheina B.Yaghlane, Khaled Mellouli
- 15h20 Posters

16h-16h30 Pause

- 16h30 Posters

Session Applications
- 16h45 Fouilles archéologiques : à la recherche d'éléments représentatifs, Cyril De Runz, Frédéric Blanchard, Eric Desjardin, Michel Herbin
- 17h10 Détection d'intrusions : de l'utilisation de signatures statistiques, Payas Gupta, Chedy Raïssi, Gérard Dray, Pascal Poncelet, Johan Brissaud

17h35-18h Réunion : Positionnement du groupe de travail Fouille de Données Complexes

18h Clôture

Catégorisation des problèmes d'intégration des modèles des cubes de données spatiales

Tarek Sboui*, Mehrdad Salehi**, Yvan Bédard***, Sonia Rivest****

Chaire industrielle CRSNG en bases de données géospatiales décisionnelles
Centre de recherche en géomatique, 0611 Pavillon Casault, Département des Sciences géomatiques
Faculté de Foresterie et de Géomatique
Université Laval, Québec, Canada,
G1K 7P4
*Tarek.Sboui.1@ulaval.ca
**Mehrdad.Salehi.1@ulaval.ca
***Yvan.Bedard@scg.ulaval.ca
****Sonia.Rivest@scg.ulaval.ca
http://sirs.scg.ulaval.ca/YvanBedard

Résumé. Les cubes de données spatiales sont des types particuliers de bases de données qui sont créées pour des fins d'analyse stratégique ou la découverte de nouvelles connaissances. Ces cubes sont le plus souvent hétérogènes car ils sont construits dans différents contextes. Cette hétérogénéité engendre des problèmes lorsqu'on veut utiliser simultanément plus d'un cube de données spatiales. Malgré l'intérêt croissant porté à l'intégration des bases de données spatiales, aucun travail n'a traité de l'intégration des cubes de données spatiales. Cet article présente une catégorisation des problèmes d'hétérogénéité liés aux modèles (schémas et métadonnées) des cubes de données spatiales. Cette catégorisation permettra ultérieurement à notre équipe d'approfondir l'analyse de l'intégration et l'interopérabilité des cubes de données spatiales.

1 Introduction

Selon Franklin (1992), jusqu'à 80% des données d'une organisation ont une composante spatiale. Les données spatiales sont de plus en plus nombreuses grâce à l'évolution des outils d'acquisition de données (ex. GPS, images satellite, photo aériennes, etc.) et des méthodes de structuration (ex. raster, vecteur) et de représentation (ex. représentations 2D, 3D). De plus, des outils et des méthodes de représentation des données spatiales (ex. des outils de visualisation) ont été développés pour mettre en évidence les caractéristiques spatiales des données (position, forme, taille, orientation, etc.) et les relations qui existent entre elles (ex. intersection, adjacence, etc.) afin de faciliter leur interprétation. Par exemple, les outils de

représentation fournissent une vue cartographique des phénomènes et aident à mieux les comprendre et les analyser (Bédard et al., 2007).

En outre, il y a eu des innovations importantes dans le domaine des technologies de l'information, particulièrement dans les technologies de base de données et des systèmes d'aide à la décision (SAD). Les SAD sont des systèmes d'information qui permettent aux analystes d'effectuer des analyses complexes. En effet, ces outils fournissent des techniques, des données et des solutions qui aident les usagers à identifier et à résoudre les problèmes liés à la prise de décisions stratégiques (Turban et Aronson, 2000). Les SAD utilisent généralement les entrepôts de données qui permettent l'exploration de données actuelles et historiques à différents niveaux d'agrégation (Yougworth, 1995; Rivest et al., 2005). Les entrepôts de données se basent généralement sur une structure multidimensionnelle, ils contiennent alors ce qu'on appelle des « cubes de données ». Les cubes de données facilitent la navigation rapide des données selon différents niveaux de granularité (ex. d'un niveau détaillé à un niveau plus général). Lorsque les cubes de données contiennent des données spatiales (nous les appelons «cubes de données spatiales»), ils permettent à la fois de profiter des avantages de la structure multidimensionnelle et de la représentation cartographique des données spatiales.

Par ailleurs, dans quelques situations, on peut avoir besoin d'utiliser plus d'un cube de données spatiales. Par exemple, supposons que l'on ait besoin d'analyser le risque de feu de forêt, et que l'on dispose de deux cubes de données spatiales : un cube contenant les types de peuplement forestier, et un autre contenant la densité de population dans des régions spécifiques. Une méthode efficace d'utiliser les cubes de données d'une façon homogène est d'intégrer ces cubes de données. L'intégration permet de 1) faciliter l'accès et la réutilisation des données qui proviennent de différentes sources en créant un schéma commun de ces sources et de 2) créer une base de données plus complète en combinant des données de différentes sources afin de répondre à des besoins spécifiques (Shibasaki et al., 1994; Ziegler et Dittrich, 2004).

L'intégration des cubes de données spatiales fait face à des problèmes liés à l'hétérogénéité de ces cubes de données. Par exemple, deux systèmes de référence spatiale différents peuvent être utilisés pour les données spatiales de deux cubes. L'hétérogénéité représente un problème majeur pour l'intégration des données spatiales et, si elle n'est pas traitée, peut engendrer une mauvaise utilisation des données et peut augmenter le risque de mauvaises décisions stratégiques. L'hétérogénéité des cubes de données spatiales peut être au niveau des modèles (schémas et métadonnées, par exemple, différentes significations, différentes références spatiales, etc.) ou bien au niveau des données (les membres des dimensions et les valeurs des mesures, par exemple, différentes représentations géométriques d'un même membre existant dans des cubes différents).

Si certaines études ont tenté d'intégrer des bases de données spatiales (Devogele, 1997; Bishr, 1998; Harvey et al., 1999; Ziegler et Dittrich, 2004; Brodeur, 2004) ou des cubes de données (Frank et Chen, 2005), aucune recherche ne traite néanmoins des cubes de données spatiales.

Dans cet article, nous présentons une catégorisation des problèmes d'intégration (i.e. problèmes d'hétérogénéité) des modèles des cubes de données spatiales et nous en donnons quelques exemples. Dans la prochaine section, nous présentons les cubes de données spatiales et le principe d'intégration de ces cubes de données. Dans la section 3, nous proposons une catégorisation des problèmes qui peuvent se présenter lors de l'intégration de

différents cubes de données spatiales et nous l'illustrons par des exemples. Nous concluons l'article dans la section 4.

2 Les cubes de données spatiales et leur intégration

L'intégration des données est un processus qui vise à combiner les données provenant de différentes sources en créant un schéma commun de ces sources ou en générant une base de données plus complète en combinant des données de différentes sources afin de répondre à des besoins spécifiques.

Dans cette section, nous présentons les caractéristiques des cubes de données spatiales et nous discutons de l'intégration de ces cubes de données.

2.1 Les cubes de données spatiales

Un cube de données est une base de données qui contient des données importées à partir de différentes sources transactionnelles de données. Ces données sont structurées en mesures agrégées selon des dimensions (Thomsen et al., 1999). Une dimension contient des membres qui sont organisés hiérarchiquement selon des niveaux de détail. Les agrégations des mesures sont généralement pré-calculées (en tout ou en partie) à partir des combinaisons possibles des membres et sont optimisées afin de faciliter une recherche rapide d'informations et faciliter le processus de prise de décisions stratégiques.

La structure des cubes de données (i.e. structure multidimensionnelle) est en accord avec le modèle mental de l'usager et, par conséquent, elle est appropriée pour l'exploration de données et pour la prise de décisions stratégiques. Quelques travaux ont montré que la structure multidimensionnelle est plus adaptée à l'analyse stratégique que la structure transactionnelle (Rivest et al., 2005; Bédard et al., 2001; OLAP Council, 1995; Yougworth, 1995). Basés sur la structure multidimensionnelle, les outils exploitant les cubes de données, tels que les outils OLAP (On-Line Analytical Processing) ou SOLAP (OLAP Spatial), peuvent fournir des fonctionnalités intuitives pour interroger des données. Les outils SOLAP utilisent généralement les cubes de données spatiales et appliquent des opérateurs spatiaux de navigation tels que le forage spatial, le remontage spatial et le forage latéral spatial. Le forage spatial permet à l'usager de naviguer d'un niveau général à un niveau plus détaillé dans une dimension spatiale géométrique (ex. visualiser les provinces d'un pays). Le remontage spatial permet de naviguer d'un niveau détaillé des données à un niveau plus général dans une dimension spatiale (ex. visualiser le pays des provinces). Le forage latéral spatial permet de visualiser les différents membres du même niveau dans une dimension spatiale (ex. visualiser les forêts de la province de Québec et celles de l'Ontario) (Bédard et al., 2005).

Dans un cube de données spatiales, les dimensions et les mesures peuvent contenir des composantes spatiales. On peut distinguer trois types de dimensions spatiales : les dimensions spatiales non-géométriques, les dimensions spatiales géométriques et les dimensions spatiales mixtes (Rivest et al., 2003). Dans la dimension spatiale non-géométrique, on considère uniquement les données nominales (par exemple, les noms des rues). Ce type de dimension spatiale ne contient aucune information sur les aspects géométriques et, par conséquent, n'est pas suffisant pour supporter l'analyse spatio-temporelle. Dans une dimension spatiale géométrique, les membres contiennent des formes

géométriques à tous les niveaux de détail (par exemple des polygones qui représentent les rivières de la Province de Québec). Ces formes géométriques sont géographiquement référencées afin de permettre leur visualisation cartographique et leur analyse. Finalement, la dimension spatiale mixte combine des données géométriques et nominales. La figure 1 présente un exemple des trois types de dimensions spatiales.

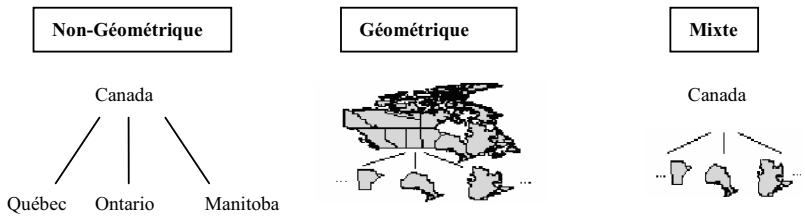


FIG. 1 – *Les trois types de dimensions spatiales (Rivest et al., 2003).*

On distingue deux types de mesures spatiales : géométrique et numérique (Bédard et al., 2007; Tchounikine et al., 2005). La mesure spatiale géométrique est le résultat d'un operateur qui retourne une géométrie (ex. l'intersection des positions d'accidents avec les routes peut être représentée par des points). La mesure spatiale numérique (non géométrique, ex. distance) résulte d'une opération métrique.

2.2 L'intégration de cubes de données spatiales

Après avoir parlé de l'intégration des sources de données en général, nous discutons dans cette section de l'intégration des cubes de données spatiales. Dans un premier temps, nous définissons la notion d'intégration de cubes de données spatiales (le "quoi"). Dans un deuxième temps, nous discutons de l'intérêt d'une telle intégration (l'"intérêt"). Finalement, nous discutons de la raison d'intégrer des cubes de données spatiales (le "pourquoi").

1- Qu'est ce que l'intégration des cubes de données spatiales ? L'intégration des cubes de données vise à créer soit 1) un schéma qui permet l'accès et la réutilisation des données qui existent dans différents cubes, soit 2) un nouveau cube de données spatiales qui combine les données existantes. Le schéma ou le nouveau cube de données spatiales doit être approprié pour l'analyse et la prise de décisions stratégiques au sein des organisations.

2- Quel est l'intérêt de l'intégration des cubes de données spatiales? Dans le domaine de la prise de décision, on peut avoir besoin d'utiliser des données qui existent dans différents cubes de données spatiales, par exemple, lors d'une situation d'urgence. Un exemple d'une situation d'urgence est le désastre naturel (par exemple, un feu de forêt) affectant des pays adjacents. Dans une telle situation, nous pouvons intégrer les données à partir de différents cubes de données spatiales, développés dans ces pays, afin d'obtenir les informations appropriées et intervenir rapidement à différents niveaux (local et national) ou dans différents domaines (géographique, économique, etc.).

3- Pourquoi ne pas intégrer les sources à partir desquelles on a créé les cubes de données spatiales ? On peut distinguer trois raisons d'intégrer les cubes de données spatiales plutôt que les sources de données :

a) Nous n'avons probablement plus accès aux sources de données à partir desquelles les cubes de données ont été créés à cause de différentes raisons (ex. l'entreprise ne donne plus accès aux sources de données).

b) Nous avons besoin des données historiques qui existent généralement uniquement dans les cubes de données spatiales. En effet, dans les sources de données transactionnelles, les données historiques sont généralement modifiées ou remplacées par de nouvelles données avant d'être détruites ou archivées, tandis que les cubes conservent les données historiques pour des fins d'analyse stratégique (Bédard et al., 2001).

c) Dans le contexte de la prise de décision, intégrer des cubes de données spatiales est plus efficace que d'intégrer les sources de données. En effet, dans un cube de données spatiales, contrairement aux sources de données, les agrégations possibles des mesures pour toutes les combinaisons possibles des membres peuvent être pré-calculées en utilisant différents opérateurs (ex. opérateurs spatiaux tels que l'intersection). Ces agrégations exigent généralement un travail laborieux. Par exemple, définir des procédures d'agrégation, définir une nouvelle couche spatiale comme agrégation d'autres couches. La réutilisation des cubes de données nous évite de redéfinir et de reconstruire ces agrégations, et ainsi d'économiser du temps et de l'argent.

Cependant, l'intégration des cubes de données spatiales fait face à des problèmes liés à l'hétérogénéité. Dans la section suivante, nous analysons les problèmes liés à l'hétérogénéité.

3 Les problèmes liés à l'intégration de cubes de données spatiales

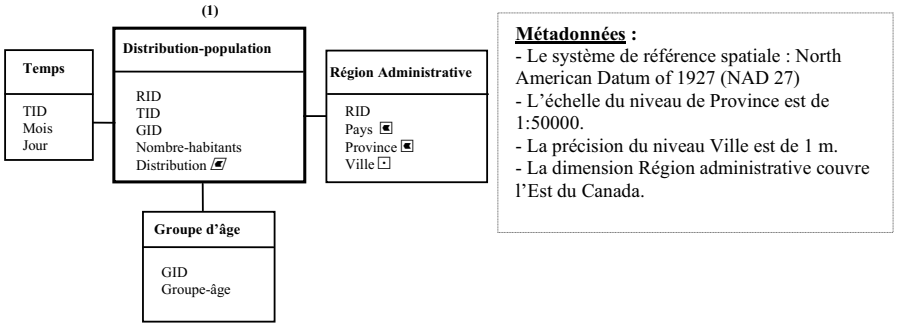
L'intégration de cubes de données spatiales consiste à intégrer les modèles (schémas et métadonnées) ainsi que les données (les membres des dimensions et les valeurs des mesures) de ces cubes de données. L'intégration des modèles fait face à des problèmes d'hétérogénéité au niveau des schémas et au niveau des métadonnées, alors que l'intégration des données fait face à des problèmes d'hétérogénéité des données. Un exemple d'hétérogénéité des données spatiales est la représentation différente d'une même rivière sur deux cartes; elle est représentée par une ligne de 2 cm sur une carte d'échelle 1:1000, alors qu'elle ne l'est que par une ligne de 1 cm sur une autre carte d'échelle 1:5000. Dans cet article, nous nous intéressons à l'intégration des modèles des cubes de données spatiales et nous catégorisons les problèmes liés à cette intégration.

Afin de montrer les problèmes d'hétérogénéité, nous présentons un exemple d'utilisation de deux cubes de données spatiales. Pour déterminer le risque de feu de forêt sur la population menacée, on intègre deux cubes de données spatiales C1 et C2 (modélisés respectivement dans la figure 2(1) et la figure 2(2)). Le premier cube de données spatiales (C1) est utilisé pour déterminer la distribution de la population dans des régions et des périodes données, alors que le deuxième cube de données spatiales (C2) est utilisé pour contrôler le risque de feu de forêt. C1 contient trois dimensions (*Temps, Région Administrative*, et *Groupe d'âge*) et une table de faits (*Distribution-population*) avec une mesure spatiale géométrique (*Distribution*), une mesure numérique (*Nombre-habitants*) et les clés étrangères des dimensions. La dimension *Région Administrative* de C1 contient trois niveaux : *Pays*, *Province* et *Ville*. La dimension *Temps* de C1 contient deux niveaux : *Mois* et *Jour*. La dimension *Groupe d'âge* de C1 contient un seul niveau.

C2 contient cinq dimensions (*Période, Région*, *Facteur environnemental*, *Cours d'eau* et *Forêt*) et une table de faits (*Risque feu*) qui permet de déterminer les endroits vulnérables au feu et analyser le risque de feu de forêt. Cette table contient une mesure numérique (*Degré-risque*) qui indique une évaluation du risque de feu de forêt. La dimension *Période* de C2 contient deux niveaux : *Année* et *Mois*. La dimension *Région* de C2 contient quatre niveaux : *Pays*, *Province, District* et *Ville*. Les dimensions *Cours d'eau*, *Forêt*, et *Facteur Environnemental* contiennent chacune un seul niveau.

Les niveaux spatiaux ainsi que les mesures spatiales sont modélisés à l'aide de pictogrammes spatiaux. Un pictogramme est un symbole qui facilite la représentation des géométries dans le modèle d'une base de données spatiale (Bédard et Larrivée, 2007). Dans notre exemple, nous utilisons les pictogrammes développés dans l'outil de modélisation spatio-temporelle *Perceptory*[1] où le pictogramme «⊡» représente une géométrie de type point, le pictogramme «🗹» représente une géométrie de type ligne, et le pictogramme «◙» représente une géométrie de type polygone. Par exemple, dans le modèle de la figure 2 (1), les géométries des niveaux de la dimension *Région Administrative* de C1 (*Ville*, *Province*, *Pays*) sont modélisées en utilisant respectivement le pictogramme «⊡» et deux pictogrammes «◙». Les pictogrammes en italiques représentent les géométries dérivées. Par exemple, la mesure spatiale *Distribution* est modélisée en utilisant le pictogramme «🗹» indiquant que la distribution de la population par groupe d'âge est dérivée à partir du nombre habitants par région administrative, le temps et le groupe d'âge.

Dans notre exemple, les métadonnées des cubes de données spatiales contiennent des informations à propos du système de référence spatiale, de l'échelle et de la précision des membres de quelques niveaux spatiaux, ainsi que la couverture spatiale de quelques dimensions. Il faut noter que les métadonnées peuvent contenir d'autres informations selon le besoin.



[1] Site web de *Perceptory* : http://sirs.scg.ulaval.ca/perceptory

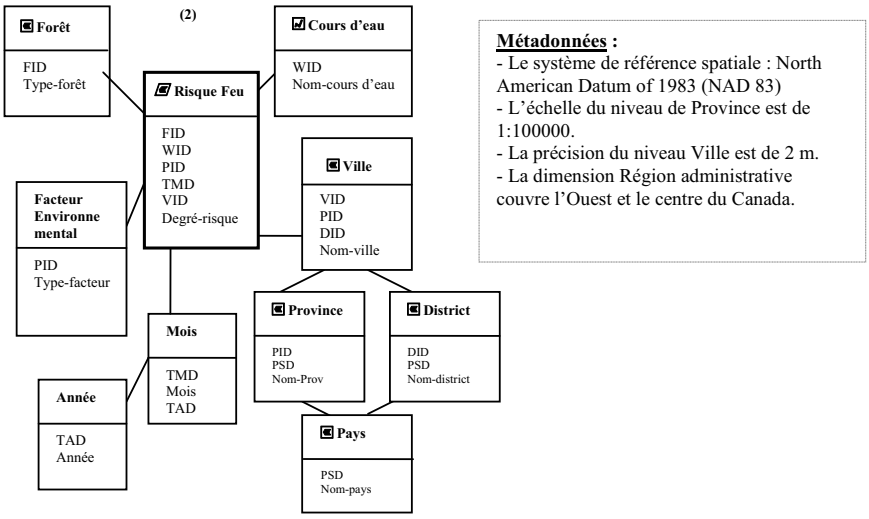


FIG. 2 (1 et 2) – *Deux modèles de cubes de données (C1 et C2).*

Les problèmes d'hétérogénéité des cubes de données spatiales peuvent exister à différents niveaux, notamment au niveau des cubes, des dimensions, des niveaux de dimensions et des mesures. Dans chaque catégorie, l'hétérogénéité peut être observée au niveau des schémas ou au niveau des métadonnées des cubes de données spatiales. Ainsi, nous proposons pour la première fois quatre catégories d'hétérogénéité qui peuvent exister entre les cubes de données spatiales.

1. Hétérogénéité Cube-à-Cube
   - *Hétérogénéité au niveau du schéma*. Elle apparaît quand deux cubes ont deux schémas différents (ex. schéma en étoile vs schéma flocon (différent types de schémas) ou schéma simple vs schéma mixte) ou ont des faits qui sont représentés selon différents nombres de dimensions ou mesures. Dans notre exemple de la figure 2, le cube de données spatiales C1 est modélisé avec un schéma en étoile, alors que le cube C2 est modélisé avec un schéma en flocon.
   - *Hétérogénéité au niveau des métadonnées* :
     - *Différence de systèmes de référence spatiale.* Il y a différents systèmes de référence spatiale qui peuvent être utilisés pour déterminer la position des objets spatiaux (ex. le système de coordonnées ellipsoïdal global (latitude-longitude-hauteur standard) et le système de cordonnées x,y,z). Dans notre exemple, le cube de données C1 est basé sur le système *North American Datum of 1927 (NAD 27)*, tandis que le cube de données C2 est basé sur le système *North American Datum of 1983 (NAD 83).*
2. Hétérogénéité Dimension-à-Dimension
   - *Hétérogénéité au niveau du schéma* :

- *Différence de hiérarchies*. Elle apparaît quand des dimensions sémantiquement liées[2] ont différentes hiérarchies[3]. Par exemple, la dimension *Région Administrative* dans le cube de données spatiales C1 est représentée par une hiérarchie spatiale simple (Pays, Province et ville), alors que dans le cube de données C2, la dimension *Région* (Pays, Province, District et ville) est représentée par une hiérarchie spatiale différente.

- *Hétérogénéité au niveau des métadonnées* :
  - *Couverture spatiale.* Elle apparaît quand deux dimensions sémantiquement liées ont des couvertures territoriales différentes. Par exemple, la dimension *Région Administrative* du cube de données spatiales C1 couvre l'Est du Canada, tandis que dans le deuxième cube de données C2, la dimension *Région* (Pays, Province, District et ville) couvre l'Ouest et le centre du Canada.

3. Hétérogénéité Niveau-à-Niveau

- *Hétérogénéité au niveau du schéma* :
  - *Différence de primitives géométriques*. Elle apparaît lorsque, dans deux cubes de données spatiales, des niveaux sémantiquement liées ont différentes primitives géométriques. Par exemple, dans le cube de données C1, le niveau *Ville* indique que chaque ville membre est représentée par un point, alors que dans le cube de données C2, le même niveau *Ville* indique que chaque ville membre est représentée par un polygone.
- *Hétérogénéité au niveau de métadonnées* : deux exemples
  - *Hétérogénéité au niveau de la résolution de représentation*. Elle apparaît lorsque des niveaux sémantiquement liés sont représentés avec un niveau de détail géométrique correspondant traditionnellement à différentes échelles cartographiques. Dans notre exemple, la résolution de représentation des détails cartographiques du niveau *Province* du cube de données C1 est typique d'une carte topographique à l'échelle 1:50000, alors que la résolution géométrique du niveau *Province* du cube de données C2 équivaut aux détails d'une carte topographique 1:100000.
  - *Hétérogénéité au niveau de la précision*. Elle apparaît lorsque des niveaux sémantiquement liés des cubes de données sont représentés par des géométries localisées avec des précisions différentes. Dans notre exemple, les niveaux *Ville* dans les deux cubes n'ont pas la même précision (précision de 1 m dans le cube de données C1 et précision de 2 m dans le cube de données C2), ce qui entraîne inévitablement des problèmes de chevauchement de frontières.

[2] Deux éléments de deux cubes de données spatiales (i.e. deux dimensions, deux niveaux ou deux mesures) sont sémantiquement liés lorsqu'ils représentent différemment les mêmes phénomènes spatiaux.

[3] Malinowski et Zimányi (2004) ont présenté une catégorisation des hiérarchies de dimensions spatiales.

4. Hétérogénéité Mesure-à-Mesure
- *Hétérogénéité au niveau du schéma* : Elle apparaît lorsque deux mesures géométriques sémantiquement liées ont différentes primitives géométriques (ex. ligne vs polygone).
- *Hétérogénéité au niveau de métadonnées* :
    - *Hétérogénéité au niveau d'agrégation*. Apparaît lorsque différentes fonctions ont été utilisées pour agréger des mesures spatiales sémantiquement liées (ex. union géométrique vs centre de gravité).

Ce qu'il faut noter ici, c'est que les hétérogénéités au niveau des schémas sont de type structurel et trouvent des solutions dans le design intelligent du cube résultant de l'intégration alors que les hétérogénéités au niveau des métadonnées ne font qu'indiquer des problèmes affectant directement le contenu, i.e. les données elles-mêmes. Dans le cas particulier des données spatiales, la solution aux problèmes d'hétérogénéité au niveau des métadonnées peut être facile à résoudre dans certains cas grâce aux logiciels spécialisés en géomatique (ex. changement de datum de référence), alors que dans d'autres cas, l'hétérogénéité est difficile, voire même impossible à résoudre (ex. passer d'une référence spatiale 2D non-contrôlée pour la distorsion due au relief à une référence spatiale 3D, comme cela est fréquemment le cas avec les données routières). La plupart du temps, l'hétérogénéité est difficile à automatiser, et fait appel à tout un domaine d'expertises spécialisées, soit la géomatique (incluant la géodésie et GPS (Global Positioning Systems), la photogrammétrie, la télédétection, la topométrie, l'hydrographie, la cartographie et les SIG (Systèmes d'Information Géographique)). Seuls quelques exemples ont été mentionnés ci-avant mais les sources d'hétérogénéité peuvent se compter par plusieurs centaines: systèmes de référence spatiale, méthodes de positionnement dans ces systèmes, instrumentations d'acquisition, méthodes d'observations, algorithmes de pré-traitement, structures propriétaires des logiciels traitant des données géométriques, structuration applicative des bases de données spatiales, algorithmes de transformation et de traitement de la donnée, modes de présentation des données, généralisation cartographique, etc. D'ailleurs, plusieurs ouvrages de référence existent et sont facilement disponibles dans chacune des disciplines utilisées en géomatique. Nous encourageons donc le lecteur à pousser davantage cette réflexion en consultant ces ouvrages puisqu'il va au-delà de l'objectif du présent article de présenter les solutions à ce problème d'hétérogénéité des cubes de données spatiales.

4 Conclusion

Les données spatiales sont considérées comme des données complexes. Ces données facilitent la compréhension et l'interprétation des phénomènes du monde réel. Une fois stockées dans des cubes, les données spatiales peuvent guider le processus d'analyse et facilitent la prise de décisions stratégiques au sein des organisations.

Les cubes de données spatiales sont de plus en plus utilisés par les systèmes d'aide à la décision. Cependant, ces cubes sont généralement modélisés différemment d'une organisation à une autre, ou même d'un concepteur à un autre (i.e. modèles hétérogènes). Dans cet article, nous avons proposé une catégorisation des problèmes d'hétérogénéité en tenant compte des différents éléments des cubes de données spatiales. Conséquemment, nous avons défini quatre catégories, notamment l'hétérogénéité *Cube-à-Cube*, l'hétérogénéité *Dimension-à-Dimension*, l'hétérogénéité *Niveau-à-Niveau* et l'hétérogénéité *Mesure-à-*

Mesure. Pour chaque catégorie, nous avons considéré les différentes composantes des modèles de cubes de données, notamment le schéma et les métadonnées. Nous avons donné quelques exemples associés à l'hétérogénéité des cubes de données spatiales. La catégorisation des problèmes d'hétérogénéité constitue une démarche nécessaire pour nos autres recherches dans le domaine de l'intégration et de l'interopérabilité des cubes de données spatiales. En effet, nous sommes en train de définir une approche qui permet d'accroître le niveau d'automatisation pour la résolution des problèmes liés à l'hétérogénéité des modèles des cubes de données spatiales. Cette approche se base sur les principes fondamentaux de la communication humaine, de la gestion du risque (de mauvaise interprétation) et de la gestion de contexte dans lequel les éléments des modèles ont été définis et peuvent être utilisés. En effet, nous représentons les cubes par des agents qui communiquent ensemble afin de 1) comprendre les éléments des modèles des cubes, 2) résoudre les conflits liés à l'hétérogénéité de ces modèles, et 3) faire appel à l'expert humain lorsque requis. Afin d'aider les agents, nous représentons explicitement les éléments du contexte et nous en inférons d'autres qui peuvent être utiles pour la compréhension des éléments des modèles, pour la résolution des conflits d'hétérogénéités et pour l'évaluation du besoin d'intervention humaine par un expert.

Remerciements

Les auteurs tiennent à remercier les organisations suivantes pour le financement de la Chaire industrielle en bases de données géospatiales décisionnelles: Conseil de Recherche en Sciences Naturelles et en Génie du Canada, Recherche et Développement Défense Canada, Hydro-Québec, DVP, Intélec Géomatique, Holonics, KHEOPS Technologies, Syntell, Ressources Naturelles Canada, Transports Québec, Université Laval.

Références

Ballard, C., D. Herreman, D. Schau, R. Bell, E. Kim et A. Valencic (1998) Data Modeling Techniques for Data Warehousing, *IBM Redbook*, IBM International Technical Support Organization, Feb-26-1998, ISBN No. 0738402451.

Bédard,Y., M.J. Proulx, et S. Rivest (2005) Enrichissement du OLAP pour l'analyse géographique : exemples de réalisation et différentes possibilités technologiques, Revue des Nouvelles Technologies de l'Information, Cépaduès-Éditions, France, pp. 1-20.

Bédard, Y. et S. Larrivée (2007) Spatial Databases Modeling with Pictogrammic Languages, Dans: *Encyclopedia of Geographic Information Sciences*, Springer, NY.

Bédard, Y., S. Rivest, et M.J. Proulx (2007) Spatial On-Line Analytical Processing (SOLAP): Concepts, Architectures and Solutions from a Geomatics Engineering Perspective, Dans: Robert Wrembel & Christian Koncilia (ed(s)), Data Warehouses and OLAP : Concepts, Architectures and Solutions, Chap. 13, IRM Press (Idea Group), London, UK, pp. 298-319.

Bédard, Y., T. Merrett, et J. Han (2001) Fundamentals of spatial data warehousing for geographic knowledge discovery. *Geographic Data Mining and Knowledge Discovery.* In: Miller, H.J., Han, J. (eds.)

Bishr, Y. (1998) Overcoming the semantic and other barriers to GIS interoperability. Int. J. Geographical Information science. 12, 299- 314.

Brodeur, J. (2004) *Interopérabilité des données géospatiales: élaboration du concept de proximité géosémantique*. Thèse de doctorat. Université Laval.

Devogele, T. (1997) *Processus d'intégration et d'appariement de bases de données géographiques – Application à une base de données routières multi-échelles.* Thèse de doctorat, Université de Versailles.

Frank, S.C. et C.W Chen (2005) Integrating Heterogeneous Data Warehouses Using XML Technologies. Journal of Information Science, Vol.31, No. 3, pp. 209-229. (SSCI).

Franklin, C. (1992) An introduction to geographic information systems: linking maps to databases. Database, vol. 15, no. 2, pp.13-21.

Harvey, F., W. Kuhn, H. Pundt, Y. Bishr, et C. Riedemann (1999) Semantic Interoperability: A Central Issue for Sharing Geographic Information Annals of Regional Science. Special Issue on Geo-spatial Data Sharing and Standardization. 213-232

Malinowski, E. et E. Zimányi (2005) Spatial Hierarchies and Topological Relationships in the Spatial MultiDimER model. In *Proc. of the 22nd British Nat. Conf. on Databases, BNCOD22*, number 3567 in LNCS, pages 17-28, Sunderland, UK, July 2005. Springer-Verlag.

OLAP Council. (1995) OLAP and OLAP Server Definitions http://www.olapcouncil.org/research/glossary.htm.

Rivest, S., Y. Bédard, M.J. Proulx, et M. Nadeau (2003) SOLAP: a new type of user interface to support spatio-temporal multidimensional data exploration and analysis. ISPRS Joint Workshop of WG II/5, II/6, IV/1 and IV/2 on Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis, Quebec, Canada.

Rivest, S., Y. Bédard, M.J. Proulx, M. Nadeau, F. Hubert, et J. Pastor (2005) SOLAP Technology: Merging Business Intelligence with Geospatial Technology for Interactive Spatio-Temporal Exploration and Analysis of Data. J. ISPRS Advances in spatio-temporal analysis and representation, 17–33.

Shibasaki, R., T. Itoh, et Y. Honda (1994) Integration of remote sensing and ground observation data for generating a global climate data set. *Canadian Journal of Remote Sensing*, 20: 435-442.

Tchounikine, A., M. Miquel, R. Laurini, T. Ahmed, S. Bimonte, et V. Baillot (2005). Panorama de travaux autour de l'intégration de données spatio-temporelles dans les hypercubes. *Revue des nouvelles technologies de l'information*, B-1, 21-33.

Thomsen, E., G. Spofford, et D. Chase (1999) *Microsoft OLAP Solutions,* John Wiley & Sons, 495 p.

Turban, E., et J.E. Aronson (2000) *Decision support systems and intelligent systems* (6th ed.), Prentice Hall, NJ.

Yougworth, P. (1995) OLAP Spells Success For Users and Developers. Data Based Advisor, December, p. 38-49.

Ziegler, P. et K.R. Dittrich (2004) Three decades of data integration - all problems solved? In 18th IFIP World Computer Congress (WCC 2004), IFIP International Federation for Information Processing, pages 3–12, Toulouse, France.

Summary

Spatial data cubes are one of the fundamental components of decisions support systems. These data cubes are quite often modelled differently. This difference (i.e. heterogeneity) is the main barrier for integrating data from different spatial data cubes. The integration of spatial databases has been the subject of several research works. However, no research dealing with spatial data cubes integration has been found. In this paper, we propose a categorization of the heterogeneity problems related to spatial data cubes models (schemas and metadata). This categorization will guide further analysis of spatial data cubes integration and interoperability.

DataTube2 : exploration interactive de données temporelles en réalité virtuelle

Florian Sureau*, Fatma Bouali**, Gilles Venturini*

*Laboratoire d'Informatique, Université François-Rabelais de Tours
64 avenue Jean Portalis, 37200 Tours, France
florian.sureau@gmail.com, venturini@univ-tours.fr
**Université de Lille2 -IUT, Dpt STID
25-27 Rue du Maréchal Foch, 59100 Roubaix, France
Fatma.Bouali@univ-lille2.fr

Résumé. Nous nous intéressons dans cet article à la fouille visuelle de données temporelles mises sous la forme de n attributs dont les valeurs sont enregistrées pendant k instants. Après un état de l'art sur les différentes approches de visualisations de telles séries, nous présentons plus particulièrement une approche ayant reçue encore peu d'attention ("DataTube"). DataTube place les données dans un tube dont l'axe représente le temps. Nous étendons ensuite cette approche : tout d'abord nous définissons plusieurs modes de visualisations (couleurs, formes, etc) et nous ajoutons un axe temporel. Ensuite nous introduisons des interactions avec la possibilité de sélectionner des attributs et des instants. Nous ajoutons une étape de classification non supervisée afin de regrouper dans la visualisation les attributs similaires. Enfin nous intégrons cette visualisation dans notre plateforme de fouille de données en réalité virtuelle VRMiner, avec un affichage stéréoscopique sur grand écran et des possibilités de navigation interactive. Nous appliquons cette visualisation sur plusieurs ensembles de données réelles et nous montrons qu'elle peut gérer 1,5 million de valeurs.

1 Introduction

Nous nous intéressons dans ce travail au problème de la fouille visuelle de données temporelles. La dimension temps intervient dans de nombreux problèmes de fouille de données et elle a donné lieu à la définition de différents problèmes (analyse, prédiction, modélisation) et à l'étude de multiples méthodes (Antunes et Oliveira (2001)). Parmi ces méthodes, certaines font appel à des procédés d'extraction des connaissances entièrement automatiques, d'autres au contraire vont plutôt faire appel à des visualisations. Ces visualisations peuvent être utilisées en prétraitement pour permettre à l'expert de mieux comprendre les données, ou encore en post-traitement pour analyser visuellement les résultats d'une méthode de fouille, ou même de manière interactive pour découvrir des connaissances.

Parmi l'importante variété des données temporelles (séries numériques ou symboliques, suite d'événements, textes, images, sons, etc), nous allons plus précisément considérer le cas de n attributs numériques prenant des valeurs sur k instants et décrivant, à chaque instant t

considéré, l'évolution d'un phénomène donné. Les objectifs de l'expert que nous considérons sont par exemple : observer simultanément l'évolution de tous les attributs, détecter les valeurs manquantes, les événements particuliers et les périodicités éventuelles, les dépendances entre attributs et notamment les attributs se comportant de manière similaire. En outre, nous souhaitons pouvoir afficher de grands volumes de données. A titre d'exemple, il peut s'agir de cours de la bourse (une variable représente un cours), de consommation (chaque variable décrit la consommation d'un produit) ou encore de données médicales (appareils de mesure).

Parmi les visualisations utilisées pour les données temporelles, il en existe une en particulier appelée "DataTube" (Ankerst (2001a)) qui a recu peu d'attention et qui pourtant nous paraît tres prometteuse. Comme nous allons le voir, DataTube utilise une approche orientée pixel pour la visualisation (Ankerst (2001b)) qui associe une valeur à chaque pixel. Ces méthodes font partie de celles qui peuvent représenter de très grands volumes de données. Par ailleurs, DataTube est comme un "tube temporel" en 3D ce qui la destine bien à être utilisée dans un environnement de réalité virtuelle. Nous avons donc décidé d'étendre cette visualisation, de l'intégrer dans notre plateforme VRMiner (Azzag et al. (2005)), et de la tester sur des volumes de données beaucoup plus grands qu'auparavant.

La suite de cet article est organisée ainsi : dans la section 2 nous détaillons les approches de visualisation et de fouille visuelle de données temporelles, et plus particulièrement la visualisation DataTube. Dans la section 3 nous détaillons les extensions menant à la définition de DataTube2. Dans la section 4, nous présentons les résultats expérimentaux sur des données réelles. Enfin nous concluons et présentons des perspectives dans la section 5.

2 Visualisation de données temporelles et principes de DataTube

Le domaine des visualisations temporelles existe depuis longtemps (Minard (1861)) (voir par exemple un survol dans (Muller et Schumann (2003)). Nous commençons donc par considérer le cas d'une séquence de symboles avec la visualisation "Arc Diagrams" (Wattenberg (2002)). Cette visualisation permet de détecter des motifs répétitifs mais elle ne visualise qu'une seule séquence de symboles ($n = 1$ dans notre notation). Une visualisation classique pour les données temporelles sont les spirales (Carlis et Konstan (1998), Weber et al. (2001)). Le centre de la spirale représente l'origine du temps. Ensuite le rayon de la spirale augmente mais un même intervalle de temps T est toujours traité en un seul tour de spirale. Ces méthodes peuvent visualiser des valeurs symboliques (Weber et al. (2001) pour $n = 1$). Elles peuvent visualiser en 2D deux séries (Weber et al. (2001)). Dans (Carlis et Konstan (1998)) plusieurs séries peuvent être visualisées en passant à une représentation 3D : jusqu'à 12 attributs visualisés sous forme d'histogrammes placés sur une spirale 2D, et jusqu'à 112 attributs en utilisant un empilement 3D de spirales 2D. Les spirales permettent de détecter des périodicités en ajustant interactivement l'intervalle T.

Dans les domaines où les échelles temporelles le permettent, la métaphore du calendrier ou de l'agenda fait partie des représentations classiques pour des événements ordonnés dans le temps. Ainsi nous pouvons citer par exemple Daassi et al. (2000), mais également van Wijk et van Selow (1999) où une étape de classification est effectuée pour rassembler les jours où la variable observée se comporte de manière identique, ou encore (Ankerst et al. (1996a)) où

chaque événement fait l'objet d'un pixel d'une couleur donnée et où les jours du calendrier sont ensuite remplis par tous les événements ayant eu lieu ce jour là. Notons que la métaphore du crayon (l'axe du crayon représente le temps, et les facettes du crayon servent à visualiser l'évolution de variables) peut être utilisée notamment dans la visualisation de données sociologiques (Francis et Pritchard (2003)).

Les plus grands volumes de données temporelles sont visualisés soit par des méthodes effectuant une étape de classification comme dans (Hébrail et Debregeas (1998)) où 2665 courbes de 144 valeurs chacune sont regroupées en 100 classes avec une carte de Kohonen (chaque classe est visualisée sous la forme d'une courbe à 144 valeurs), ou encore dans les méthodes orientées pixels comme "Recursive Pattern" (Keim et al. (1995)) où 530000 valeurs sont visualisées, ou encore "Circle segments" (Ankerst et al. (1996b)), ou bien comme "DataJewel" (Ankerst et al. (1996a)). Nous nous sommes donc intéressés plus particulièrement aux travaux de ces auteurs. En ce qui concerne les grands volumes de données, citons la visualisation "Time Tube" (Chi et al. (1998)) qui visualise spécifiquement les accès à un site Web (7588 pages) avec une représentation arborescente (l'arbre représente les pages et l'épaisseur des arcs représente le nombre d'accès).

La visualisation "DataTube" (Ankerst (2001b)) peut être classée dans les méthodes orientées pixels même si elle s'en distingue par une différence importante provenant de l'utilisation de la 3D. Les valeurs de la matrice Attributs$\times$Temps sont représentées par des codes de couleur, et les deux bords de cette matrice qui correspondent au temps sont repliés l'un sur l'autre pour former un tube. L'axe du tube représente donc l'axe temporel, et une "couronne" sur le tube représente la valeur des attributs pour l'instant considéré. "DataTube" a été appliquée à des données boursières (50 cours). Ce type de représentation axiale du temps a été utilisée également dans une méthode appelée "Kiviat tube" (Hackstadt et Malony (1994)) où chaque instant de l'axe est un graphique de type "Star coordinates" (Kandogan (2000)). Cette visualisation a été appliquée à la visualisation de la charge de 64 processeurs, mais nous ne l'avons pas retenue car elle peut générer des occlusions. Après avoir contacté M. Ankerst, nous avons appris que DataTube n'avait pas été plus développée. Pourtant, elle comporte des avantages et des potentiels comme nous allons le montrer dans la suite de l'article.

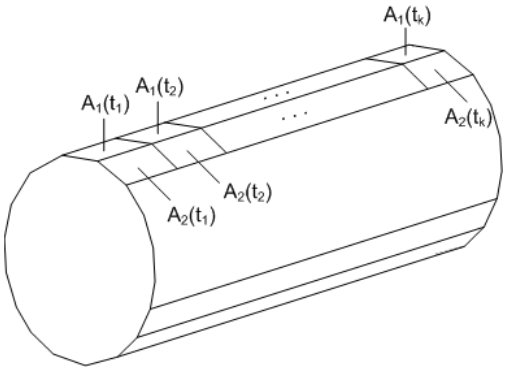
3 DataTube2

3.1 Définition de la visualisation DataTube2 et de ses interactions

Nous notons par $A_1, ..., A_n$ les n attributs numériques décrivant les données. Nous supposons que les valeurs de ces attributs ont été enregistrées initialement dans un intervalle de temps $[0, T]$, de manière non synchrone. Ensuite nous pouvons définir des échelles temporelles différentes (heure, jour, mois, etc) et regrouper les instants dans des intervalles en effectuant la somme des valeurs des attributs dans cet intervalle. Notons que d'autres opérateurs "d'agrégation" seraient possibles comme par exemple la moyenne. Ces opérations de prétraitement sont incluses dans notre outil mais nous ne les détaillerons pas plus dans la suite : nous notons directement ces instants/intervalles de mesure par $t_1, ..., t_k$. Les données fournies à l'entrée de notre méthode de visualisation sont donc représentées par une matrice $n \times k$ et nous notons par $A_i(t_j)$ la valeur prise par l'attribut A_i à l'instant t_j. Les valeurs d'un attribut A_i sont ensuite normalisées sur un intervalle $[0, 1]$ en considérant, à la demande de l'utilisateur, soit les

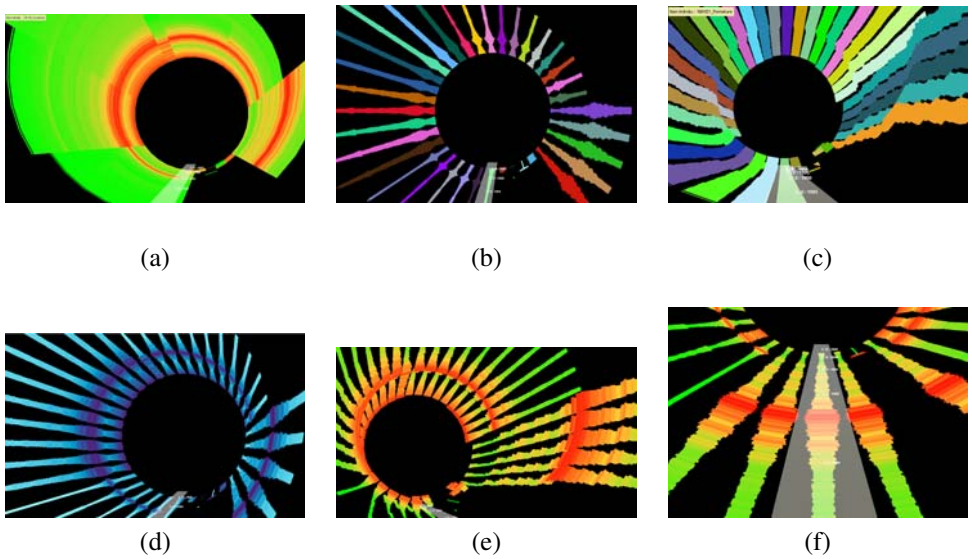
	t_1	t_2	...	t_k
A_1	$A_1(t_1)$	$A_1(t_2)$	...	$A_1(t_k)$
A_2	$A_2(t_1)$	$A_2(t_2)$	...	$A_2(t_k)$
...	...	...	...	...
A_n	$A_n(t_1)$	$A_n(t_2)$	...	$A_n(t_k)$

(a)



(b)

FIG. 1 – *Matrice des données en (a) et définition du tube temporel en (b).*



(a) (b) (c)

(d) (e) (f)

FIG. 2 – *Représentation des données par couleur (a), largeur (b), hauteur (c), en combinant couleur et largeur (d), couleur, largeur et hauteur (e), sur les données indices boursiers. En (f) est représenté l'axe temporel.*

minimum et maximum de A_i, soit les minimum et maximum globaux, suivant la nature des données.

La visualisation de base dans DataTube (et DataTube2) consiste donc à représenter cette matrice sous la forme d'un tube temporel comme illustré sur la figure 1. Dans cette visualisation, chaque valeur $A_i(t_j)$ est donc représentée par une facette rectangulaire. Un instant t_j est donc codé par un ensemble de facettes formant une couronne. L'évolution d'un attribut A_i est représentée par une ligne parallèle à l'axe du tube. Une couleur est attribuée à chaque facette en fonction de la valeur de $A_i(t_j)$. Les valeurs manquantes (qui peuvent simplement refléter l'absence d'événement dans l'intervalle de temps considéré) sont représentées par défaut dans une couleur donnée (noire par exemple dans toutes nos visualisations).

Les premières propriétés de cette visualisation sont les suivantes. Son mode de représentation est facilement compréhensible par des utilisateurs non spécialistes en fouille de données (d'après les présentations que nous avons pu en faire avec de vrais experts du domaine, voir section 4). Tous les attributs sont visualisés simultanément ce qui permet de les comparer entre eux, mais aussi de détecter les valeurs manquantes. L'axe du tube représente le temps : la perception de l'écoulement du temps est donc très intuitive (nous verrons dans la section 3.3 que l'utilisation d'un écran stéréoscopique y contribue de manière significative). En naviguant à l'intérieur du tube, la perspective donne des effets visuels très intéressants : les données temporellement proches de la position de l'utilisateur dans l'espace 3D sont agrandies par rapport à celles qui se situent plus loin dans le temps et cet effet est accentué par la stéréoscopie pour l'oeil de l'utilisateur. Il en résulte donc un effet de "focalisation" (zoom) sur les instants proches et de conservation du contexte sur les instants suivants plus éloignés.

Dans DataTube2, nous avons ajouté d'autres fonctionnalités. Tout d'abord, nous proposons d'autres modes de visualisation des valeurs $A_i(t_j)$ (voir figure 2) :

- par couleur : $A_i(t_j)$ est représentée par une face de taille constante dont seule la couleur va varier. Pour fixer la couleur, l'utilisateur peut choisir 3 couleurs qui correspondent aux valeurs minimum, médiane et maximum des données (par exemple Vert/Orange/Rouge). Le choix des couleurs peut dépendre du domaine traité,
- par largeur : la largeur des facettes représente les $A_i(t_j)$. L'utilisateur peut fixer l'amplitude maximale de cette largeur (par rapport à la place disponible sur le tube pour A_i),
- par hauteur : la hauteur des facette varie en fonction de $A_i(t_j)$, ce qui peut donner des effets d'ondulation. Par contre, cela peut engendrer aussi des occlusions, et l'utilisateur a la possibilité dans ce cas de contrôler l'opacité des facettes,
- en fil de fer : les facettes ne sont pas colorées et seules les contours apparaissent.

Ces modes de visualisation peuvent êtres combinés et permettent, suivant le type de données et le domaine traité, de faire apparaître des informations différentes, en particulier des combinaisons comme couleur et largeur, etc, voir figure 2(d) et (e).

Nous donnons ensuite la possibilité à l'utilisateur d'ajouter un axe temporel pour représenter explicitement le temps, ce qui n'existait pas dans DataTube. Cet axe prend la forme d'un "chemin" composé de dalles, où chaque dalle représente un instant t_j (voir figure 2(f)). Ce chemin est placé initialement en bas et à l'intérieur du tube. Les dalles sont transparentes de manière à laisser percevoir les données situées en dessous d'elles. De plus, un label de type texte indique périodiquement à quel instant correspond une dalle. Enfin, ces dalles sont cliquables pour sélectionner un instant donné et le mettre en valeur par rapport aux autres.

En ce qui concerne l'interaction avec la visualisation, nous proposons deux modes :
– pour les bases telles que $n \times k \leq 80000$, chaque facette est cliquable. Un clic gauche permet d'afficher en haut et à droite de l'écran le nom de l'attribut, l'instant considéré et la valeur $A_i(t_j)$. Un clic droit permet de mettre en avant (vers l'axe du tube) l'ensemble des facettes correspondant à A_i,
– pour les autres bases, notre implémentation en Java3D ne permet pas de gérer individuellement autant de facettes. Les facettes sont donc regroupées par ligne, et forment un seul objet 3D pour chaque attribut A_i. Un clic donne donc le nom de l'attribut.

3.2 Regroupement des attributs similaires

Pour améliorer la visualisation et la perception de comportements similaires entre les attributs, nous proposons de réorganiser l'ordre des n attributs autour du tube. Cette réorganisation a pour but de représenter de manière proche sur le tube des attributs ayant des valeurs proches. De nombreuses méthodes de réorganisations de matrices existes, notamment Climer et Zhang (2006). Ici nous utilisons une méthodes simple basée sur une distance entre deux attributs, de la manière suivante :

$$dist(A_i, A_j) = \sqrt{\sum_{t=1}^{k} [A_i(t) - A_j(t)]^2}$$

L'heuristique mise en place est la suivante : elle part d'une liste L d'attributs qui est initialement vide. Elle commence par sélectionner les deux attributs les plus proches selon la distance euclidienne définie précédemment et elle les ajoute à L. Ensuite, elle choisit parmi les attributs restants celui qui est le plus proche du premier ou du dernier élément de L, et elle insère dans L l'attribut à l'extrémité ainsi déterminée. Une fois que tous les attributs sont placés, le tube est réorganisé suivant l'ordre indiqué par L. Cette heuristique est déterministe et sa complexité est en $O(n^2)$.

3.3 Intégration dans VRMiner

VRMiner est une plateforme de fouille visuelle de données en 3D et en réalité virtuelle développée dans notre équipe. Elle comprend un écran large stéréoscopique (écran polarisé) pour plusieurs utilisateurs et du matériel pour l'interaction (Flock of bird, SpacePilot, gants de données) (Azzag et al. (2005)). L'intégration de DataTube2 dans VRMiner permet d'améliorer la visualisation et d'augmenter les interactions et l'immersion dans les données. Ainsi, à l'aide de la visualisation stéréoscopique, on constate que la perception de la profondeur est importante en particulier pour bien appréhender l'écoulement du temps. Sans la profondeur, notre visualisation ressemble sur un écran 2D à un disque et seuls les déplacements permettent d'avoir une idée de la forme tubulaire.

Dans le monde virtuel, la navigation a lieu à l'aide du SpacePilot (6 degrés de liberté) qui à l'avantage d'être complet au niveau des mouvements et qui permet d'utiliser conjointement la souris pour pointer dans la visualisation. L'utilisateur est placé initialement devant le tube, avec une orientation parallèle à l'axe, et devant le chemin de l'axe temporel. Dans le mode "Walk", il peut librement se déplacer le long de l'axe, en suivant le chemin par exemple. Il peut apercevoir le déroulement global des données vers le fond de la scène, mais aussi observer sur les cotés les

Bases	# valeurs	n	k	Temps d'exécution [E-T]
Site Web	1481088	464	133 jours (ou 3192 h)	5917 ms [23]
Indices boursiers	244225	6	57 années	20232 ms [155]
EECG	67136	64	1049	7329 ms [73]

TAB. 1 – *Bases de données réelles testées avec les paramètres de visualisation (n, k) et les temps d'exécution (algorithme de réorganisation + visualisation).*

données en gros plan. L'axe de rotation du SpacePilot est utilisé pour faire pivoter l'utilisateur sur l'axe du tube : cela donne l'impression que le tube tourne autour de son axe, et il est ainsi possible de placer certaines données en haut et de même pour l'axe temporel.

4 Résultats

4.1 Données réelles étudiées

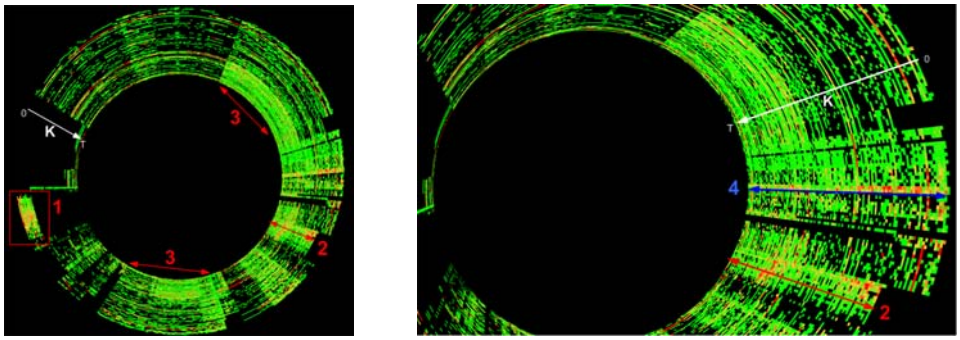
Nous avons appliqué notre visualisation à trois bases de données réelles concernant un site Web, des indices boursiers, des données d'electro-encéphalographies ainsi que des données de consommation (dont nous ne parlerons pas par accord de confidentialité). Les caractéristiques de ces données sont représentées dans le tableau 1. Le premier jeu de données étudié est une série de fichiers logs du site Web de Polytech'Tours. Nous visualisons ainsi pour chaque instant, le nombre de visites reçues pour chaque page du site. Un deuxième jeu de données représente l'évolution d'indices boursiers nationaux depuis leur création. Le troisième jeu de données porte sur des électro-encéphalogrammes. Nous avons réalisé les tests sur un MacBook Pro (2.4GHZ Intel Core Duo, 4Go de RAM) pour obtenir les résultats de la dernière colonne du tableau 1 sur 20 essais en moyenne.

4.2 Accès à un site Web

Les données contiennent ici un ensemble de 464 attributs correspondants aux pages du site (de type htm, html, php, aspx), les logs extraits ont donc étés épurés afin de réduire le nombre d'attributs et de ne garder que les éléments pertinents. Les impacts sur ces pages sont mesurés sur une durée de 133 jours, au final 1481088 impacts ont été enregistrés sur cette période. La période étudiée est étalée du mois de juin au mois d'octobre : on peut ainsi étudier l'influence de la fin de la période scolaire, des vacances et de la rentrée des étudiants sur les visites du site.

La figure 3 montre l'évolution du nombre de visites des pages Web en fonction du temps, caractérisée par la couleur des données (nombre d'impacts linéairement croissant de la couleur verte à la couleur rouge), avec la journée comme unité de temps :

- L'annotation *1* de la figure 3-b montre que les pages concernant les "PIP" (Parcours Ingénieurs Polytech) ont été beaucoup visitées début juin puis de moins en moins jusqu'à la disparition des pages sur le site, pour devenir les "PEIP" (Parcours des Ecoles d'Ingénieurs Polytech).
- Ces pages "PEIP", annotées *2* sur les figures 3(a) et 3(b), ont été visitées principalement début juillet puis le nombre de visites décroit, pour reprendre plus fortement début sep-



(a) (b)

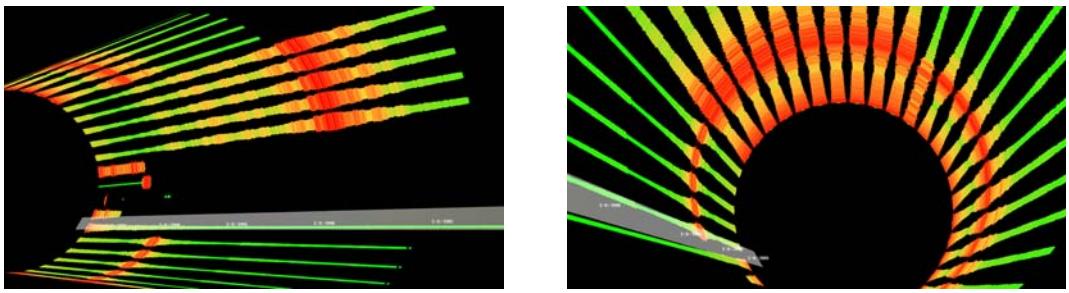
FIG. 3 – *Site Internet 1 - Vue globale en (a) et zoom en (b).*

tembre, cet intêret pour ces pages correspond notamment à la mise en ligne des pages, à la période d'inscription des étudiants, des vacances puis de la rentrée des étudiants.

– La zone représentée par l'annotation *3* montre que les pages "Revue de presse" et "Actualités" contenant des articles de presse ainsi que des galeries de photos et vidéos des diverses activités, informations et événements, sont visitées régulièrement et pas seulement lors de l'ajout de ces pages sur le site, ce qui montre l'intérêt des personnes pour les événements actuels mais aussi plus anciens. On peut noter des visites ponctuelles au mois de juillet jusqu'à mi-août, correspondant aux vacances, puis un regain d'intérêt à partir de fin août, correspondant à la rentrée des étudiants ainsi qu'à l'arrivée de nouveaux articles sur le site.
– La zone *4* représente les visites sur la page d'accueil. On peut observer qu'il y a beaucoup de visites de mi-juin à fin juillet avec plus de 1900 visites par jours. En période de vacances les visites se font plus rares (ce qui montre que ce ne sont pas les robots des moteurs de recherche qui font le plus de visites) puis reprennent début septembre.

D'autres informations plus localisées sont lisibles, telles que des galeries beaucoup moins visitées que d'autres, les pages disparues ou non accessibles. Une telle visualisation appliquée à des données logs de serveurs Web permet d'avoir une vision plus globale des données qu'avec les outils d'analyse classiques à deux dimensions. Elle permet d'évaluer le niveau de visites des pages Web mais aussi de voir les zones d'un site peu ou pas accessibles, les moments de l'année de la journée ou de la semaine importants. Ces informations sont donc essentielles pour un Webmaster.

Ici nous n'avons traité que les impacts sur les pages mais un rapport de logs fournit d'autres informations qui peuvent êtres visualisées, telles que le nombre de sessions utilisateur, les navigateurs...) offrant ainsi un grand nombre de possibilités. Enfin, nous avons également visualisé les données en utilisant une échelle heure par heure, ce qui représente près de 1500000 valeurs (un chiffre très supérieur aux valeurs indiquées dans l'état de l'art).



(a) (b)

FIG. 4 – *Indices boursier*

4.3 Indices boursiers

Nous examinons ici l'évolution journalière de sept indices boursiers (CAC40, S&P, NASDAQ, FTSE, NIKKEY, DAX, Dow Jones Industrial), suivant plusieurs critères tels que les valeurs à l'ouverture, à la fermeture, maximum, minimum ainsi que le volume de transactions enregistrées. Les données sont étalées sur une durée de plus de 80 années, c'est à dire depuis la création du plus vieil indice, donnant ainsi une visualisation soit globale par année, soit plus sélective en visualisant des moments précis de l'historique. Les visualisations de la figure 4 montrent par exemple les informations suivantes :

- Dans les années 1985 à 1988, on note une forte hausse de l'indice *NIKKEY*, alors que dans les années 2000, tous les autres indices augmentent sauf *NIKKEY* qui reste très faible,
- Depuis 2005, la majorité des indices augmente progressivement sauf *NIKKEY* et *NASDAQ*,
- Plus localement, on peut voir que les différentes valeurs d'un indice évoluent de manière identique tout au long de l'année.

4.4 Exploration d'électro-encéphalogrammes

L'équipe Vieillissement et mémoire du laboratoire Langage Mémoire et Développement Cognitif (UMR 6215, Université de Poitiers et Université de Tours) nous a fourni des données issues d'électro-encéphalogramme illustrant par des courbes la réaction de 64 zones du cerveau à des stimuli visuels (Fay et al. (2005)). Les données représentent donc l'évolution des 64 électrodes dans le temps, ici sur une durée de deux secondes par pas de 2 ms. Notre objectif est d'évaluer la performance de l'algorithme de regroupement décrit dans la section 3.2, ainsi que l'intérêt de DataTube2 pour ce type de données.

Sur la figure **??**(a) nous pouvons voir directement l'évolution de chaque électrode par rapport aux autres, les comparer grâce au regroupement effectué et évaluer les zones du cerveau actives lors de la réception d'un stimulus (ici l'apparition d'une image devant les yeux), ainsi que les répercutions sur les autres électrodes. La visualisation DataTube2 permet de voir directement les relations entre les électrodes et les signaux du cerveau, contrairement aux courbes

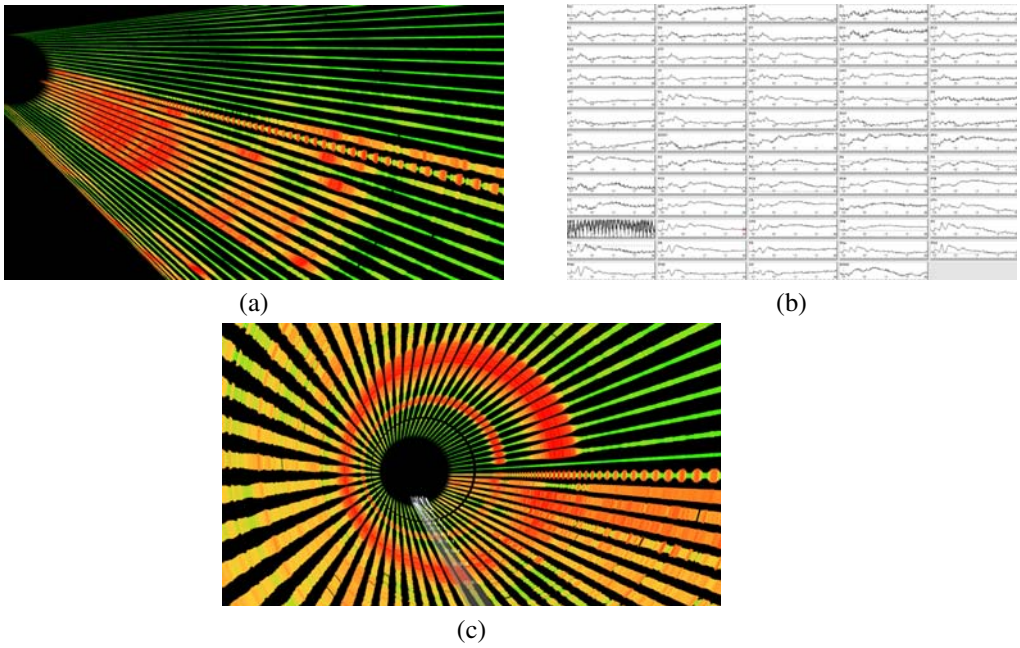


FIG. 5 – *Electro-encéphalogrammes lors d'un stimulus visuel en (a) et (b), effet d'un clignement de l'oeil en (c)*

couramment utilisées, visibles sur la figure **??**(b). La figure 5(c) montre l'état des électrodes lorsque le patient cligne de l'oeil et donc les zones actives du cerveau.

5 Conclusion

Nous avons étendu dans cet article une méthode de visualisation dans le but de traiter des domaines réels. Notre contribution en terme de rendu par la couleur et par la vision en réalité virtuelle, les différentes interactions, ainsi que la réorganisation des attributs par similarité apporte un degré de lisibilité supplémentaire à la visualisation. Cela permet à la fois une analyse minutieuse et globale des données sans modifications et de manière très intuitive. Nous traitons des ensembles de données plus conséquents que les approches visuelles concurrentes.

Parmi les perspectives, nous souhaitons repousser les limites du nombre de données visualisées, par amélioration des méthodes d'affichage. Nous testons actuellement d'autres algorithmes de réorganisation pour évaluer les résultats et apporter différentes solutions à l'utilisateur. Nous pourrions définir aussi d'autres mesures de similarité, par exemple pour des données symboliques avec des distances entre séquences. Egalement, l'affichage des couleurs et plus généralement le codage attributs/facettes peut être amélioré en proposant à l'utilisant des échelles non linéaires (pour la coloration par exemple). Ainsi, des méthodes de distorsion

pourraient être appliquées sur les couleurs afin de faire apparaître des phénomènes qui sont invisibles à l'échelle globale.

Ces premiers résultats, soumis aux experts des différents domaines (logs Web et EEG), ont montrés que la visualisation permettait une analyse rapide des données pour une personne non initiée. Notre but est maintenant de la développer pour la rendre encore plus intuitive et de la soumettre aux utilisateurs pour traiter des données à plus grande échelle (logs web sur une ou plusieurs années et EEG concernant des études psychologiques approfondies).

Enfin il pourra être intéressant d'introduire la notion de hiérarchie dans les données ainsi que dans la méthode de réorganisation. En effet, pour des données issues d'électroencéphalogrammes avec 64 électrodes par patient nous pourrions réorganiser la visualisation pour que les électrodes soient replacées sur la visualisation par patient et non plus indépendamment les unes des autres. Ou encore, la notion de hiérarchie permettrait de garder une structure arborescente de l'organisation du site web dans le cas d'étude de logs.

Remerciements

Nous tenons à remercier Simon Assani et Romain Lucas pour leur aide dans l'implémentation de DataTube2. Nous remercions également Lucie Angel, Badiaa Bouazzaoui et Michel Isingrini pour nous avoir fourni les données d'EEG.

Références

Ankerst, M. (2001a). *Visual Data Mining*. Dissertation.de.

Ankerst, M. (2001b). Visual data mining with pixel-oriented visualization techniques. *Proceedings of the ACM SIGKDD Workshop on Visual Data Mining*.

Ankerst, M., D. Jones, A. Kao, et C. Wang (1996a). Datajewel : Tightly integrating visualization with temporal data mining. *ICDM Workshop on Visual Data Mining*.

Ankerst, M., D. Keim, et H. Kriegel (1996b). Circle segments : A technique for visually exploring large multidimensional data sets. *Proc. Visualization 96*.

Antunes, C. et A. Oliveira (2001). Temporal data mining: An overview. *KDD Workshop on Temporal Data Mining*.

Azzag, H., F. Picarougne, C. Guinot, et G. Venturini (2005). Vrminer: A tool for multimedia database mining with virtual reality. *Processing and Managing Complex Data for Decision Support.*.

Carlis, J. et J. Konstan (1998). Interactive visualization of serial periodic data. *Proceedings of the 11th annual ACM symposium on User interface software and technology*, 29–38.

Chi, E., J. Pitkom, J. Ma&inlay, P. Pirolli, R. Gossweiler, et S. Card (1998). Visualizing the evolution of web ecologies. *In Proceedings of the Conference on Human Factors in Computing Systems CHI98*.

Climer, S. et W. Zhang (2006). Rearrangement Clustering: Pitfalls, Remedies, and Applications. *The Journal of Machine Learning Research 7*, 919–943.

Daassi, C., M. Dumas, M. Fauvet, L. Nigay, et P. Scholl (2000). Visual exploration of temporal object databases. *proc. of BDA00 Conference*, 24–27.

Fay, S., M. Isingrini, R. Ragot, et V. Pouthas (2005). The effect of encoding manipulation on word-stem cued recall: an event-related potential study. *Cognitive brain research 24*(3), 615–626.

Francis, B. et J. Pritchard (2003). Visualisation of historical events using lexis pencils. *Case Studies of Visualization in the Social Sciences 30*.

Hackstadt, S. T. et A. D. Malony (1994). Visualizing parallel program and performance data with ibm visualisation data explorer. Master's thesis.

Hébrail, G. et A. Debregeas (1998). Interactive interpretation of kohonen maps applied to curves. *Proceedings of the 4 thInternational Conference on Knowledge Discovery and Data Mining. AAAI press, Menlo Park*, 179–183.

Kandogan, E. (2000). Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. *IEEE Symposium on Information Visualization 2000*.

Keim, D., M. Ankerst, et H. Kriegel (1995). Recursive pattern: A technique for visualizing very large amounts of data. *Proceedings of the 6th conference on Visualization'95*.

Minard, C. (1861). Carte figurative des pertes successives en hommes de l'armée française dans la campagne de russie 1812-1813.

Muller, W. et H. Schumann (2003). Visualization methods for time-dependent data-an overview. *Simulation Conference, 2003. Proceedings of the 2003 Winter 1*, 737–745.

van Wijk, J. et E. van Selow (1999). Cluster and calendar based visualization of time series data. *Proceedings of IEEE Symposium on Information Visualization*, 4–9.

Wattenberg, M. (2002). Arc diagrams: visualizing structure in strings. *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, 110–116.

Weber, M., M. Alexa, et W. Muller (2001). Visualizing time-series on spirals. *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on*, 7–13.

Summary

We deal in this paper with visual mining of temporal data, where the data are represented by n time-dependent attributes (or series). We describe the state of the art in temporal data visualization, and we concentrate on a specific visualization (DataTube) which has received yet little attention. DataTube uses a tubular shape to represent the data. The axis of the tube represents the time. We perform several extensions to this visualization: we define several visualizations (color, shapes, etc) and we add a temporal axis. We introduce several interactions with the possibility to select attributes and time steps. We add a clustering algorithm in order to cluster together the attributes with similar behavior. We integrate this visualization in our data mining virtual reality platform VRMiner (with stereoscopic display and interactive navigation). We apply this visualization to several real-world data sets and we show that it can deal with 1,5 million values.

Propriétés d'un modèle conceptuel multidimensionnel pour les données complexes

Nouria Harbi, **Omar Boussaid, Fadila Bentayeb**

Équipe BDD, Laboratoire ERIC, Université Lumière – Lyon 2
5 avenue Pierre Mendès-France
69676 Bron Cedex
{nouria.harbi, omar.boussaid,fadila.bentayeb}@univ-lyon2.fr

Résumé.
Nous avons constaté que dans la plus part des cas, la modélisation des applications multidimensionnelles est liée à l'implantation [Torlone 2003]. Cette démarche montre l'importance accordée aux caractéristiques de l'information manipulée afin d'avoir des temps de réponse acceptables par les décideurs
Cet article présente une réflexion sur une approche d'aide à la conception et à la validation d'un modèle conceptuel multidimensionnel sans se préoccuper des contraintes physiques liées à l'implémentation. Il s'agit de voir s'il est possible d'utiliser un socle minimum de propriétés comme critères de validation pour définir les différents éléments du modèle : faits, dimensions, hiérarchies.
Les propriétés sont liées à la structuration des dimensions, à la structuration des faits et des mesures, à la cohérence de l'interrogation des données, au traitement symétrique des faits et des dimensions au cours des manipulations des données et à la dynamique du système d'information décisionnel. Nous allons nous intéresser particulièrement à la modélisation multidimensionnelle des données complexes.

1. Introduction

A l'heure actuelle, il est largement accepté que la modélisation de référence pour les systèmes d'information décisionnelle soit la modélisation multidimensionnelle [Inmo 1996 ; Jarke et al 2001]. Les systèmes d'information décisionnels ont pour objectif d'assurer le support du processus de prise de décision. Autrement dit, ces derniers doivent permettre des analyses complexes des données. Cependant, les caractéristiques des tâches d'analyse de données suivant un grand nombre d'axes ne peuvent pas toutes être représentées avec un modèle de système d'information classique telles que les agrégations, les calculs complexes, les consolidations...
De ce fait, le modèle entité-association est reconnu tant par la communauté des chercheurs que d'industriels comme un modèle orienté plutôt vers la gestion des données et par conséquent inadéquat pour les applications d'aide à la décision [Kimball 1996 ; Blaschka et al 1999].
Pour pallier aux inadéquations de ce modèle et répondre à l'objectif d'analyse des données, la modélisation multidimensionnelle a été proposée. Historiquement, la préparation des données à l'analyse avait pour préalable d'extraire des données (des vues) à partir de sources de données qui étaient en général relationnelles. Les modèles multidimensionnels proposés correspondaient à des modèles logiques basés sur des concepts relationnels (tables, clefs

principales, clefs étrangères, associations…). Qui de plus, au vue de l'importance de la volumétrie des données, la dénormalisation était tolérée afin de satisfaire des contraintes de performances. Dans la plus part des cas, la modélisation des applications multidimensionnelles est liée à l'implantation [Torlone 2003]. Cette démarche montre l'importance accordée aux caractéristiques de l'information manipulée afin d'avoir des temps de réponse acceptables par les décideurs. Ainsi le modèle conceptuel était confondu avec le modèle logique. C'est le cas du schéma en étoile par exemple, qui depuis toujours est perçu comme un modèle conceptuel pour exprimer les besoins d'analyse et est formulé en même temps dans des termes de modèle logique. Cette confusion biaise le discours de l'analyste et du coup met en évidence l'absence d'un véritable modèle conceptuel multidimensionnel reconnu par tous.

L'avènement du décisionnel date maintenant de deux décennies presque et depuis de nombreux travaux, sur la modélisation conceptuelle multidimensionnelle, existent dans la littérature. Toutefois, aucun consensus n'a pu se dégager sur un modèle conceptuel multidimensionnel standard. Cependant, différentes propriétés concernant ce modèle se dégagent à partir des différentes propositions déjà avancées dans ce domaine. Pour définir un modèle multidimensionnel qui représente uniquement les concepts du monde réel indépendamment de toute implantation physique et de tout formalisme logique, différentes listes de propriétés ont été proposées dans [Vassiliadis and Sellis 1999 ; Blaschka et al 1999 ; Abelló et al 2001 ; Rafanelli 2003 ; Lujan-Mora 2005 ; Annoni 2007]. L'idée de ce papier est de réunir l'ensemble de ces propriétés que nous avons trouvées dans la littérature pour servir comme base de contraintes pour que toute proposition de modèle conceptuel multidimensionnel doit satisfaire.

Nous allons présenter, d'abord dans une première partie de cet article, les concepts de base du modèle multidimensionnel. Puis dans une seconde partie, la présentation des propriétés sera illustrée via un exemple. Enfin dans une dernière partie, nous discuterons de l'application de ces propriétés à la modélisation conceptuelle des entrepôts de données complexes.

2. Modélisation multidimensionnelle

La modélisation multidimensionnelle vise à représenter les données en fonction de l'analyse prévue par les décideurs. Elle représente l'information à analyser, comme un point dans un espace à plusieurs dimensions qui sont les axes de l'analyse [Chrisment et al 2005]

Un autre constat concernant les nombreuses propositions sur la modélisation conceptuelle multidimensionnelle est qu'elles ne reposent pas sur des bases théoriques standards [Rizzi et al 2006]. Actuellement, bien qu'aucun modèle n'est reconnu comme standard ; il se dégage néanmoins trois concepts qui sont admis par la communauté des chercheurs et des industriels comme étant les bases d'une modélisation multidimensionnelle, le concept de «cube de données», mais aussi ceux de «dimension»' et de «faits». Nous définissons ces concepts comme :
- *Faits* : Un fait représente un sujet d'analyse, appelé aussi une donnée factuelle ou un centre d'intérêt sur lequel porte l'analyse. Il est exprimé en fonction d'indicateurs appelés mesures et d'axes d'observation appelés dimensions.

- *Dimension* : Une dimension représente un axe suivant lequel l'analyse des données est faite. Elle se compose de descripteurs des faits et peut être organisées sous forme d'hiérarchies.
- *Mesure* : Elle représente un indicateur de performance de l'activité à analyser. C'est la propriété qui caractérise un fait. Généralement c'est une donnée numérique, additive permettant des opérations d'agrégation.
- *Cube de données* : Un cube de données représente la structure dont les cellules contiennent des données mesurées (*mesures*) et dont les arêtes (*dimensions*) contiennent les axes d'analyse naturels des données [Kimball et al 1998]

Les premiers modèles proposés sont basés sur des structures (*Cube*) ne représentant pas tous les concepts exprimés par les utilisateurs mais répondant aux caractéristiques de l'implantation.

Pour définir un modèle multidimensionnel qui représente les concepts du monde réel indépendamment de tout aspect physique, d'autres concepts viennent s'ajouter pour compléter l'environnement de cette modélisation.

Les applications d'analyse multidimensionnelle requièrent des concepts qui se rapprochent de la vision des données par les décideurs et de la sémantique du décisionnel. Les concepts introduits sont : les hiérarchies, les différents types de schémas en étoile, en flocon de neige et en constellation [Kimball 1996], les paramètres, les attributs faibles, [Golfarelli et al 1998].

- *Hiérarchie* : Elle définit plusieurs paliers d'observations des faits pour une dimension donnée. Cette dernière contient des paramètres et/ou des attributs faibles (voir plus loin). Une hiérarchie est composée de sous ensemble de paramètres organisés en plusieurs niveaux représentant des granularités différentes.

Exemple : Les paramètres *Jour*, *Mois Semestre* et *Année* sont organisés en hiérarchie. C'est à dire un ensemble de paliers d'observation représentant des niveaux de granularité différents.

- *Paramètre* : C'est un membre (propriété d'une dimension) qui représente également un descripteur de faits. Celui-ci a la particularité de représenter également un descripteur pouvant définir un niveau de granularité de la hiérarchie.

Exemple : Les paramètres *Ville*, *Département*, *Pays* de la dimension «*LIEU-MAGASIN*».

- *Attribut faible* : est un membre d'une dimension ou d'une hiérarchie qui ne peut pas être transformé en niveau hiérarchique.

Exemple : Le libellé du département (*Lib Dép*) décrit le paramètre *Département.*

- *Schéma en étoile* : est une représentation multidimensionnelle des données proposée par [Kimball 1996] qui représente les faits au centre et des dimensions qui rayonnent autour des faits. Cette représentation est un standard reconnu.
- *Schéma en flocon de neige* : est un schéma en étoile dont certaines (ou toutes) dimensions sont organisées en hiérarchie.

Exemple : les dimensions «*PERIODE*», «*LIEU-MAGASIN*» organisées en hiérarchie permettent une représentation en flocon de neige (*cf* FIG 1)

- *Constellation (schéma en constellation)* : est un regroupement de schémas en étoile qui partagent des dimensions. Il met en avant la corrélation entre les faits et il évite de définir plusieurs fois la même dimension au sein d'une même organisation.

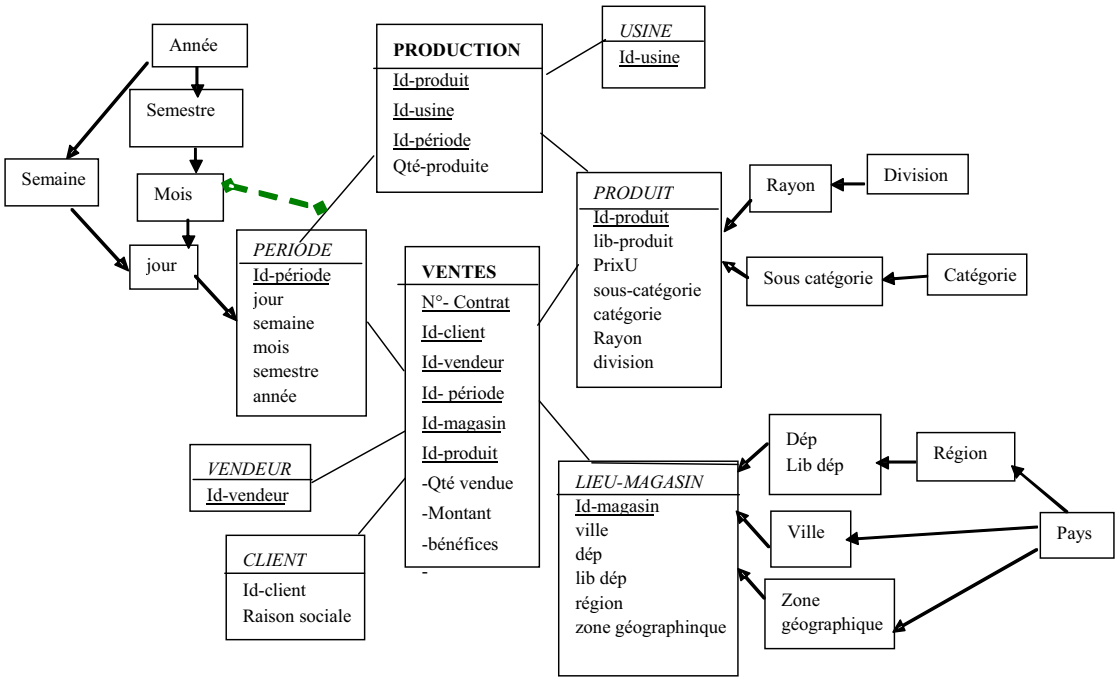


FIG 1 – *Schéma en constellation*

L'ensemble de ces concepts qui sont spécifiques aux analyses multidimensionnelles ont des propriétés liées à :

- la structuration des dimensions afin d'exprimer les liens qui existent entre les paramètres des dimensions ;
- la structuration des faits et des mesures, car les faits peuvent être liés. Un fait peut être composé de plusieurs mesures et une mesure peut dériver d'une ou plusieurs autres mesures ;
- la cohérence de l'interrogation des données car une requête peut être construite à partir du résultat d'une précédente avec des opérations d'augmentation ou de réduction du niveau de détail des données suivant la hiérarchie ;
- le traitement symétrique des faits et des dimensions au cours des manipulations des données. Cette propriété est liée aux algèbres et aux langages de définition et de manipulation des données.

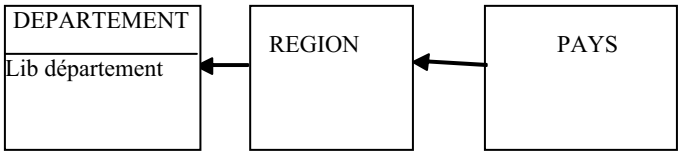
D'autre part, il existe d'autres propriétés qui sont liées à la dynamique du système d'information décisionnel [Annoni 2007]. L'ensemble de ces propriétés constituent un socle sur lequel doit reposer tout modèle conceptuel multidimensionnel.

3. Propriétés d'un modèle conceptuel multidimensionnel

Propriétés liées aux structures des dimensions

Propriété 1 : Les hiérarchies simples d'une dimension : elles explicitent le chemin entre les niveaux d'une dimension.

Exemple :

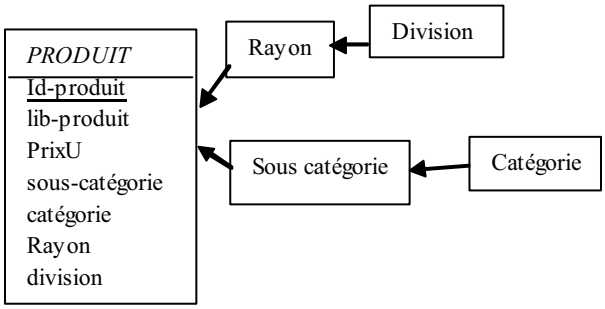


C'est une hiérarchie simple à trois niveaux de la dimension «*LIEU-MAGASIN*»
Cette hiérarchie est appelée également hiérarchie stricte si elle présente une relation 1-N entre ces niveaux : un département appartient à une seule région et une région appartient à un seul pays. D'autre part, un pays a plusieurs régions ; une région a plusieurs départements.

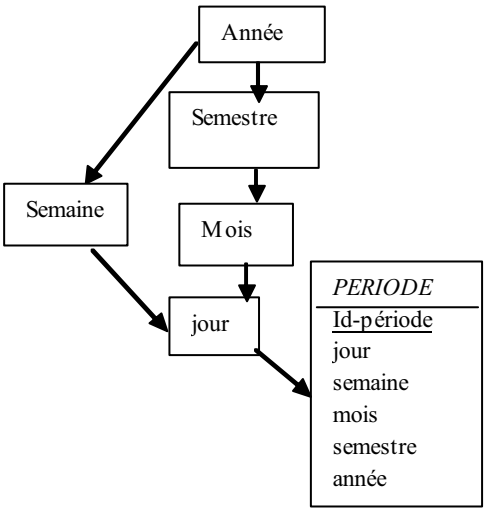
Propriété 2 : Les hiérarchies multiples (ou parallèle) et alternatives : elles partagent plusieurs niveaux d'une dimension.
Si une dimension a au moins deux hiérarchies simples dont les niveaux fins ne sont pas les mêmes, elle possède une hiérarchie multiple (parallèle). Si les niveaux fins sont les mêmes, ces hiérarchies sont dites alternatives

Exemple *:*

a) Hiérarchie multiple ou Parallèle



b) Hiérarchie alternative



Propriété 3 : Les hiérarchies non strictes : elles sont définies quand une instance d'un niveau fin peut appartenir à plusieurs niveaux supérieurs associés.

Exemples : 1 jour (quantième) peut appartenir à plusieurs mois.

Il s'agit d'une dimension qui partage entre plusieurs niveaux les mêmes membres. C'est une relation N-N entre membres.

Propriété 4 : Les éléments terminaux multiples : ils indiquent qu'une dimension peut être reliée aux faits par plus d'un de ces paramètres (éléments).
Ce concept est important car tous les faits d'un schéma en constellation ne s'expriment pas en fonction du même niveau de granularité d'une dimension.

Exemple : le fait «*VENTES*» est au niveau granularité *Jour* alors que le fait «*PRODUCTION»* est au niveau granularité *Mois*. Ce concept est représenté graphiquement par : ◆– – –◆

Propriété 5 : Les rôles d'une dimension : ils permettent de représenter qu'un fait peut s'exprimer plusieurs fois en fonction d'une même dimension.
Ces rôles sont pertinents car dans de nombreux projets une dimension intervient plus d'une fois. La conception qui en résulte est la duplication de la dimension selon les différents rôles. C'est un abus de conception car c'est le même concept qui participe à la relation avec des rôles différents.

Exemples : - la dimension '*USINE'* intervient deux fois : une fois en tant que «*USINE DE PRODUCTION*» et une fois en tant que «*USINE, VENTE USIN»*.

Un aéroport intervient une fois en tant que «*AEROPORT DE DEPART*» et une seconde fois en tant que «*AEROPORT D'ARRIVEE*».

Propriété 6 : Une dimension dégénérée : elle désigne un attribut d'un fait qui permet de l'identifier de manière unique et qui n'est pas une donnée qui caractérise le fait. Autrement dit l'attribut n'est pas une mesure. Il agit comme une clef seulement.

Exemple : Dans le fait «*VENTES*», le *N° de contrat* identifie de manière unique le fait «*VENTES*». Alors que le fait peut aussi être identifiable de manière unique par les clefs étrangères des dimensions en fonction des quelles s'exprime une vente : «*PRODUIT*», «*VENDEUR*», «*LIEU-MAGASIN*», «*CLIENT*», «*PERIODE*».

Les propriétés liées aux structures des faits et aux mesures :

Propriété 7 : Les mesures multiples : elles caractérisent un fait composé de plusieurs dimensions.

Exemple : le fait «*VENTES*» est composé des dimensions : «*PRODUIT*», «*VENDEUR*», «*LIEU-MAGASIN*», «*CLIENT*», «*PERIODE*».

Propriété 8 : Les faits multiples : ils constituent le schéma en constellation. Ils permettent de représenter plusieurs domaines d'activités d'un système décisionnel complexe.

Exemple : Dans FIG. 2 les faits : «*VENTES*» et «*PRODUCTION*»

Propriété 9 : Les mesures dérivées : elles définissent les mesures obtenues par calcul à partir d'une ou plusieurs mesures d'un fait et d'autres données. Ce calcul peut être arithmétique une fonction analytique ou une fonction d'agrégation

Exemple : Dans le domaine «*VENTES*»', la mesure *Chiffre d'affaire* peut être déduite par un calcul arithmétique simple à partir de la mesure *Qté vendue* et une données source qui est le prix unitaire *PrixU.*

Propriété 10 : Les liens entre mesures : Ils explicite les règles de calcul des mesures dérivées. C'est une des tâches des décideurs pour préciser les causes et les tendances. C'est aussi un moyen pour contrôler la fiabilité des données. Le concepteur déroule tout le calcul à l'origine d'une mesure. La connaissance des liens entre les mesures permet de procéder à cette validation plus efficace et par la même occasion connaître les mesures concernées par la modification d'une mesure donnée.

Exemple :
- Evaluation des commissions des vendeurs,
- Calcul des chiffres d'affaires par produit par client, par vendeur…,

Propriété 11 : Un fait dégénéré : il est une mesure telle qu'à une valeur de celle-ci sont associées plusieurs valeurs d'une dimension pour un même rôle

Exemple : Dans le cas où une vente est réalisée par plusieurs vendeurs, la mesure *Commission* du fait «*VENTES*» est transformée en fait dégénéré associé au lien entre le fait «*VENTES*» et la dimension «*VENDEUR*».

La cohérence de l'interrogation

La cohérence de l'interrogation est liée à la fiabilité des données. En effet, une requête construite via des opérations de manipulation sur les données suivant une hiérarchie permet de réduire ou d'augmenter le niveau de détail des données. Elle se caractérise par la pertinence des agrégations.

Propriété 12 : La pertinence des agrégations : elle définit les fonctions d'agrégation valides pour le passage d'un niveau de détails *N* à un niveau moins détaillé *M* pour une hiérarchie d'une dimension. Cette propriété indique les consolidations des mesures qui ont un sens pour les décideurs [Husemann et al 2000].

Exemple : La fonction d'agrégation généralement utilisée par défaut est la somme. Si dans un cas la fonction somme n'a pas de sens pour une mesure, la fiabilité des données obtenues n'es pas garantie.

Aspect dynamique

La dynamique s'évalue en fonction des traitements que le modèle permet de représenter. Par ailleurs, les traitements du système d'information décisionnel se rapportent à la dérivation des données et à la préparation des données

Propriété 13 : La dérivation des données : elle regroupe les traitements d'extraction, de chargement, de suivi et de sécurité des données (traitements de l'ETL).

Propriété 14 : La préparation des données : elle regroupe les traitements liés à la validité, l'historisation, le rafraîchissement, l'archivage, le calcul et la consolidation des données. Autrement dit ceux liés à la caractérisation des données pour la prise de décision.

Répertorier cette liste de propriétés constitue déjà un recueil prometteur pour la modélisation conceptuelle multidimensionnelle. Cette liste n'est pas exhaustive, loin s'en faut mais peut servir de base de départ. Elle est appelée à être étendue de par son application aux données complexes avec leur spécificité et surtout de par l'évolution de l'OLAP avec son élargissement à d'autres techniques d'analyse, telles la recherche d'information et la fouille de données.

4. Modélisation multidimensionnelle des données complexes

La modélisation multidimensionnelle des données complexes fait déjà l'objet de nombreux travaux de recherche et constitue un véritable challenge scientifique. Ces travaux suivent deux tendances. La première consiste à décrire les données complexes et à les représenter

sous une forme adaptée aux techniques classiques de l'analyse multidimensionnelle. La seconde consiste à proposer des nouvelles formes de structuration des données et à adapter l'analyse multidimensionnelle à ces nouvelles représentations.

Historiquement, l'analyse multidimensionnelle, dite encore l'analyse en ligne (OLAP), est une démarche d'analyse de données exprimant des faits à observer. Ces derniers sont décrits par des descripteurs regroupés dans des dimensions représentant les axes d'observations et par des mesures qui sont les indicateurs à observer. Ces mesures ont la particularité d'être numériques et ayant des propriétés d'additivité pour agréger les très nombreuses données et les résumer en une information pertinente. On utilise souvent l'expression de « données factuelles » pour qualifier ces données. Or, c'est plutôt l'aspect numérique qui les caractérise le mieux. La qualification factuelle est moins appropriée car la notion du *fait* est un élément conceptuel qui caractérise plutôt l'analyse multidimensionnelle. Ainsi les notions de faits, de dimensions, d'hiérarchies…, deviennent des éléments d'une modélisation conceptuelle multidimensionnelle quelque soit la nature des données et ce indépendamment de leurs représentations logique et physique.
L'analyse en ligne classique a une autre spécificité liée également aux données numériques. C'est la notion de données historisées qui est un des éléments de la définition des entrepôts de données avancées par Inmon [Inmon 1996]. Cette notion est utile et nécessaire aux fonctions d'agrégation utilisées sur les données numériques, telles la somme, la moyenne… C'est ainsi que l'OLAP classique, et ce malgré les nombreux succès de son utilisation au sein des entreprises et d'autres organisations, s'avère de plus en plus comme une démarche restreinte limitée par des contraintes liées aux données numériques. L'OLAP classique est donc peu appropriée à l'analyse multidimensionnelle des données complexes. Cette analyse en ligne est alors appelée évoluer [Rizzi et al 2006]. Bien entendu cette évolution doit s'affranchir de ces contraintes liées aux données numériques.

La modélisation multidimensionnelle des données complexes doit tenir compte de la spécificité de celles-ci. La préparation des données complexes à l'analyse en ligne passe par une structuration de données basée sur les concepts multidimensionnels : fait, dimension, hiérarchie… Différentes propositions existent déjà dans la littérature. Les documents par exemple, qui sont une catégorie de données complexes, ne contiennent que des données textuelles. Elles nécessitent par conséquent des fonctions d'agrégations spécifiques à la nature de ces données.
Amous et al [Amous et al 2002] proposent un modèle spécifique pour chaque type de données puis une structure générale capable de modéliser un document suivant les différents types de données qu'il contient. R. Tounier a élaboré un modèle conceptuel multidimensionnel en galaxie composé uniquement de dimensions éventuellement organisées en hiérarchies [Tournier 2007]. Dans ce modèle en galaxie, une dimension, quelle qu'elle soit, peut jouer le rôle de la dimension sujet (équivalent aux faits), les dimensions restantes servent de dimensions d'analyse. Pour effectuer des analyses OLAP sur cette structuration en galaxie, l'auteur a construit une algèbre OLAP contenant des opérations appropriées aux données textuelles et notamment une fonction d'agrégation calculant des mots clefs moyens.
D'autres approches existent également. Notamment celle de Boukraâ et Al. [Boukraâ et al 2008], où les auteurs proposent un modèle conceptuel multidimensionnel quelle que soit la nature des données complexes. Celui-ci modélise des objets complexes sous la forme de triplet.

C'est également un modèle dimensionnel, c'est-à-dire composé uniquement de dimensions. Un objet complexe est désigné comme l'objet sujet, les autres objets jouent le rôle d'objets d'analyse. Les liens entre objets sont plus variés et plus riches sémantiquement.
Ces deux exemples constituent déjà une nouvelle forme de modélisation dite dimensionnelle et qui préparent les données complexes à l'analyse en ligne.

Concernant les modèles basés sur XML, aussi bien pour les entrepôts XML que pour les entrepôts de documents XML, ceux-ci sont des modèles logiques voire physiques. En effet XML est un formalisme pouvant décrire toute sorte de modèle. Les schémas XML sont utilisés pour décrire le modèle logique d'une structure multidimensionnelle (entrepôt, magasin ou cube de données). XML peut également servir de modèle physique lorsqu'il s'agit de stocker des documents XML.

L'analyse des données complexes nécessite des techniques appropriées pouvant s'agir de recherche d'information ou de fouille de données. Par conséquent, il est évident que des algèbres OLAP soient définies pour de telles analyses. D'autre part, la modélisation conceptuelle multidimensionnelle doit non seulement tenir compte de ces techniques d'analyse mais également de pouvoir représenter les concepts du monde réel. Les propriétés énoncées dans la section 3 constituent un socle permettant à tout modèle conceptuel de données complexes de remplir de tels objectifs.

5. Conclusion et Perspectives

Dans cet article, nous avons présenté un ensemble de propriétés que doit satisfaire un modèle conceptuel multidimensionnel et amorcé une réflexion sur la modélisation multidimensionnelle et l'analyse en ligne OLAP de données complexes. Il s'agit dans un premier temps d'étudier la possibilité d'utiliser quelques propriétés de base pour aider à concevoir et à valider un modèle multidimensionnel. L'objectif étant de pouvoir proposer un modèle conceptuel éloigné des préoccupations d'implémentation. Il faudra envisager d'étendre la liste de ces propriétés.

Une première idée serait d'élaborer un modèle multidimensionnel générique prenant en compte l'ensemble des propriétés de base. Pour chaque type de données, nous pourrons instancier le modèle selon les caractéristiques souhaitées pour l'application (type d'hiérarchie, …) et créer le modèle conceptuel correspondant.

La piste qui se dégage serait l'utilisation de XML comme formalisme de modélisation à la fois conceptuelle, logique et physique de données multidimensionnelles. Cette démarche peut être utilisée dans le contexte de modélisation multidimensionnelle des données complexes en terme de validation.

6. Bibliographie

Abelló, A., Samos, J., and Saltor, F. (2001). A framework for the classification and description of multidimensional data models. In Mayr et al. [2001], pages 668–677.3540425276.

Abelló, A., Samos, J., and Saltor, F. (2002), Yam^2 (yet another multidimensional model) : An extension of uml. In Nascimento et al. [2002], pages 172–181.0769516386.

Amous I., Jedidi A., Sedes F., (2002), A contribution to multimedia document modeling and organizing. In: BELLAHSENE Z., PATEL D., ROLLAND C. Eds. Object-oriented information systems : 8th international conference, OOIS 2002, 2-5 septembre 2002, Montpellier, France. Heidelberg, Berlin, Allemagne: Springer-Verlag, 2002, pp 434-444. (Lecture notes in computer science, n° 2425)

Annoni, E. (2007). Eléments méthodologiques pour le développement des systèmes décisionnels dans un contexte de réutilisation. Thèse de doctorat, Université Paul Sabatier – Toulouse1, Toulouse, France.

Blaschka, M., Sapia, C., Hofling, G., and Dinter, B. (1999). An overview of multidimensional data models for olap. Technical report. Technical Report FR-1999-001.

Boukraâ, D., Bouss aid, O. (2008). A multidimensional Conceptual model for Complex Data, in Encyclopedia of Data Warehousing and Mining - 2nd Edition, John Wang Eds , to appear.

Chrisment, C., Pujolle, G., Ravat, F., Teste, O., and Zurfluh, G. (2005). Les entrepôts de données. In Zurfluh, G., editor, Trait´e Informatique des Techniques de l'Ingénieur - H3870, page 10. Techniques de l'Ingénieur.

Golfarelli, M., Maio, D., and Rizzi, S. (1998). The dimensional fact model : A conceptual model for data warehouses. Int. J. Cooperative Inf. Syst., 7(2-3) :215–247.

Husemann, B., Lechtenborger, J., and Vossen, G. (2000). Conceptual data warehouse modeling. In Jeusfeld et al. [2000], page 6.

Inmon, W. H. (1996). Building the data warehouse (2nd ed.). John Wiley & Sons, Inc., New York, NY, USA. 0471141615.

Jarke, M., Lenzerini, M., Vassiliou, Y., and Vassiliadis, P. (2001). Fundamentals of Data Warehouses. Springer-Verlag New York, Inc., Secaucus, NJ, USA. 3540420894.

Kimball, R. (1996). The data warehouse toolkit : practical techniques for building dimensional data warehouses. John Wiley & Sons, Inc., New York, NY, USA.

Kimball, R., Reeves, L., Thornthwaite, W., Ross, M., and Thornwaite, W. (1998). The Data Warehouse Lifecycle Toolkit : Expert Methods for Designing. John Wiley & Sons, Inc., New York, NY, USA. 0471255475.

Lujan-Mora, S. (2005). Data Warehouse Design with UML. PhD thesis, Department of Software and Computing Systems, University of Alicante, Alicante, Espagne.

Rafanelli, M., editor (2003). Multidimensional Databases : Problems and Solutions. Idea Group. 1591400538.

Rizzi, S., Abell´o, A., Lechtenb¨orger, J., and Trujillo, J. (2006). Research in data warehouse modeling and design : dead or alive ? In Song and Vassiliadis [2006], pages 3–10. 1595935304.

Torlone, R. (2003). Conceptual multidimensional models. In Multidimensional Data bases : Problems and Solutions, pages 69–90.

Tournier, R. (2007). Analyse en ligne (OLAP) de documents. Thèse de doctorat, Université Paul Sabatier – Toulouse1, Toulouse, France.

Vassiliadis, P. and Sellis, T. K. (1999). A survey of logical models for olap data bases.SIGMOD Record, 28(4) :64–69.

Post-classification d'images texturées par fusion crédibiliste

Hicham Laanaya[*,**], Arnaud Martin[*],
Ali Khenchaf[*], Driss Aboutajdine[**]

[*]ENSIETA-E^{3}I^2-EA3876, 2, rue François Verny 29806 Brest cedex 9,
laanayhi, Arnaud.Martin, Ali.Khenchaf@ensieta.fr
http://www.ensieta.fr/e3i2/
[**]GSCM-LRIT, Université Mohammed V-Agdal, Faculté des sciences de Rabat, Maroc
aboutaj@fsr.ac.ma,
http://www.fsr.ac.ma/GSCM/

Résumé. Nous présentons dans cette étude plusieurs approches de fusion de classifieurs en vue d'améliorer la classification d'images texturées, qui ne sont généralement pas employées dans ce cadre d'application. C'est ce que nous nommons l'étape de post-classification des images texturées. Trois approches de fusion de classifieurs sont présentées : l'approche par vote, un approche crédibiliste, et une fusion fondée sur un classifieur. Ces approches sont comparées pour la classification d'images sonar. Les résultats montrent l'intérêt de cette étape de post-classification particulièrement avec l'approche crédibiliste pour améliorer les résultats de classification d'images texturées.
Mots clés : Classification d'images texturées, fusion de classifieurs, fonctions de croyance.

1 Introduction

La classification des images texturées joue un rôle important dans le domaine du traitement d'images et de la reconnaissance des formes et en particulier pour la classification des images sonar (Laanaya et al., 2007; Leblond, 2006). Cette classification est fondée sur l'utilisation d'approches de classification supervisées (nécessitant une connaissance *a priori* des classes des données), et elle est réalisée à partir de paramètres de texture. L'extraction des paramètres de texture ne peut se faire au niveau pixel, car la texture est définie sur un ensemble de pixels. Ainsi l'unité de la classification dépend du nombre de pixels nécessaire pour extraire ces paramètres de texture, par exemple 8×8, 16×16 ou 32×32 pixels. La précision des classes obtenues sur l'image peut être améliorée en faisant glisser ces unités (imagettes) avec un pas de recouvrement en affectant la classe trouvée sur l'imagette uniquement au centre de l'imagette. Si cette approche permet d'obtenir une précision plus grande et ainsi des frontières entre classes plus lisses, elle ne tient pas compte de toute l'information obtenue par les classifications successives sur les zones des imagettes qui se recouvrent. En effet, nous avons sur chaque partie de l'image plusieurs classifieurs qui s'expriment et que nous pouvons donc fusionner en vue d'améliorer les résultats de classificaiton. C'est cette étape que nous nommons post-classification.

Dans un premier temps, en section 2, nous reviendrons sur le principe de l'approche de base pour la classification d'images texturées qui considère le centre des fenêtres glissantes, puis nous présentons les approches de fusion de classiifeurs que nous proposons d'employer sur les résultats de classification des fenêtres glissantes. Pour finir, nous présentons les résultats des différentes approches de fusion pour la classification des images sonar.

2 Classification d'images texturées par fenêtres glissantes

Pour caractériser la texture d'une image, on la découpe en fenêtres de taille $L \times L$ pixels avec un pas de recouvrement l ($l < L$) spécifié par l'utilisateur de façon à obtenir une classification plus précise. Sur chaque imagette de ce découpage sont calculés des vecteurs de paramètres de texture à partir des méthodes d'analyse de texture (Martin et al., 2004). Ce sont ces paramètres (en utilisant ou non des méthodes d'extraction-réduction de paramètres) qui sont utilisés par le classifieur. A partir de la classification des imagettes nous obtenons une classification de l'image initiale en des zones homogènes. Pour avoir une meilleure classification (frontière lisse) il faut utiliser un l proche de la taille des imagettes L. Notons que plus l est proche de L plus le temps de classification est grand puisque le nombre d'imagettes à classifier augmente avec l.

La classification s'effectue sur des imagettes de taille $L \times L$, mais dans l'approche la plus simple, seule la zone centrale, de taille $(L-l) \times (L-l)$ pixels, est affectée à la classe trouvée pour éviter le recouvrement des classes. Nous avons ainsi une classification de résolution $(L-l) \times (L-l)$ pixels. La figure 1 illustre cette méthode.

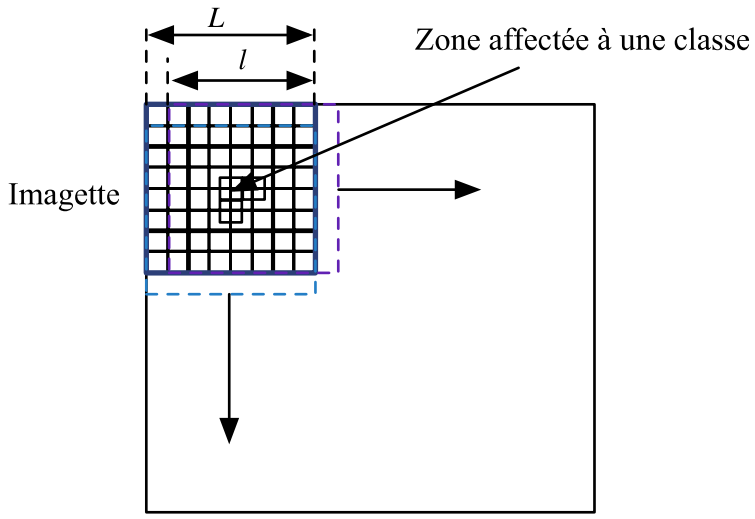


FIG. 1 – *Classification d'une image en utilisant une fenêtre glissante.*

Afin d'améliorer les résultats de classification, nous proposons de considérer des approches de fusion de classifieurs de façon à tenir compte de toutes les classifications effectuées sur les zones recouvertes.

3 Post-classification par fusion

L'approche exposée précédemment repose sur l'affectation de la classe trouvée à la zone centrale de l'imagette. Mais cette zone appartient à d'autres imagettes voisinantes qui peuvent être classifiées en d'autres classes. Notre approche se fonde sur la fusion des résultats de classification des imagettes qui contiennent « cette zone centrale ».

Soient N_c le nombre de classes, L la taille de l'imagette utilisée pour le parcours de l'image étudiée I, l le pas de recouvrement et soit r la taille effective de la zone étudiée ($r = L - l$). Cette zone z, de taille $r \times r$ pixels, appartient à un ensemble d'imagettes $I_i^z, i = 1, \ldots, N_z$ où N_z est le nombre d'imagettes contenant la zone z et dépend de la position de z dans l'image. Chaque imagette I_i^z appartient à une classe $C_q, q = 1, \ldots, N_c$.

La figure 2 illustre cette approche en dessinant quelques imagettes contenant la zone étudiée. Pour le cas de cette figure, la zone étudiée z est loin des bords de l'image, dans ce cas la zone z appartient à un nombre maximal d'imagettes au contraire du cas où cette zone est proche des bords (moins de L pixels entre la zone z et les bords de l'image I).

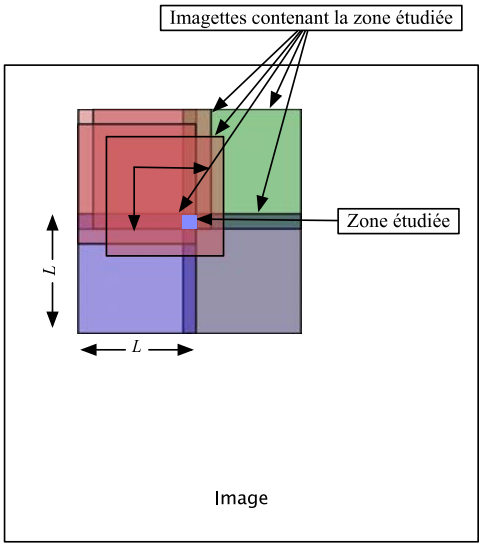


FIG. 2 – *Classification d'une image par fusion.*

Le schéma de la figure 3 donne les étapes de la fusion des informations calculées sur les imagettes $I_i^z, i = 1, \ldots, N_z$, contenant la zone z, ($I_{i_0}^z$ l'imagette centrée en z) : sur chaque imagette I_i^z on calcule l'information F_i^z, cette information peut être : la classe de l'imagette I_i^z (fusion par vote (*cf.* section 3.1)), les paramètres extraits de l'imagette I_i^z et la classe de cette imagette (fusion par classification (*cf.* section 3.2)) ou des fonctions de masses (fusion crédibiliste (*cf.* section 3.3)).

Il est donc intéressant de considérer toutes ces informations pour le calcul de la classe de la zone z. C'est donc un problème de fusion d'informations pour une décision optimale. Nous trouvons dans (Martin, 2005) une description générale de différentes approches de combinaison de résultats de classification.

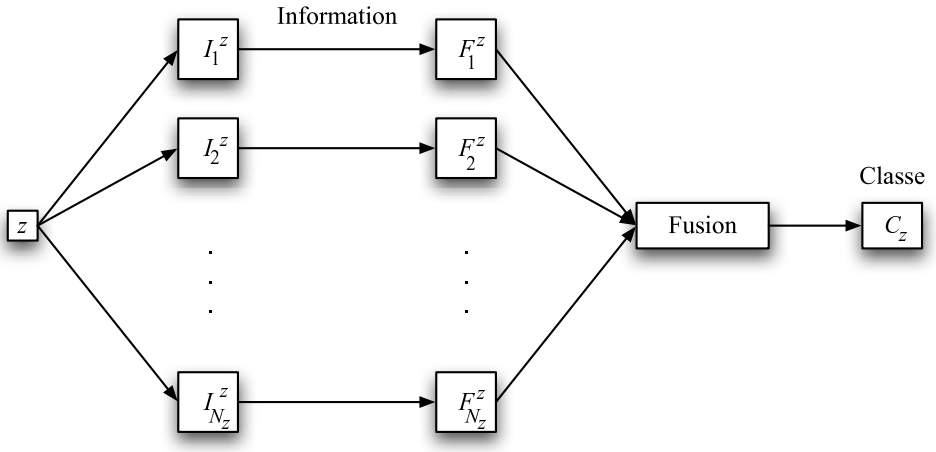


FIG. 3 – *Fusion d'information pour la classification d'une zone z.*

3.1 Fusion par vote

L'approche la plus simple de fusion est celle du vote qui repose sur la combinaison de fonctions indicatrices données par les classes des imagettes classifiées (Kuncheva, 2004). La fusion est ainsi faite suivant le principe du vote en prenant le maximum sur les nombres de fois où la zone z est assignée à une classe donnée. Nous calculons l'histogramme normalisé p_z du nombre de fois où les imagettes $I_i^z, i = 1, \ldots, N_z$, sont classifiées en une classe $C_q, q = 1, \ldots, N_c$:

$$p_z(q) = \frac{\text{card}\{i = 1, \ldots, N_z \,;\, I_i^z \in C_q\}}{N_z}. \tag{1}$$

Par cette approche, la classe C_z de la zone z est le maximum de p_z :

$$C_z = \underset{q=1,\ldots,N_c}{\text{argmax}}\; p_z(q). \tag{2}$$

Si le maximum est atteint pour plusieurs valeurs de q, on peut choisir, par exemple, la classe de la zone qui précède la zone z.

Soit $S_z = p_z(C_z)$ la valeur du maximum sur p_z. En parcourant l'image I, on forme une matrice des classes des zones z notée I_s et une matrice contenant les valeurs S_z notée I_c. Cette matrice (avec une valeur maximale de 1) indique une sorte de « certitude » sur la classification de chaque zone : une valeur proche de 1 indique que le classifieur est « sûr » de la classe affectée à cette zone.

Nous n'avons pas fait de différence, en terme de distance, entre les imagettes contenant la zone étudiée z pour la décision de la classe de cette zone. Nous pouvons envisager une pondération de ces imagettes, en affectant un poids « grand » pour les imagettes voisinantes l'imagette $I_{i_0}^z$ centrée en z et un poids « faible » pour les autres imagettes. Cette fonction poids est une fonction ψ_ρ décroissante de la distance entre la zone étudiée et les centres des imagettes contenant cette zone. Par exemple :

$$\psi_\rho(x) = e^{-\rho x}, \rho \geq 0, \tag{3}$$

ou la fonction échelon :

$$\psi_\rho(x) = \mathbb{1}_{[0,\rho]}(x). \tag{4}$$

L'utilisation d'une fonction échelon limite le nombre d'imagettes utilisées pour la décision de la classe de la zone z.

Dans ce cas, pour trouver la classe C_z de la zone z, on effectue un vote pondéré en pondérant les valeurs de l'histogramme p_z par la sommation des poids des imagettes de la même classe (*cf.* figure 4 pour les distances et les poids associés) :

$$p_w(q) = p_z(q) \sum_{i \in C_q} \psi_\rho(d_2(G_z, G_i^z)), \tag{5}$$

où G_z et G_i^z sont, respectivement, les centres des imagettes $I_{i_0}^z$ et I_i^z et d_2 la distance euclidienne dans $\mathbb{R}^2$.

$$C_z = \underset{q=1,\ldots,N_c}{\operatorname{argmax}} p_w(q). \tag{6}$$

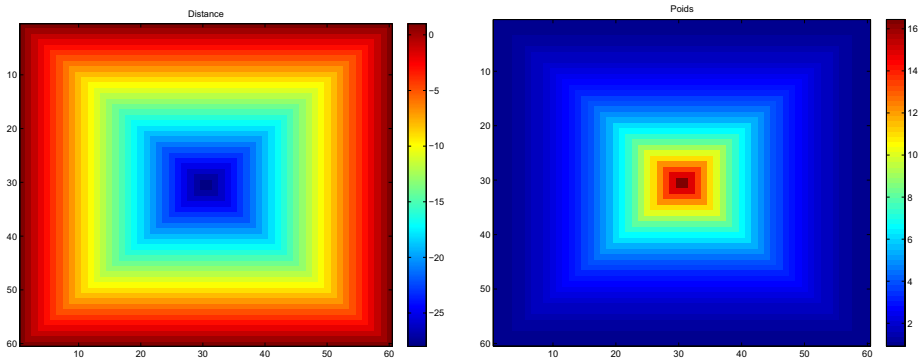


FIG. 4 – *Distances et poids en fonction de la position des imagettes contenant z ($L = 32$, $l = 28$ et $\rho = 0.1$ (cf. équation (3))).*

La matrice « certitude » I_c est définie, dans ce cas, par $I_c(z) = p_w(C_z)$. De cette façon on s'affranchit dans la plupart des cas des problèmes d'indétermination.

3.2 Fusion par classification

Supposons qu'on cherche la classe d'une zone z de taille $r \times r$. Pour cela, nous considérons les N_z imagettes $I_i^z, i = 1, \ldots, N_z$ de tailles $L \times L$ pixels contenant cette zone. Chaque imagette $I_i^z, i = 1, \ldots, N_z$ de paramètres x_i^z est classifiée en une classe y_i^z parmi les N_c classes ($x_i^z \in \mathbb{R}^p$ et $y_i^z \in \{1, \ldots, N_c\}$). Notons $X_z = \{(x_i^z, y_i^z); i = 1, \ldots, N_z; i \neq i_0\}$. On peut donc prédire la classe C_z de la zone z en utilisant l'ensemble X_z pour base d'apprentissage du classifieur utilisé.

Nous avons utilisé ici une classification supervisée pour la décision de la classe puisqu'on a une connaissance *a priori* des classes des imagettes I_i^z, une connaissance trouvée à partir de la première classification de l'image. La fusion par classification dépend, donc, de la qualité de

la première classification. Notons que le classifieur utilisé pour la fusion et celui utilisé pour une première classification peuvent être différents.

Il existe une multitude d'approches de classification, citons par exemple les k-plus proches voisins (Duda et Hart, 1973) et les machines à vecteurs de support (SVM) (Vapnik, 1998). Si nous utilisons les k-plus proches voisins il faut $k_z \leq N_z$ pour chaque zone z. Notons que la fusion par k-plus proches voisins avec $k_z = N_z$ est équivalent à l'approche de fusion par vote.

Nous considérons un classifieur à partir des machines à vecteurs de support dans la partie expérimentations.

3.3 Fusion crédibiliste

Nous proposons ici l'utilisation de la théorie des fonctions de croyance pour la fusion des résultats de classification des fenêtres glissantes.

La théorie des fonctions de croyance est fondée sur la manipulation des fonctions de masse. Les fonctions de masse sont définies sur l'ensemble de toutes les disjonctions du cadre de discernement $\mathbf{C} = \{C_1, \ldots, C_{N_c}\}$ et à valeurs dans $[0, 1]$, où C_q représente l'hypothèse "l'observation appartient à la classe q". Généralement, il est ajouté une condition de normalité, donnée par :

$$\sum_{A \in 2^{\mathbf{C}}} m(A) = 1, \tag{7}$$

où $m(.)$ représente la fonction de masse. La première difficulté est donc de définir ces fonctions de masse selon le problème. A partir de ces fonctions de masse, d'autres fonctions de croyance peuvent être définies, telles que les fonctions de crédibilité, représentant l'intensité que toutes les sources croient en un élément, et telles que les fonctions de plausibilité représentant l'intensité avec laquelle on ne doute pas en un élément.

De façon à estimer les fonctions de masse à combiner, Appriou (2002) propose deux modèles répondant à trois axiomes qui impliquent la considération de $N_c \times N_z$ fonctions de masse aux seuls éléments focaux possibles $\{C_q\}$, $\{C_q^c\}$ et $\mathbf{C}$. Un axiome garantit de plus l'équivalence avec l'approche bayésienne dans le cas où la réalité est parfaitement connue (méthode optimale dans ce cas). Ces deux modèles sont sensiblement équivalents sur nos données, nous utilisons dans cet article le modèle donné par :

$$\begin{cases} m_{iq}(C_q)(z) = \dfrac{\alpha_{iq} R_i p(F_i^z/C_q)}{1 + R_i p(F_i^z/C_q)} \\ m_{iq}(C_q^c)(z) = \dfrac{\alpha_{iq} R_i}{1 + R_i p(F_i^z/C_q)} \\ m_{iq}(\mathbf{C})(z) = 1 - \alpha_{iq} \end{cases} \tag{8}$$

où p est une probabilité, $R_i = (\max_{i,q} p(F_i^z/C_q))^{-1}$ est un facteur de normalisation, et $\alpha_{iq} \in [0, 1]$ est un coefficient d'affaiblissement permettant de tenir compte de la fiabilité de l'information fournie par l'imagette $I_i^z : F_i^z$ pour une classe C_q, que nous choisissons ici égale à 0.95. La difficulté de ce modèle est alors l'estimation des probabilités $p(F_i^z/C_q)$. Dans le cas où la donnée F_i^z de l'imagette I_i^z est la réponse d'un classifieur exprimée sous la forme de la classe (donnée symbolique), l'estimation de ces probabilités peut être faite par les matrices de confusion sur une base d'apprentissage.

La combinaison des $N_c \times N_z$ fonctions de masse que nous employons ici est la règle proposée par Yager (1987) qui permet de répartir le conflit intrinsèque (masse sur $\emptyset$) à la fusion de nombreuses fonctions de masse. La règle est définie pour deux fonctions de masse m_1 et m_2 et pour tout $A \in 2^{\mathbf{C}}$ par :

$$\begin{cases} m(A) = (m_1 \oplus m_2)(A) = \sum_{B \cap C = A} m_1(B) m_2(C), \\ m(\mathbf{C}) = (m_1 \oplus m_2)(\mathbf{C}) + m(\emptyset). \end{cases} \tag{9}$$

De nombreuses autres règles ont été proposées, un bref état de l'art ainsi que de nouvelles règles de combinaison gérant le conflit sont données par Martin et Osswald (2007).

Afin de conserver un maximum d'informations, il est préférable de rester à un niveau crédal (*i.e.* de manipuler des fonctions de masse) pendant l'étape de combinaison des informations pour prendre la décision sur les fonctions de masse issues de la combinaison. Si la décision prise par le maximum de crédibilité peut être trop pessimiste, la décision issue du maximum de plausibilité est bien souvent trop optimiste. Le maximum de la probabilité pignistique, introduite par Smets (1990), reste le compromis le plus employé. La probabilité pignistique est donnée pour tout $X \in 2^{\mathbf{C}}$, avec $X \neq \emptyset$ par :

$$\mathrm{betP}(X) = \sum_{Y \in 2^{\mathbf{C}}, Y \neq \emptyset} \frac{|X \cap Y|}{|Y|} \frac{m(Y)}{1 - m(\emptyset)}. \tag{10}$$

4 Expérimentations

La base de données est constituée de 42 images sonar fournies par le GESMA (Groupe d'Etudes Sous-Marines de l'Atlantique) qui ont été obtenues à partir d'un sonar Klein 5400 au large des côtes finistériennes. Ces images ont été labelisées à partir d'un logiciel développé spécialement en spécifiant le type du sédiment présent (sable, ride, vase, roche, cailloutis ou ombre) (voir figure 5) et le degré de certitude de l'expert (sûr, moyennement sûr ou non sûr) (Laanaya, 2007). Parmi ces sédiments, nous avons considéré trois classes distinctes, particulièrement importantes pour la navigation sous-marine et les sédimentologues. Ainsi la première classe regroupe roche et cailloutis, la deuxième classe les rides et la troisième le sable et les vases.

L'unité retenue pour l'extraction de paramètres de texture et pour la classification est l'imagette de taille 32×32 pixels (soit environ 640×640 cm).

4.1 Extraction de paramètres

Nous avons calculé sur ces imagettes six paramètres extraits à partir des matrices de cooccurrence calculés sur les imagettes (Martin et al., 2004). Les matrices de cooccurrence C_θ sont calculées en comptant les occurrences identiques de niveaux de gris entre deux pixels contigus dans une direction θ donnée. Quatre directions sont considérées : 0, 45, 90 et 135 degrés. Six paramètres d'Haralick (l'homogénéité, le contraste, l'entropie, la corrélation et l'uniformité) sont ensuite calculés et moyennés sur les quatre directions.

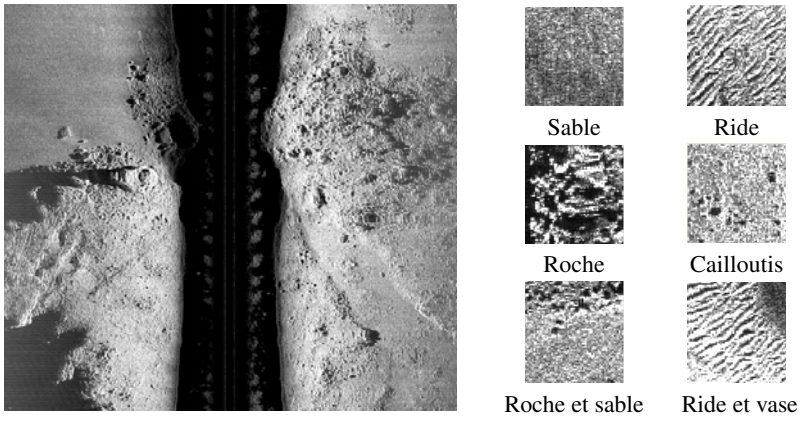


FIG. 5 – *Exemple d'image sonar (fournie par le GESMA) et d'imagettes étiquetées.*

4.2 Evaluation de la classification

L'évaluation de la classification des images sonar est une chose délicate (Martin et al., 2006). Le résultat de classification d'une image peut être évaluée visuellement en le comparant avec la vérité terrain (image de référence). Mais, pour évaluer l'algorithme de classification, nous devons considérer plusieurs configurations possibles. Généralement, les algorithmes de classification sont évalués en utilisant la matrice de confusion. Cette matrice (MC) est composée par les nombres MC_{ij} des éléments de la classe i qui sont classifiés en classe j. Nous pouvons normaliser cette matrice pour obtenir des taux qui sont faciles à interpréter :

$$\mathrm{MC}_{N_{ij}} = \frac{\mathrm{MC}_{ij}}{\sum_{k=1}^{N_c} \mathrm{MC}_{ik}} = \frac{\mathrm{MC}_{ij}}{N_i}, \tag{11}$$

où N_c est le nombre de classes considérées et N_i est le nombre des éléments de la classe i. Nous pouvons calculer à partir de cette matrice de confusion normalisée le vecteur des taux de bonne classification (TC) :

$$\mathrm{TC}_i = \mathrm{MC}_{N_{ii}}, \tag{12}$$

et le taux de classification moyen TC_m, où N représente le nombre d'image totale présenté à l'algorithme de classification :

$$\mathrm{TC}_m = \frac{\sum_{i=1}^{N_c} N_i \mathrm{TC}_i}{N}, \tag{13}$$

un vecteur de probabilités d'erreur de classification (PE) :

$$\mathrm{PE}_i = \frac{1}{2}\left(\sum_{j=1, j\neq i}^{N_c} \mathrm{MC}_{N_{ij}} + \sum_{j=1, j\neq i}^{N_c} \frac{\mathrm{MC}_{N_{ji}}}{N_c - 1} \right), \tag{14}$$

et la probabilité moyenne pondérée des PE_i :

$$\mathrm{PE}_m = \frac{\sum_{i=1}^{N_c} N_i \mathrm{PE}_i}{N}. \tag{15}$$

Nous avons proposé dans (Martin et al., 2006) une approche qui tient compte de la certitude et de l'imprécision de l'expert : si une imagette de classe i avec une certitude, associée à un poids $w \in [0,1]$, est classée en classe j alors $\mathrm{MC}_{N_{ij}}$ sera $\mathrm{MC}_{N_{ij}} + w$ et si une imagette contenant plus d'une classe est classée en une classe i, alors, pour $j = 1, \ldots, N_c$, $\mathrm{MC}_{N_{ij}}$ sera $\mathrm{MC}_{N_{ij}} + N_j/L^2$, où N_j est le nombre de pixels j de l'imagette

4.3 Résultats

Nous présentons ci-dessous les résultats obtenus en utilisant la méthode des Machines à Vecteurs de Support pour la classification (Vapnik, 1998). Nous avons utilisés le logiciel *libSVM* (Chang et Lin, 2001) pour nos tests en utilisant un noyau gaussien avec $C = 1$ et $\gamma = 1/6$ le paramètre de ce noyau. Ce même classifieur est employé de nouveau pour l'approche de fusion par classification.

Nous avons utilisés 24 images sonar pour l'apprentissage du classifieur et 18 images sonar pour le test. Nous ne considérons ici que les imagettes homogènes, de taille 32×32 pixels, pour l'apprentissage du classifieur. Nous avons ainsi 20424 imagettes avec 2353 imagettes roches et cailoutis, 2583 imagettes ride et 15488 imagette sable et vase. Le classifieur utilisé pour la fusion utilise une base de données d'apprentissage d'une taille $N_z - 1$ pour chaque zone z.

La figure 6 donne la segmentation manuelle d'une image sonar utilisée. La figure 7 montre le résultat de classification d'une image sonar par les quatre approches considérées.

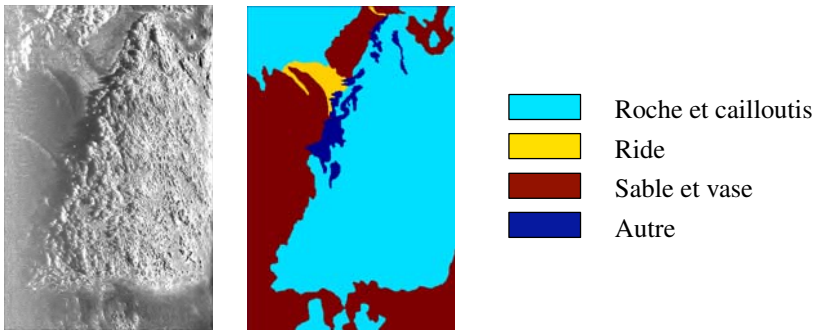


FIG. 6 – *Image sonar et une segmentation manuelle de cette image.*

Comme noté dans la section 3.1, nous pouvons calculer une matrice certitude pour la décision de la classe en utilisant la fusion par vote. La figure 8 donne la matrice certitude pour l'exemple de l'image sonar présentée sur la figure 6. Nous présentons sur la même figure l'image du "conflit" pour l'approche de fusion crédibiliste : la masse sur l'ensemble $\mathbf{C}$.

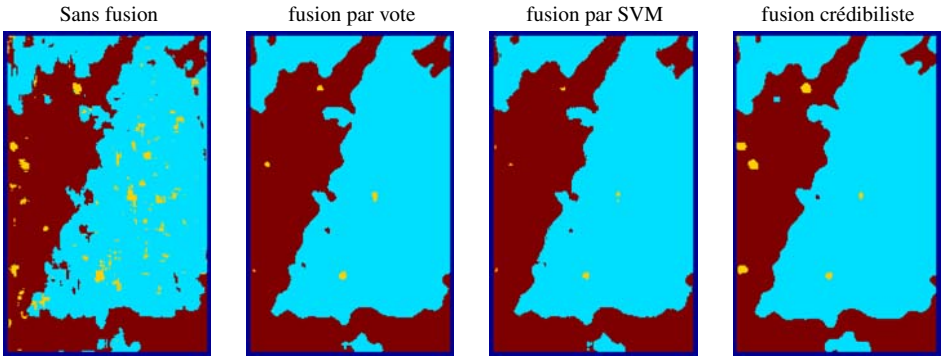


FIG. 7 – *Classification automatique d'une image sonar.*

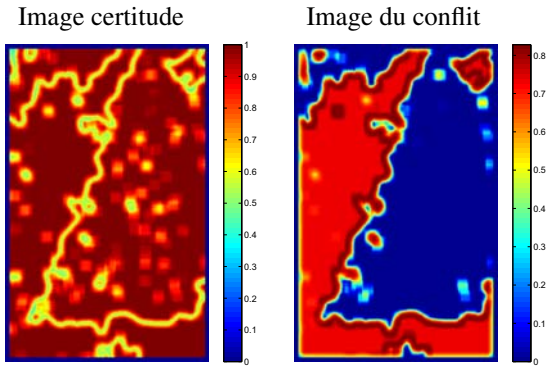


FIG. 8 – *Image certitude pour la fusion par vote et image de conflit pour la fusion crédibiliste.*

Nous avons considéré la matrice de confusion (16), calculée sur la base d'apprentissage pour l'estimation des masses des imagettes, elle est donnée par :

$$\begin{pmatrix} 1753 & 141 & 459 \\ 221 & 1353 & 1009 \\ 138 & 91 & 15259 \end{pmatrix} \tag{16}$$

Nous présentons dans le tableau 1 les résultats obtenus sans et avec fusion en utilisant les trois approches décrites précédemment. Ces résultats sont présentés sous forme de matrices de confusion normalisées en utilisant l'approche présentée dans (Martin et al., 2006). Nous adopterons l'intervalle de confiance à 95% pour représenter les taux de classification moyens et les probabilités d'erreurs moyennes. Ce tableau, (*cf.* tableau 1), donne aussi le temps CPU pour les trois approches de fusion pour la classification de l'image présentée sur la figure 6. Nous remarquons que l'approche de fusion crédibiliste est beaucoup plus lente que les approches par vote et classification.

Nous remarquons que les différentes approches par fusion donnent des taux de classification significativement meilleurs que le taux trouvé par la première approche considérée. En

	MC_N (%)	TC_m (%)	PE_m (%)	Temps CPU (s)
Avant	$\begin{pmatrix} 68.68 & 4.48 & 26.84 \\ 13.51 & 55.42 & 31.07 \\ 0.71 & 0.43 & 98.86 \end{pmatrix}$	88.11±0.08	19.26±0.09	0
Vote	$\begin{pmatrix} 70.99 & 1.42 & 27.60 \\ 8.93 & 57.38 & 33.69 \\ 0.49 & 0.28 & 99.23 \end{pmatrix}$	89.03±0.08	18.10±0.09	11.67
SVM	$\begin{pmatrix} 70.92 & 1.42 & 27.66 \\ 8.82 & 57.39 & 33.79 \\ 0.49 & 0.27 & 99.24 \end{pmatrix}$	89.03±0.09	18.11±0.09	15.37
Crédibiliste	$\begin{pmatrix} 74.14 & 1.55 & 24.31 \\ 11.92 & 58.91 & 29.17 \\ 1.04 & 0.57 & 98.39 \end{pmatrix}$	89.22±0.09	17.14±0.09	2599.40

TAB. 1 – *Résultats de classification avec et sans fusion.*

effet, en termes de taux de classification, l'approche crédibiliste donne un taux significativement meilleur avec un taux de 89.22±0.09%. Les deux approches par vote et par classification donnent des taux de classification similaires (89.03±0.09%). En terme de probabilités d'erreur, Les approches par vote donnent les faibles valeurs avec 17.14±0.09% pour l'approche crédibiliste, 18.10±0.09% pour l'approche par vote et 18.11±0.09% pour l'approche de fusion par classification. Le taux de classification trouvé en considérant le centre des fenêtres glissantes est de 88.11±0.08% avec une probabilité d'erreur moyenne de 19.26±0.09%. En terme de taux de classification par classe, nous remarquons que les trois approches ont pu améliorer les différents taux pour les trois classes, sauf pour le cas de fusion crédibiliste pour la classification de la troisième classe (sable et vase).

5 Conclusion

Nous avons étudié dans cet article des approches originales de fusion de classifieurs pour améliorer les résultats de classification d'images texturées, étape nous nous appelons : post-classification. Ainsi la fusion des résultats de classification des fenêtres glissantes, considère non pas une seule imagette (l'imagette centrée en la zone étudiée), mais toutes les imagettes contenant la zone à classifier. Nous avons étudié une approche fondée sur le vote, une approche employant un classifieur, et une approche crédibiliste fournissant une amélioration significative des résultats pour la classifications d'images sonar. L'approche crédibiliste est la plus coûteuse en temps de calcul mais permet d'avoir les meilleurs performances dans le cadre de cette application.

Références

Appriou, A. (2002). *Décision et Reconnaissance des formes en signal*. Hermes Science Publication.

Chang, C. C. et C. J. Lin (2001). Libsvm : a library for support vector machines. *Software available at http ://www.csie.ntu.edu.tw/∼cjlin/libsvm/*.

Duda, R. O. et P. E. Hart (1973). *Pattern Classification and Scene Analysis*. John Wesley and Sons.

Kuncheva, L. I. (2004). *Combining Pattern Classifiers : Methods and Algorithms*. Wiley-Interscience.

Laanaya, H. (2007). Classification en environnement incertain : application à la caractérisation de sédiment marin. *Thèse de l'Université de Bretagne Occidentale UFR Science de la Matière, de l'Information et de la Santé*.

Laanaya, H., A. Martin, A. Khenchaf, et D. Aboutajdine (2007). Régression floue et crédibiliste par SVM pour la classification des images sonar. *Extraction et Gestion des Connaissances (EGC),24-26 January, Namur, Belgique*, 21–32.

Leblond, I. (2006). Recalage à long terme d'images sonar par mise en correspondance de cartes de classification automatique des fonds. *Thèse de l'Université de Bretagne Occidentale UFR Science de la Matière, de l'Information et de la Santé*.

Martin, A. (2005). Fusion de classifieurs pour la classification d'images sonar. *RNTI Extraction des connaissances : Etat et perspectives*, 259–268.

Martin, A., H. Laanaya, et A. Arnold-Bos (2006). Evaluation for uncertain image classification and segmentation. *Pattern Recognition 39*.

Martin, A. et C. Osswald (2007). Toward a combination rule to deal with partial conflict and specificity in belief functions theory. In *International Conference on Information Fusion*, Québec, Canada.

Martin, A., G. Sevellec, et I. Leblond (2004). Characteristics vs decision fusion for sea-bottom characterization. *Colloque Caractérisation in-situ des fonds marins, 21-22 Octobre, Brest, France*.

Smets, P. (1990). Constructing the pignistic probability function in a context of uncertainty. *Uncertainty in Artificial Intelligence 5*, 29–39.

Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wesley and Sons.

Yager, R. (1987). On the Dempster-Shafer Framework and New Combination Rules. *Informations Sciences 41*, 93–137.

Summary

In this paper, we present various approaches for combining classifiers to improve classification of textured image, which are not generally used in this application framework. This is what we call post-classification step of textured images. Three approaches to combine classifiers are presented: the majority vote approach, belief approach, and classification-based approach. These approaches are compared for sonar image classification. The results show the interest of this post-classification step, particularly with the belief approach, to improve texture image classification results.
Keywords: Textured images classification, classifier fusion, belief functions.

Application du Modèle des Croyances Transférables dans le cadre d'expertises en Entomologie Médico-Légale

Gildas Morvan*, Alexandre Veremme*,**, David Mercier*, Éric Lefèvre*

*LGI2A EA 3926
Université d'Artois - Faculté des Sciences Appliquées
Technoparc Futura 62400 Béthune - France
{gildas.morvan, alexandre.veremme}@fsa.univ-artois.fr
{david.mercier, eric.lefevre}@univ-artois.fr
**ERASM - HEI
13 rue de Toul 59046 Lille - France
alexandre.veremme@hei.fr

Résumé. La représentation et la gestion d'informations dans le cadre des systèmes complexes, où les flux de données sont importants, posent bien souvent problème, en particulier lorsque des prises de décisions sont souhaitées. Cela est d'autant plus difficile lorsque ces informations sont imprécises et incertaines. En Entomologie Médico-Légale, un expert, déterminant la date de la mort d'une victime retrouvée dans un écosystème complexe, doit ainsi traiter ce type d'informations imparfaites tout en restant objectif et prudent. Pour résoudre ces problématiques, un système informatique d'aide à la décision a été mis en œuvre. Dans ce système, afin d'obtenir une décision de meilleure qualité, une étape de fusion est intégrée. Celle-ci permet de compiler les résultats du système et repose sur le *Modèle des Croyances Tranférables* (MCT).

1 Introduction

Dans le cadre d'une enquête criminelle, il est essentiel d'obtenir un maximum d'informations sur les conditions dans lesquelles le crime a été commis. De nombreuses techniques ont vu le jour ces dernières décennies afin d'exploiter au mieux les indices présents sur une scène de crime. L'une de ces techniques, l'Entomologie Médico-Légale, exploite les indices entomologiques (*i.e.* les insectes nécrophages retrouvés sur ou à proximité de la victime) afin de déterminer l'intervalle post-moterm (IPM). Cette technique soulève un intérêt croissant dans les polices scientifiques du monde entier. En effet, parmi l'ensemble des méthodes de datation de la mort, l'Entomologie Médico-Légale est la seule utilisable en pratique lorsque l'IPM est supérieur à 48 heures. Cependant, la diversité des modèles utilisés par les différents experts amène à s'interroger sur la fiabilité des résultats fournis. Il semble donc judicieux d'intégrer une approche de fusion afin de fournir un résultat global cohérent et objectif. Cet article présente une première solution à ce problème développée dans le cadre du *Modèle des Croyances Transférables* (MCT). L'organisation de cet article est la suivante. Dans la section 2, les fondements biologiques de l'Entomologie Médico-Légale sont présentés. La section 3 introduit

les bases de l'architecture du système d'aide à la décision. Les concepts de base du MCT sont rappelés dans la section 4. La section 5 présente les résultats expérimentaux de l'approche proposée. Enfin, la dernière section (Section 6) énonce les différentes perspectives liées à ce travail.

2 Introduction à l'Entomologie Médico-Légale

2.1 Développement des diptères nécrophages

Les méthodes modernes de datation utilisées en Entomologie Médico-Légale se basent essentiellement sur l'utilisation de modèles de développement des diptères (mouches) nécrophages, premières espèces à coloniser un cadavre. Ces modèles considèrent que la vitesse de développement des diptères dépend — comme pour de nombreuses autres espèces, animales ou végétales — de la température (Stinner et al., 1974) :

$$\frac{da}{dt} = f(T(t)) \tag{1}$$

Où $\frac{da}{dt}$ représente la vitesse de développement, $T(t)$ la température T ressentie par un individu en fonction du temps t, et f un modèle de développement. De nombreux modèles de développement, plus ou moins fiables et faciles à utiliser en pratique, ont vu le jour. Le lecteur intéressé pourra se référer à Wagner et al. (1984) pour une étude comparative de ces modèles.

2.2 Expertises entomologiques

S'il existe une méthode de datation basée sur l'étude des *escouades* ou groupes d'insectes colonisant le corps à des stades particuliers de la décomposition (Mégnin, 1894), la méthode la plus utilisée aujourd'hui consiste à résoudre numériquement l'équation (1). Il est ainsi très simple, *dans un environnement contrôlé*, de prédire le temps nécessaire au développement d'un insecte et de dater le moment t_1 auquel il a été pondu à partir du taux de développement accumulé Δa par l'insecte au moment t_2 de la découverte du corps :

$$\Delta a = \int_{t_1}^{t_2} f(T(t))dt \tag{2}$$

Dans un écosystème complexe, il est en revanche beaucoup plus difficile de savoir à quelles températures s'est développé un insecte. Les entomologistes considèrent généralement que la température ressentie par les insectes retrouvés sur le corps de la victime est égale à la température relevée par la station météorologique la plus proche. Cela est inexact pour plusieurs raisons. Tout d'abord, notons que les victimes sont rarement retrouvées au pied d'une station météorologique et que l'environnement dans lequel un cadavre est retrouvé peut être sujet à des microclimats. De plus, un corps a une certaine inertie thermique qui absorbe les variations de la température extérieure. Enfin, les larves ont un comportement grégaire induisant des augmentations locales de température. Cette troisième cause d'imprécision a été signalée dans de nombreux articles comme une source importante d'erreur dans l'estimation de l'intervalle post-mortem (Marchenko, 2001). Cependant aucune méthode ne permet à l'heure actuelle de prendre en compte ce phénomène.

3 Le projet ForenSeek

Le projet ForenSeek (Morvan et al., 2007a) a pour objectif le développement d'un Système Informatique d'Aide à la Décision (SIAD) destiné aux experts entomologistes. Ce projet ambitionne de résoudre les problèmes liés aux méthodes classiques énoncés plus haut. Les différents composants du SIAD : un modèle multi-agents du développement des diptères nécrophages ainsi qu'un système de raisonnement abductif sont présentés dans cette section. Leur interfaçage et leur utilisation pour la détermination d'un IPM sont détaillés par la suite.

3.1 Modélisation multi-agents du système cadavre - entomofaune

Afin d'améliorer la qualité des expertises entomologiques, il est nécessaire de prendre en compte l'ensemble des paramètres écosystémiques intervenant dans le développement des diptères. Les augmentations de température émergeant des interactions entre les larves, il est alors naturel de se tourner vers la modélisation multi-agents pour représenter l'écosystème *cadavre - entomofaune*. Ainsi chaque *acteur* du système est modélisé sous la forme d'un agent, *i.e.* un système informatique automonome, situé dans un environnement et capable de communiquer, directement ou à travers l'environnement, avec d'autres agents. Il est ainsi possible de simuler le comportement thermique et nutritionnel d'une masse de larves et son influence sur le développement des individus qui la constitue.

Le modèle ne sera que brièvement présenté. Le lecteur intéressé pourra se référer à Morvan et al. (2007b) pour une présentation plus complète et formelle du modèle ainsi que de son cadre d'utilisation. Le modèle est composé d'un environnement maillé et de plusieurs types d'agents encapsulant un ensemble de sous-modèles. Un sous-modèle calcule une ou plusieurs propriétés systémiques spécifiques. Le tableau TAB. 1 récapitule les différents types d'agents et l'environnement présents dans le modèle ainsi que les sous-modèles qu'ils encapsulent.

agent	acteur(s) modélisé(s)	sous-modèle(s) encapsulé(s)
larva	larve de diptère	- déplacement - développement - émission de chaleur - nutrition - mortalité
layer	femelles gravides	- population - ponte - attraction
environnement	corps humain	- comportement thermique

TAB. 1 – *Description des agents du modèle.*

L'agent *larva* modélise une larve de diptère. Il se déplace dans l'environnement en fonction de signaux qu'il perçoit et l'utilise pour y puiser les ressources nécessaires à son développement. Lorsque l'agent se nourrit au sein d'un agrégat, il augmente localement la température de l'environnement. L'agent *layer* modélise la population de femelles gravides, *i.e.* portant des œufs, présente dans l'écosystème ainsi que le comportement de ponte d'une femelle. De

nombreuses espèces peuvent coloniser un corps. Les différents sous-modèles sont donc paramétrables en fonction de l'espèce. De plus, pour une même propriété, *e.g.* le taux de développement, différents sous-modèles ont été développés. On appelle donc *modèle* une association particulière de sous-modèles pour un ensemble d'espèces donné.

3.2 Modélisation du processus d'expertise

Définitions et présentation du modèle de raisonnement L'*abduction* est une forme de raisonnement identifiée par C.S. Pierce (1932) consistant à déterminer la cause la plus probable d'une *observation surprenante*. Le processus d'expertise entomologique développé dans ForenSeek implémente un modèle de raisonnement abductif. En effet, le modèle présenté dans la section précédente est purement prédictif ; de par sa nature computationnelle il ne peut être utilisé directement pour produire des rétrodictions (*i.e.* des prédictions à propos du passé). Il peut être en revanche utilisé pour tester l'adéquation entre une hypothèse (une heure possible de la mort) et un ensemble d'observations (*e.g.* le taux de développement d'insectes retrouvés sur le corps). Ainsi, le modèle sera utilisé afin de produire une mesure de cohérence entre une hypothèse et un ensemble d'observations. Formellement, un raisonnement peut être représenté sous la forme d'un syllogisme $< c, r, C >$ où c est un *cas*, *e.g.* Socrate est un homme, r est une *règle*, *e.g.* les hommes sont mortels, et C est la *conclusion*, *e.g.* Socrate est mortel, déduite à partir de c et r. Cette représentation permet d'identifier trois types (ou modes) de raisonnement (Aliseda-Llera, 1998) :

Raisonnement	Mode
$c, r \vdash C$	Déduction
$c, C \mid\sim_i r$	Induction
$r, C \mid\sim_a c$	Abduction

TAB. 2 – *Les différents modes de raisonnement dans le modèle peircien.*

Dans des écrits postérieurs, Peirce redéfinit l'abduction comme la capacité à sélectionner parmi un ensemble d'hypothèses, celle qui pourrait le plus probablement expliquer l'observation surprenante. Si Peirce attribue cette capacité de sélection à l'*intuition*, plus récemment des philosophes ont étudié la nature stratégique (heuristique) du raisonnement abductif (Hintikka, 2001; McGrew, 2003; Paavola, 2003). Il existe de nombreux termes qui recouvrent plus ou moins la notion d'abduction (raisonnement en faveur de la meilleure explication, rétroduction, *etc.*). De même, la notion d'abduction recouvre à la fois le *processus* de détermination de la meilleure explication et le *résultat* de ce processus. Dans cet article, nous utiliserons les termes de "raisonnement abductif" et d'"abduction" pour désigner processus et résultat.

En Intelligence Artificielle, l'abduction a été essentiellement étudiée dans le cadre du raisonnement diagnostique. Ces recherches ont permis le développement de formalismes et de méthodes de résolution. Cependant, le raisonnement diagnostique est un cas particulier de raisonnement abductif et les méthodes de résolution proposées ne semblent pas cognitivement plausibles. En toute généralité, un raisonnement abductif est un processus récursif de sélection et d'évaluation d'une hypothèse. Dans cet article, seule la méthode d'évaluation sera exposée.

Évaluation d'une hypothèse Nous avons développé un système à base de règles permettant de déterminer efficacement si une hypothèse est compatible avec un ensemble d'observations. Ce système encode les informations entomologiques fournies par l'expert sous la forme d'un ensemble de règles qui sera intégré aux agents lors des simulations. Si aucune des règles n'est violée durant la simulation, nous considérerons la simulation comme valide (*i.e.* compatible avec les observations entomologiques). Le modèle étant stochastique, il sera nécessaire de réaliser un nombre de simulations statistiquement représentatif afin d'évaluer correctement une hypothèse. La mesure de cohérence $c_{j,k}$ associée à une hypothèse ω_k et à un modèle Mod_j est calculée comme suit :

$$c_{j,k} = \frac{s^v_{j,k}}{s^t_{j,k}} \tag{3}$$

Où $s^v_{j,k}$ et $s^t_{j,k}$ représentent respectivement le nombre de simulations valides et le nombre total de simulations réalisées en utilisant le modèle Mod_j et l'hypothèse ω_k.

Afin d'améliorer l'estimation de l'IPM, il semble judicieux d'utiliser plusieurs modèles ainsi que leurs degrés de cohérence. En effet, l'utilisation d'informations variées et diverses permet d'aboutir à une décision plus robuste qu'une décision prise par une information unique. Cette étape dite de fusion est présentée dans le cadre du *Modèle des Croyances Transférables* (Smets et Kennes, 1994; Smets, 1998).

4 Fusion de données dans le cadre du Modèle des Croyances Transférables

L'approche de fusion utilisée ici repose sur le *Modèle des Croyances Transférables* (MCT). Ce modèle est basé sur une interprétation subjective de la théorie des fonctions de croyance (Dempster, 1968; Shafer, 1976). Il permet de prendre en compte les imperfections des informations, aussi bien les incertitudes que les imprécisions. Dans le cadre du MCT, deux niveaux sont distingués :

- le niveau *crédal*, où sont représentées (partie statique) et manipulées (partie dynamique) les informations disponibles,
- le niveau *pignistique* où la décision est prise.

4.1 Représentation des connaissances

Soit $\Omega = \{\omega_1, \cdots, \omega_k, \cdots, \omega_K\}$ un ensemble fini, appelé cadre de discernement. Une fonction de croyance bel est une mesure floue non additive de 2^Ω dans $[0, 1]$ définie par :

$$bel(A) \triangleq \sum_{\emptyset \neq B \subseteq A} m(B) \qquad \forall\, A \subseteq \Omega \tag{4}$$

où m, appelé généralement fonction de masse, est une fonction de 2^Ω dans $[0, 1]$ qui vérifie : $\sum_{A \subseteq \Omega} m(A) = 1$. Chaque sous-ensemble $A \subseteq \Omega$ tel que $m(A) > 0$ est appelé élément focal de m. Ainsi, la masse $m(A)$ représente le degré de croyance attribué à la proposition A et qui n'a pas pu, compte tenu de l'état de la connaissance, être affecté à un sous-ensemble plus

spécifique que A. Notons que dans le MCT la fonction de masse m peut être non normalisée. Il est alors possible d'avoir $m(\emptyset)$ strictement positif. Cette interprétation, appelée monde ouvert, permet de relâcher la contrainte d'exhaustivité du cadre de discernement initialement introduite par Shafer (1976). Les fonctions de croyance sont de nos jours reconnues pour la modélisation des informations imprécises et incertaines (de l'ignorance totale à la connaissance complète). Ainsi, une situation d'ignorance complète correspond à la fonction de croyance vide définie par $m(\Omega) = 1$. La connaissance parfaite sera représentée par une fonction de croyance certaine et précise, c'est-à-dire une fonction où la totalité de la masse est allouée à un singleton unique de Ω.

La seconde étape au niveau credal correspond à la révision des croyances. Considérons une masse $m(A)$ strictemement positive allouée à un sous-ensemble A de Ω. Si nous apprenons avec certitude que la vérité se situe dans un sous-ensemble B de Ω, la masse initialement allouée à A devra alors être transférée à $A \cap B$. Cette règle correspond à la règle de conditionnement non normalisée de Dempster. Le terme "*modèle de croyance transférable*" est issu de ce transfert de masse créditée à des sous-ensembles en fonction des informations disponibles. Issu de ce principe de conditionnement, la règle de combinaison conjonctive permet de fusionner deux fonctions de masse, m_1 et m_2, issues de sources d'informations distinctes et fiables. Cette règle est définie par :

$$m_1 \textcircled{\cap} m_2(A) = \sum_{B \cap C = A} m_1(B) m_2(C), \ \forall A \subseteq \Omega \tag{5}$$

Il est à noter que cette règle, généralement appelée règle de Dempster non normalisée, permet de combiner des informations incertaines extraites sous forme de fonctions de croyance. Si nécessaire, la condition $m(\emptyset) = 0$ peut être retrouvée en divisant chaque masse par un coefficient de normalisation. L'opération résultante, appelée règle de Dempster et notée $\oplus$, est définie $\forall\ A \subseteq \Omega$ par :

$$m_1 \oplus m_2(A) \triangleq \frac{m_1 \textcircled{\cap} m_2(A)}{1 - m(\emptyset)} \tag{6}$$

où la quantité $m(\emptyset)$ est appelée degré de conflit entre les fonctions m_1 et m_2 et peut être calculée en utilisant l'équation suivante :

$$m_1 \textcircled{\cap} m_2(\emptyset) = \sum_{B \cap C = \emptyset} m_1(B) m_2(C). \tag{7}$$

L'utilisation de la règle de Dempster est possible si et seulement si m_1 et m_2 ne sont pas en conflit total, c'est-à-dire s'il existe deux éléments focaux B et C de m_1 et m_2 qui satisfassent $B \cap C \neq \emptyset$. Cette règle possède, tout comme la combinaison conjonctive, des propriétés intéressantes comme l'associativité, la commutativité et la non-idempotence. Toutefois, la règle de Dempster a été très discutée (Zadeh, 1979; Yager, 1987; Smets, 1990), c'est pourquoi d'autres solutions ont vu le jour (Yager, 1983; Smets, 1993; Lefevre et al., 2002). Une description de la gestion du conflit dans le cadre du MCT est présentée de manière plus détaillée dans Smets et Ristic (2004).

Outre ces deux outils, les fonctions de croyance peuvent être affaiblies, déconditionnées, marginalisées, étendues entre elles (Smets et Kennes, 1994; Smets, 1993). Cette palette d'outils de manipulation des fonctions de croyance explique l'intérêt suscité par cette théorie dans le domaine de la fusion d'informations.

4.2 Niveau pignistique

L'étape d'aggrégation précédemment définie permet ainsi d'obtenir un résumé exhaustif de l'information sous forme d'une fonction de masse unique m qui est utilisée pour la prise de décision. En basant son raisonnement sur des arguments de rationnalité développés dans le MCT, Smets et Kennes (1994) proposent de transformer m en une fonction de probabilité $BetP$ définie sur Ω (appelée fonction de probabilité *pignistique*) qui se formalise pour tout $\omega_k \in \Omega$ par :

$$BetP(\omega_k) = \frac{1}{1 - m(\emptyset)} \sum_{A \ni \omega_k} \frac{m(A)}{|A|} \tag{8}$$

où $|A|$ représente la cardinalité de $A \subseteq \Omega$. Dans cette transformation, la masse de croyance $m(A)$ est uniformément distribuée parmi les éléments de A. Une justification de cette transformation a été donnée par Smets (2005). A partir de cette distribution de probabilité, il est alors possible d'utiliser les outils classiques de la théorie de la décision statistique. Cette théorie préconise de choisir l'action pour laquelle l'espérance du coût est la plus faible. Ainsi, on peut définir l'espérance pignistique d'une fonction $f : \Omega \rightarrow \mathbb{R}$ comme son espérance mathématique relativement à $BetP$:

$$E_{BetP}(f) = \sum_{\omega_k \in \Omega} f(\omega_k) BetP(\omega_k). \tag{9}$$

Supposons que l'on considère le problème du choix d'une action parmi un ensemble fini $\mathcal{A} = \{\delta_1, \ldots, \delta_K\}$. L'action δ_k correspond à l'action de choisir l'hypothèse $\omega_k \in \{\omega_1, \ldots, \omega_K\}$. La mise en œuvre d'une action δ_i alors que la réalité est l'hypothèse ω_k est supposée entraîner un coût noté $\lambda(\delta_i|\omega_k)$. Le risque conditionnel de décider δ_i s'exprime alors de la façon suivante :

$$R_{Bet}(\delta_i) = \sum_{\omega_k \in \Omega} \lambda(\delta_i|\omega_k) BetP(\omega_k). \tag{10}$$

L'action $\delta \in \mathcal{A}$ qui minimise ce risque est celle qui sera retenue. Dans le cas de coûts $\{0, 1\}$, la minimisation du risque conditionnel revient à choisir l'hypothèse de plus grande probabilité pignistique. D'autres règles ont été développées pour la prise de décision (Denœux, 1997). Dans la suite, la prise de décision repose sur le maximum de probabilité pignistique.

5 Mise en oeuvre et Résultats

5.1 Construction des fonctions de croyance

L'objectif, dans le cadre de l'Entomologie Médico-Légale, est d'estimer de façon la plus précise possible l'IPM. Pour cela, il est alors préférable d'utiliser un ensemble de modèles fournissant pour chaque hypothèse (représentant l'heure du décès possible) une mesure de cohérence avec les observations entomologiques.

De manière plus formelle, soit ω_k une hypothèse correspondant à une heure possible du décès. Le cadre de discernement s'écrit alors : $\Omega = \{\omega_1, \ldots, \omega_K\}$. Pour chaque modèle Mod_j (avec $j \in [1, J]$) une mesure de cohérence est déterminée (Section 3.2). On peut alors, à partir de cette mesure, construire une fonction de masse pour chaque modèle selon chaque hypothèse.

Cette fonction peut s'écrire de la façon suivante :

$$\left\{ \begin{array}{rcl} m_{j,k}(\{\omega_k\}) & = & \beta_j . \frac{s^v_{j,k}}{s^t_{j,k}} \\ m_{j,k}(\Omega) & = & 1 - \beta_j . \frac{s^v_{j,k}}{s^t_{j,k}} \end{array} \right. \tag{11}$$

où β_j correspond à un coefficient de confiance associé au modèle Mod_j. Ce coefficient peut être fixé ou estimé en fonction, par exemple, des résultats obtenus par le modèle Mod_j dans un certain contexte (Guo et al., 2006). Pour la suite de notre étude, ce coefficient est fixé, dans un premier temps, à 0.9 de manière heuristique.

Afin de déterminer la fonction de masse selon chaque modèle, on peut utiliser l'équation (5). La masse résultante s'écrit alors :

$$m_j = \textcircled{\cap}_{k=1}^{K} m_{j,k} \tag{12}$$

De manière identique, la fonction de masse représentant l'ensemble des connaissances s'exprime de la manière suivante :

$$m = \textcircled{\cap}_{j=1}^{J} m_j \tag{13}$$

A partir de cette fonction de masse unique, le calcul de la probabilité pignistique peut se faire à l'aide de l'équation (8). Le risque pignistique (équation (10)) est alors calculé. L'hypothèse retenue est celle qui permet de minimiser ce risque.

5.2 Expérimentation

De par la difficulté à obtenir des données sur des études réelles, la méthode proposée n'a pu être appliquée que sur un seul cas. Cette expérience considère un ensemble de 141 hypothèses, *i.e.* 141 heures entre le moment de la disparition de la victime et de la découverte du corps. L'expertise a été réalisée en utilisant une combinaison de deux sous-modèles de développement, le modèle des *degrés jours accumulés* (ADD) (Wagner et al., 1984) et le modèle de Stinner (Stinner et al., 1974), et trois sous-modèles d'émission de chaleur, soit six modèles différents.

Pour chaque modèle, une mesure de cohérence entre une hypothèse concernant l'heure du décès et les relevés entomologiques est déterminée. Pour ce test, 141 hypothèses sont testées pour chaque modèle. Ces hypothèses sont identiques pour tous les modèles et constituent le cadre de discernement. Les mesures de cohérence obtenues sont présentées sur la figure FIG. 1.

L'équation (11) permet d'obtenir la masse de croyance reflétant la cohérence d'une hypothèse par rapport au relevé sur le terrain. Après la fusion de ces fonctions initiales, une fonction de masse synthétisant l'information pour chaque modèle est obtenue. Ces fonctions de masse sont présentées sur la figure FIG. 2.

Après une dernière étape de fusion, la probabilité pignistique peut être calculée. Son évolution est représentée sur la figure FIG. 3 pour chaque hypothèse.

Dans le cas d'une prise de décision, l'hypothèse sélectionnée serait ω_{79}, cette hypothèse possédant la probabilité pignistique maximale. On constate ainsi que le processus de fusion décrit dans le cadre de cet article permet de réduire de manière importante le nombre d'hypothèses cohérentes avec les relevés entomologiques. En effet, selon les modèles, il y avait entre treize et vingt-trois hypothèses cohérentes. Après la fusion, il n'en reste plus qu'une seule ce qui permet de cerner de façon plus précise l'heure du décès.

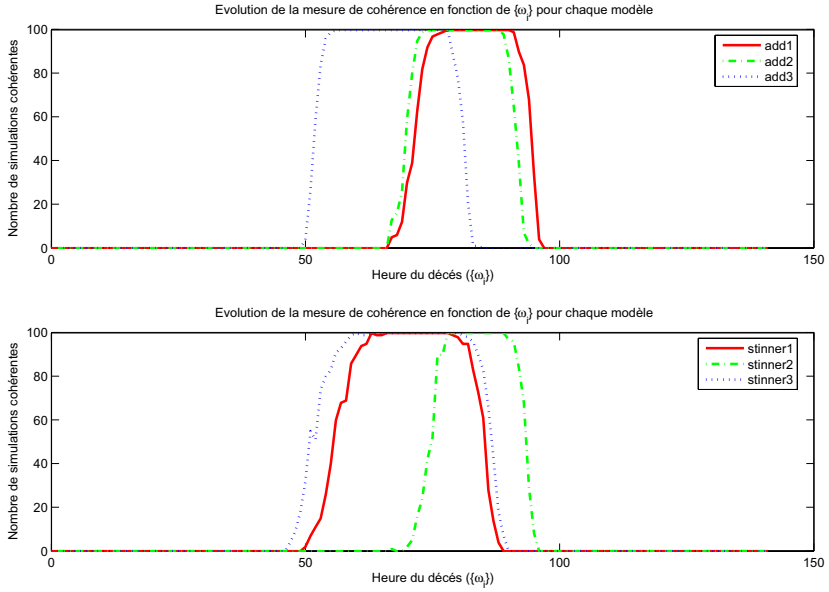


FIG. 1 – *Représentation de la mesure de cohérence pour chaque hypothèse selon le modèle étudié.*

6 Conclusions et perspectives

Même si les premiers résultats apparaissent très intéressants et prometteurs, ceux-ci sont toutefois à nuancer puisque basés sur un unique exemple théorique. Dans ce cas précis, l'association du modèle prédictif de décomposition, du modèle abductif d'expertise et du système de fusion basé sur le modèle des croyances transférables permet de réduire considérablement l'intervalle des hypothèses, en cernant de manière plus précise les heures probables de la mort, et semble donc être, pour les médecins légistes, une excellente alternative aux méthodes traditionnelles.

Le projet n'en est encore qu'à ses débuts mais déjà les perspectives de recherche sont nombreuses et cela quelque soit le niveau de l'application. En ce qui concerne en particulier le processus de fusion, le modèle de croyances transférables convient à la problématique médico-légale mais des points essentiels devront être abordés et approfondis : la détermination des valorisations — affaiblissements et renforcements — des fonctions de croyances des différents modèles, ou la prise en compte des incertitudes locales de ces croyances ou encore, plus généralement, l'optimisation du processus global de fusion. En effet, des modèles identiquement fiables ont été ici considérés dans la fusion et une amélioration du schéma actuel, grâce à une boucle de rétroaction couplée à un système d'apprentissage et de mémorisation, devrait permettre d'améliorer l'objectivité et la prudence du système. De plus, la construction des fonctions de croyance ne prend pas en compte actuellement les incertitudes liées à l'aspect

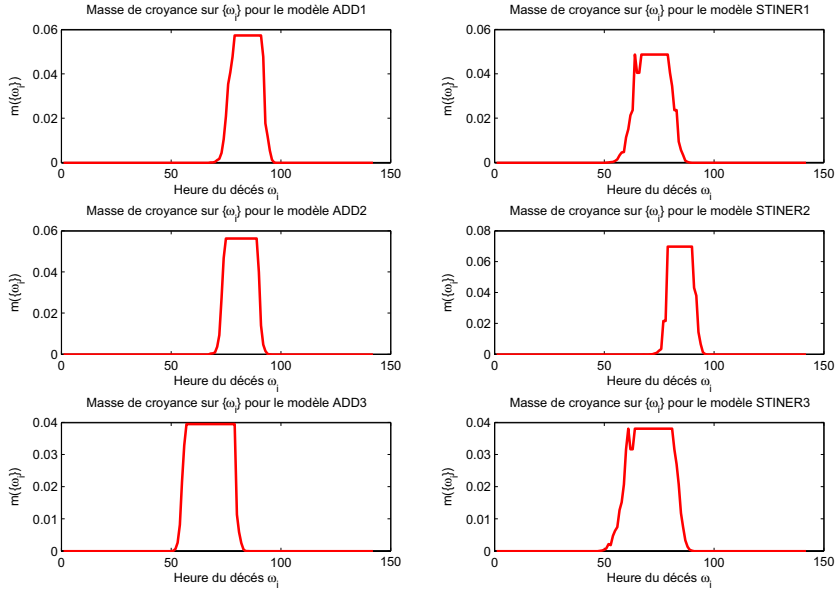


FIG. 2 – *Représentation de la fonction de masse pour chaque hypothèse selon le modèle étudié.*

stochastique du modèle de décomposition. Enfin, considérant l'importance du nombre d'hypothèses et des temps de calculs considérables inhérents à ce système complexe, il sera nécessaire d'optimiser et d'améliorer le processus de fusion.

Remerciements

Ce travail est financé par le Ministère de la Recherche et la Fondation Norbert Ségard. Ce travail est réalisé dans le cadre d'une collaboration entre différents laboratoires. Les auteurs tiennent ainsi à remercier Daniel Dupont (Erasm - HEI Lille), Philippe Kubiak (LAGIS - École Centrale de Lille), Gilles Goncalves, Daniel Jolly (LGI2A - Université d'Artois), Benoît Bourel et Damien Charabidze (Laboratoire d'Entomologie - Institut de Médecine Légale de Lille) pour leur soutien.

Références

Aliseda-Llera, A. (1998). *Seeking Explanations : Abduction in Logic, Philosophy of Science and Artificial Intelligence*. Ph. D. thesis, Stanford University, Department of Computer Science.

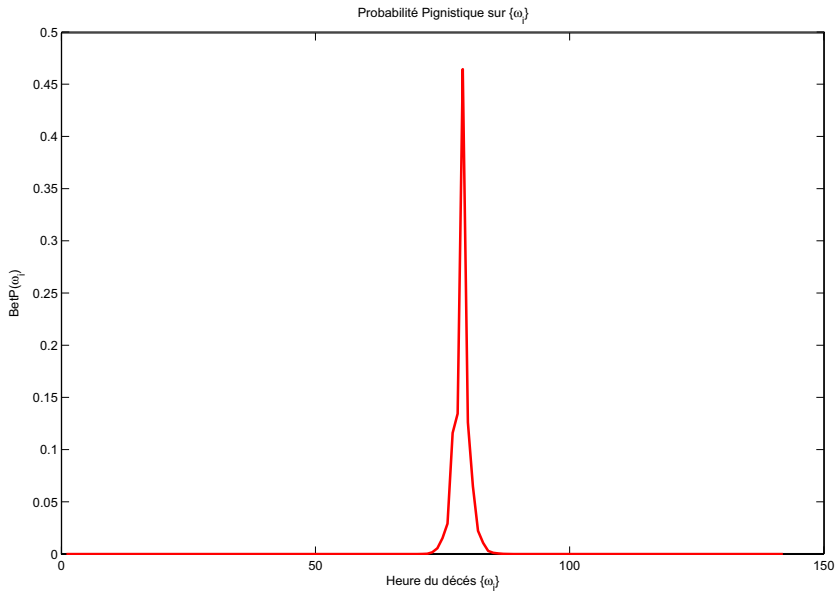


FIG. 3 – *Probabilité pignistique de chaque hypothèse* $\{\omega_i\}$.

Dempster, A. (1968). A generalization of bayesian inference. *Journal of Royal Statistical Society, Serie B 30*, 205–247.

Denœux, T. (1997). Analysis of evidence-theoretic decision rules for pattern classification. *Pattern Recognition 30*(7), 1095–1107.

Guo, H., W. Shi, et Y. Deng (2006). Evaluating sensor reliability in classification problems based on Evidence Theory. *IEEE Transactions on Sytems, Man and Cybernetics - Part B 36*(5), 970–981.

Hintikka, J. (2001). Is logic the key to all good reasoning ? *Argumentation 15*, 35–57.

Lefevre, E., O. Colot, et P. Vannoorenberghe (2002). Belief function combination and conflict management. *Information Fusion 3*(2), 149–162.

Marchenko, M. I. (2001). Medicolegal relevance of cadaver entomofauna for the determination of the time of death. *Forensic Science International 120*, 89–120.

McGrew, T. J. (2003). Confirmation, heuristics, and explanatory reasoning. *British Journal for the Philosophy of Science 54*, 553–567.

Mégnin, P. (1894). *La faune des cadavres*. G. Masson.

Morvan, G., D. Charabidze, et A. Veremme (2007a). http ://www.forenseek.org.

Morvan, G., D. Jolly, D. Dupont, et P. Kubiak (2007b). A decision support system for forensic entomology. In *Proceedings of the* 6^{th} *EUROSIM congress*.

Paavola, S. (2003). Abduction as a logic and methodology of discovery : the importance of strategies. *Foundations of Science 9*(3), 267–283.

Pierce, C. S. (1932). *Collected Papers of Charles Sanders Pierce*, Volume 2. Havard University Press.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton, New Jersey : Princeton University Press.

Smets, P. (1990). The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*(5), 447–458.

Smets, P. (1993). Belief functions : The disjunctive rule of combination and the generalized bayesian theorem. *International Journal of Approximate Reasoning 9*, 1–35.

Smets, P. (1998). *The Transferable Belief Model for Quantified Beleif Representation*, pp. 267–301. Kluwer Academic Publishers, Dordrecht, The Nertherlands.

Smets, P. (2005). Decision making in the TBM : the necessity of the pignistic transformation. *International Journal Of Approximate Reasoning 38*, 133–147.

Smets, P. et R. Kennes (1994). The transferable belief model. *Artificial Intelligence 66*(2), 191–234.

Smets, P. et B. Ristic (2004). Kalman filter and joint tracking and classification in the TBM framework. In P. Svensson et J. Schubert (Eds.), *Proceedings of the Seventh International Conference on Information Fusion*, Volume I, Mountain View, CA, pp. 46–53. International Society of Information Fusion.

Stinner, R. E., A. P. Gutierrez, et G. D. Butler Jr (1974). An algorithm for temperature-dependant growth rate simulation. *The Canadian Entomologist 106*, 519–524.

Wagner, T. L., H.-I. Wu, P. J. Sharpe, R. M. Schoolfield, et R. N. Coulson (1984). Modeling insect development rates : A literature review and application of a biophysical model. *Annals of the Entomological Society of America 77*(2), 208–225.

Yager, R. (1983). Hedging in the combination of evidence. *Journal of Information and Optimization Sciences 4*(1), 73–81.

Yager, R. (1987). On the dempster-shafer framework and new combination rules. *Information Sciences 41*, 93–138.

Zadeh, L. (1979). *On the Validity of Dempster's Rule of Combination of Evidence*. University of California, Berkeley. ERL Memo M79/24.

Summary

Representing and dealing information in the framework of complex systems, where data flows are important can often be problematic, especially when decision making is required and the more imprecise and uncertain the information, the more difficult. A Forensic Entomology expert determining the time since the death of a victim found in a complex ecosystem has to process this type of imperfect information while keeping objective and careful. To solve those problems, a computer based decision support system has been set up. The transferable belief model (TBM) is then used to compile the results.

Clustering de trajectoires contraintes par un réseau

Ahmed KHARRAT*, Karine ZEITOUNI*
Sami FAIZ**

* Laboratoire PRISM 45, avenue des états unis
78035 Versailles
Ahmed.kharrat@prism.uvsq.fr, Karine.Zeintouni@prism.uvsq.fr
http://www.prism.uvsq.fr/users/karima/
**INSAT, Tunisie
Sami.Faiz@insat.rnu.tn

Résumé. Cet article se place dans le cadre de la fouille de données spatiotemporelle. Il propose une méthode de classification automatique des trajectoires d'objets mobiles dont le mouvement est contraint par un réseau routier. Cette méthode est basée sur la densité du réseau. Elle se compose de deux phases : la première découvre des chemins denses, puis la seconde s'appuie sur ces chemins denses pour former des groupes de trajectoires - ou plus exactement de sous-trajectoires - qui leurs sont similaires. Des paramètres donnés par l'utilisateur permettent de définir le seuil de densité pour les chemins denses ainsi que le seuil de similarité au sein des clusters.

1 Introduction

La gestion d'objets mobiles a connu un regain d'intérêt ces dernières années, encouragée par la disponibilité d'outils de localisation, à travers les téléphones cellulaires, les GPS ou récemment le RFID (Radio Frequency IDentification). Il devient possible de générer des bases de trajectoires d'objets mobiles à une large échelle. Par exemple, des applications de contrôle et de prévision du trafic utilisent des flottes de véhicules équipées de GPS comme des sondeurs du trafic. Leurs relevés GPS (appelés *Floating Car Data*) alimentent une base de trajectoires d'objets mobiles. Dans ce cas, comme dans la majorité des cas, le mouvement de l'objet mobile est contraint par le réseau. On parle de trajectoire contrainte par un réseau.

Ces applications ont généré de nouveaux problèmes qui ont motivé la recherche sur la gestion d'objets mobiles, en général et sur la fouille de données spatio-temporelle, en particulier. Le clustering des trajectoires fait partie de ces recherches. C'est une technique de fouille de données (ou datamining) largement utilisée dans des applications telles que l'étude de marché, l'analyse financière ou encore le traitement d'images. Plusieurs types d'algorithmes de clustering ont été proposés dont K-means (Lloyd, 1981), BIRCH (Zhang et al., 1996), DBSCAN (Ester et al., 1996) et OPTICS (Ankerst et al., 1999). Les recherches récentes sur le clustering de trajectoires d'objets mobiles ont utilisé ces algorithmes en les adaptant au domaine étudié (Lee et al., 2007).

Notre travail s'inspire notamment de l'algorithme DBSCAN (Ester et al., 1996). L'idée clé derrière notre approche est que la connaissance de la densité du trafic sur le réseau per-

mettrait de guider le clustering des trajectoires. Dans un premier temps, nous définissons la similarité entre sections de routes et l'utilisons pour les grouper en chemins denses. Dans un deuxième temps, nous proposons une mesure de similarité entre trajectoires, puis nous nous en servons pour former des groupes autour des chemins denses. Comme dans (Lee et al., 2007), le temps n'est pas considéré dans notre approche (on parle de temps relaxé ou time relaxed). Sommairement, les contributions de cet article sont les suivantes :

- Nous proposons une méthode de clustering de trajectoires d'objets mobiles complètement novatrice basée sur la densité du réseau. Tout comme l'approche de (Lee et al., 2007), cette méthode ne considère pas l'aspect temporel - bien qu'elle respecte le sens de parcours des trajectoires - et regroupe des sous-trajectoires plutôt que des trajectoires entières. Ainsi, une trajectoire peut appartenir à plusieurs clusters. Cependant, la méthode proposée tire profit des propriétés du réseau pour générer des groupes pertinents.
- Nous implémentons notre approche par un algorithme en deux phases permettant dans la première de grouper les sections de route et dans la seconde de grouper les trajectoires.
- Nous définissons de nouvelles fonctions de similarité.

Le reste de cet article est organisé comme suit. Nous détaillerons dans la section 2 un état de l'art sur la similarité et le clustering des trajectoires. Nous expliquerons ensuite la démarche de clustering dans la section 3. Nous présenterons dans la section 4 la première phase de l'algorithme - appelé NETSCAN - pour le clustering des sections de route. Nous décrirons ensuite la deuxième phase de l'algorithme. Enfin, nous conclurons cet article à la section 6 et proposerons des pistes pour la poursuite de cette recherche.

2 Etat de l'art

La recherche sur le clustering de trajectoires d'objets mobiles et étroitement liée à trois sujets que sont la représentation de trajectoires, la similarité et les algorithmes de clustering proprement dits. De manière orthogonale, nous distinguons la prise en compte des critères suivants : l'aspect contraint ou libre de la trajectoire, l'aspect temporel, le respect du sens de parcours et enfin la prise en compte d'une partie ou de la totalité de la trajectoire dans le clustering. Cette section décrit les principaux travaux traitant de ces trois sujets en les situant par rapport aux critères ci-dessus.

Concernant la représentation de trajectoires, celle-ci peut-être soit géométrique (Lee et al., 2007), soit symbolique (Hadjieleftheriou et al., 2005) pour les trajectoires évoluant dans un réseau. En effet, si l'on connaît d'avance la géométrie et la topologie du réseau, une représentation naturelle de la trajectoire peut être donnée par la liste des sections traversées avec éventuellement l'instant auquel l'objet est passé d'une section à une autre si l'on représente l'aspect temporel. Cette représentation est très précise au niveau spatial, mais peut-être moins précise au niveau temporel. Néanmoins, elle peut suffire dans de nombreux cas et particulièrement dans notre contexte où le temps n'est pas pris en compte.

Concernant les travaux sur la similarité de trajectoires d'objets mobiles, nous citons d'abord les travaux dans le contexte de trajectoires libres, puis pour des trajectoires contraintes.

Yanagiswa et al. (2003) proposent d'abord une modélisation par la forme 2D. L'aspect spatial y est représenté par des vecteurs de même longueur, tandis que la description spatio-temporelle est obtenue par des vecteurs de longueurs différentes, mais à intervalle de temps régulier. Ainsi, la même formule de distance euclidienne entre ces vecteurs permet de définir la similarité spatiale et spatio-temporelle. Shim et Chang (2003) considèrent la similarité des sous-trajectoires et ont proposé un algorithme de distance 'K-warping'. Lin et al. (2005) ne considèrent que l'aspect spatial dans leur calcule de similarité et proposent une nouvelle distance OWD (One Way Distance). On trouve des travaux similaires dans Valachos et al. (2003), Sakurai et al. (2005), et Chen et al. (2005). Valachos et al. (2002) se sont intéressés aux valeurs extrêmes (outliers) qui dégradent les performances de la distance euclidienne et la déformation du temps (le time warping). Par conséquent, ils proposent l'utilisation d'une fonction de distance non métrique et qui se base sur l'algorithme de la plus longue sous séquence commune (LCSS) avec la conjonction de la fonction SigmoidMatch pour aligner (matching) les séquences de deux trajectoires. Zeinalipour-Yazti et al. (2006) introduisent une recherche de similarité spatiotemporelle distribuée basée sur la mesure de distance LCSS et proposent deux nouveaux algorithmes offrant de bonnes performances.

Toutes ces méthodes ne sont pas appropriées au calcul de similarités sur le réseau routier puisqu'elles se basent sur la distance euclidienne et non de la distance réelle, c'est à dire par la route, sur le réseau routier. C'est ce dernier point qui a motivé la proposition de Hwang et al. (2005) qui ont été les premiers à proposer une mesure de similarité basée sur la distance spatiotemporelle entre deux trajectoires utilisant la distance par la route. L'algorithme de recherche de trajectoires similaires consiste en deux étapes : l'étape de filtrage basée sur la similarité spatiale sur le réseau routier et l'étape de raffinement pour la recherche de trajectoires similaires à base de distance temporelle. Tiakas et al. (2006) et Chang et al. (2007) utilisent également une distance spatiotemporelle, basée sur le réseau routier, dans leur algorithme de recherche de trajectoires similaires.

Pour les travaux sur le clustering de trajectoires, nous citons les deux suivants :

Gaffney et Smyth (1999) ont soutenu que la représentation de trajectoire à base de vecteurs est inadéquate dans plusieurs cas. Pour surmonter cette lacune ils ont introduit un modèle de mixtures de régression probabiliste et ils ont montré comment l'algorithme EM pourrait être utilisée dans le clustering des trajectoires. Cette approche considère des trajectoires non contraintes. De plus, elle groupe les trajectoires entières et non des sous-trajectoires dans un cluster.

Lee et al. (2007) proposent eux un algorithme appelé TRACLUS qui groupe des sous-trajectoires similaires dans chaque cluster. Il consiste en deux phases : le partitionnement des trajectoires sous forme de segments de droites et puis le groupage de ces derniers selon leurs similarités. Néanmoins, ce travail suppose toujours un mouvement non contraint de trajectoires. De plus, il ne considère ni le temps ni le sens de parcours des trajectoires.

Aucun des travaux précédents ne permet le clustering de sous-trajectoires contraintes par le réseau.

Clustering de trajectoires contraintes par un réseau.

3 Procédure de clustering

Le clustering consiste à former, à partir d'une base de données, des groupes d'objets similaires (Han et Kamber 2006). On entend généralement par procédure de clustering l'ensemble des étapes devant être réalisées depuis la mise à disposition d'un jeu de données jusqu'au traitement des résultats obtenus suite au partitionnement de l'espace des données en clusters. Classiquement, les procédures de clustering comprennent les étapes suivantes dans l'ordre :(i) Représentation des données, (ii) Définition d'un critère de similarité, (iii) Clustering, (iv) Abstraction des données (v) Evaluation de la qualité du clustering.

Nous proposons une méthode de clustering en deux phases. La première permet de grouper les sections de route. Nous parlerons de clustering de sections par abus de langage. La seconde phase effectue véritablement le clustering des trajectoires. Étant donné que ces types de custering portent sur des objets complexes, nous devons préciser pour chaque phase la représentation, la similarité et l'algorithme spécifique de clustering.

3.1 Représentation des données

La représentation du réseau est donnée par l'ensemble de sections de route. Par ailleurs, connaissant l'ensemble des trajectoires, nous calculons une matrice de transitions associée au réseau routier (cf. FIG. 1). Celle-ci fournit des statistiques sur les passages aux carrefours et les mouvements tournants, en reportant le nombre d'objets mobiles qui transitent d'une section à une section connexe. Cette matrice sera notée par la suite *M* et une cellule *M(i,j)* dénote le nombre d'objets mobiles traversant de la section S_i vers la section S_j.

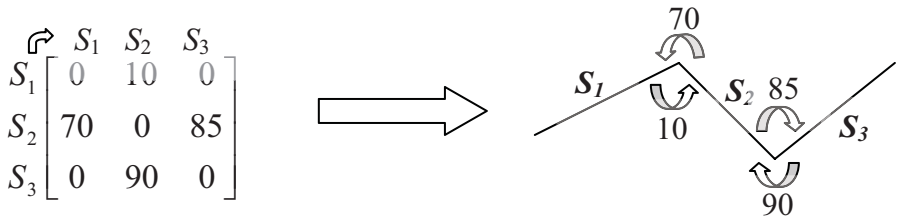


FIG.1 - *Matrice de transitions attribuée à un réseau.*

Nous adoptons une représentation symbolique des trajectoires (Du Mouza, 2005), (Wan et Zeitouni, 2005). Dans cette représentation, une trajectoire d'objets mobiles est une séquence de symboles dont chacune se réfère à une section de route.

$$TR=<S_{i1}, \ldots\ldots\ldots\ldots, S_{in}>$$

L'ordre de ces symboles détermine le sens du déplacement.

3.2 Critères de similarité

La similarité représente la base de l'opération de clustering. Nous définissons la similarité à deux niveaux. Au niveau du réseau, la similarité est calculée entre deux transitions

comme la différence de leurs valeurs de densité. À noter que cette mesure ne concerne que des transitions consécutives.

$$Sim\_route\ (M(i,j),\ M(j,k)) = |M(i,j) - M(j,k)|$$

Au niveau des trajectoires, nous définissons une mesure de similarité entre deux trajectoires dont une est la référence. Cette mesure reflète une ressemblance à un objet et n'est pas symétrique. Elle permet de comparer les trajectoires effectives à une trajectoire fictive type. Pour cela, la similarité est calculée comme le rapport entre la longueur commune entre une trajectoire et la référence d'un côté, et la longueur de la trajectoire référence de l'autre. La longueur d'une trajectoire ou de la partie commune des trajectoires est égale au nombre de sections qu'elle contient.

$$Sim\_traj = Longueur\ (partie\_commune)/longueur\ (traj\_réf\ )$$

Par comparaison aux travaux cités dans la section 2, la similarité y est basée sur la distance euclidienne et/ou de formes (Yanagiswa et al., 2003). Ceci vient du fait qu'ils représentent les trajectoires par leur géométrie et leur forme. Notre travail adopte un critère tout à fait différent car il se place dans le contexte contraint. Il utilise l'information disponible sur la densité du réseau, d'une part et la représentation symbolique des trajectoires permettant de se ramener à des similarités de séquences comme dans Chen et al. (2005), d'autre part.

3.3 Clustering

L'étape du clustering correspond à la phase de regroupement proprement dite et mène à l'obtention d'une partition de la base de données aussi pertinente que possible. C'est à ce niveau, que nous proposons un algorithme de clustering nommé NETSCAN en deux étapes. Dans une première étape, il génère les clusters de sections de route les plus fréquentées formant chacun un chemin dense sur le réseau routier. La seconde étape permet de classifier les trajectoires d'objets mobiles selon ces chemins denses. La figure 2 résume les principales étapes de NETSCAN. Notre algorithme partage les mêmes caractéristiques que l'algorithme DBSCAN et se met en oeuvre en deux étapes :

1. Groupement de sections : Trouver les chemins du réseau les plus denses en termes d'objets mobiles qui y transitent.
2. Groupement des trajectoires : Pour chacun de ces chemins, regrouper la trajectoire qui lui est similaire.

Clustering de trajectoires contraintes par un réseau.

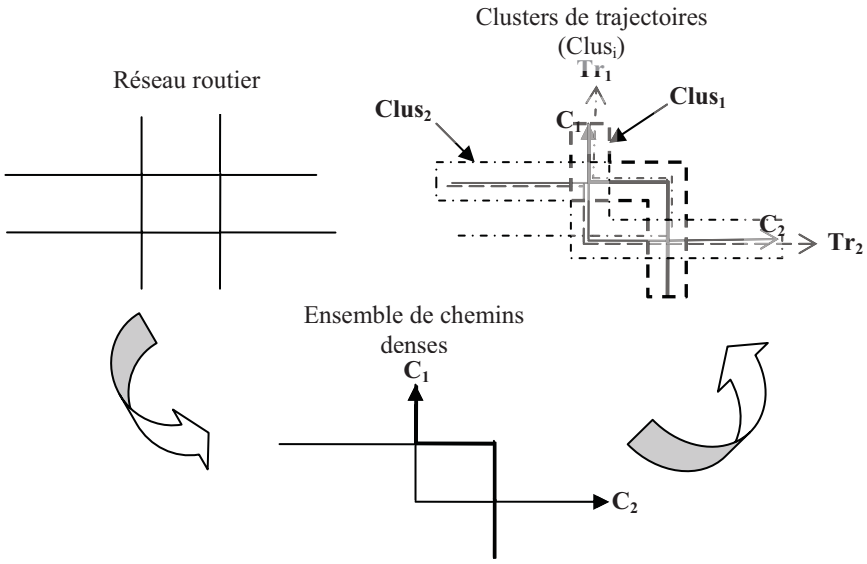


FIG.2 – *Exemple de clustering de trajectoire via l'algorithme NETSCAN*

4 Clustering des sections

L'algorithme proposé s'appelle NETSCAN. Il se compose de deux parties. La première, NETSCAN-PHASE 1 décrite ici, effectue le clustering des sections. Cette première phase s'inspire du principe de densité introduit dans l'algorithme DBSCAN (Ester et al., 1996) en l'appliquant sur des sections de routes. Elle prend en entrée l'ensemble de sections qui constituent le réseau routier, la matrice de transitions entre les sections de routes (cf. section 3.1), un seuil α de densité et un seuil ε de similarité entre les densités des transitions. Lors de cette phase, l'algorithme regroupe les sections où transite le maximum d'objets mobiles en premier (les transitions denses). Il groupe ensuite les transitions connexes dont les densités sont similaires constituant ainsi des chemins denses.

Pour ce faire, le processus commence par la transition ayant la densité maximum. Ensuite, le groupage s'étend aux transitions connexes dans les deux sens pour chercher celles ayant une densité proches à ε près. De cette façon, de proche en proche, on génère un « chemin dense ». Pour assurer la non-réutilisation des transitions figurant dans des chemins denses, on les marque à la première affectation.

L'extension du chemin dense se fait dans les deux sens si les conditions sont remplies, à savoir le non marquage de la transition à rajouter, le respect des seuils de densité α et de similarité ε. Les clusters de sections obtenus correspondent aux chemins les plus denses sur le réseau. La figure 3 décrit la première phase de l'algorithme NETSCAN.

Tout comme les trajectoires, les chemins denses sont représentés sous forme de séquences de sections (cf. section 3.1). Chaque section est donnée par un symbole qui l'identifie.

Algorithme NETSCAN-PHASE 1 /* Découverte d'itinéraires denses*/

Entrée :
- Ensemble de sections S = {S_1, S_2, …, $S_{nbsections}$}
- Matrice de transitions M.
- Seuil ε -- écart maximal de densité entre sections voisines.
- Seuil α -- densité minimum requise dans une transition d'un chemin.

Sortie :
- Ensemble ordonné de chemins O = < C_1, C_2, …, $C_{nbchemins}$ >.

Algorithme :

1. O ← vide -- Initialisation
2. k ← 0
3. Tant qu'il existe des transitions M(i,j) >= α et non marquées
4. k++
5. M(d,f) = max (M [i, j])
6. C_k ← <S_d, S_f> -- on génère un nouveau chemin dense avec cette transition
7. Marquer la transition M(d,f)
8. Tant qu'il existe u tel que M(f,u)>= α et non marquée – extension en avant
9. Sélectionner M(f, f_succ) telle que |M(d,f) – M(f,u)| est minimum
10. Si |M(d,f) – M(f,_succ)| <= ε
11. Insérer queue (C_k, S_{f_succ})
12. Marquer M(f, f_succ)
13. D← f ; f ← f_succ -- étendre le chemin
14. Sinon Sortie de la boucle
15. Tant qu'il existe u tel que M(u,d)>= α et non marquée -- extension en arrière
16. Sélectionner M(d_préd, d) telle que |M(d,f) – M(u,d)| est minimum
17. Si |M(d,f) – M(u,d)| <= ε
18. Insérer tête (C_k, $S_{d\_préd}$)
19. Marquer M(d_préd, d)
20. F← d; d ← d_préd -- étendre le chemin
21. Sinon Sortie de la boucle
22. Ajout du chemin C_k à l'ensemble O
23. Retourner O

FIG.3 – *Algorithme NETSCAN–Phase 1 : clustering des sections en chemins denses*.

5 Clustering des trajectoires

Cette section présente la deuxième partie de l'algorithme NETSCAN correspondant au clustering de trajectoires. Celle-là se base sur le résultat obtenu par la première partie de l'algorithme présentée dans la section précédente. En effet, les chemins denses sont considérés comme des centres naturels de clusters de trajectoires d'objets mobiles.

Clustering de trajectoires contraintes par un réseau.

L'algorithme de clustering de trajectoires consiste à grouper les trajectoires selon leurs similarités avec chacun des chemins denses générées lors de la phase 1 de NETSCAN. La mesure de similarité utilisée est celle définie dans la section 3.2. L'algorithme prend en entrée l'ensemble de chemins denses, l'ensemble des trajectoires et le seuil σ. Il parcourt dans l'ordre les chemins denses et calcule pour chacun sa similarité avec chaque trajectoire d'objets mobiles. Si cette similarité répond au seuil σ, alors la trajectoire est retenue dans le cluster. Plus exactement, la partie commune de la trajectoire avec le chemin dense est ajoutée au cluster. Le nombre de clusters retournés à la fin de l'algorithme est égal au nombre de chemins denses.

Algorithme NETSCAN-PHASE 2

Entrée :

- Ensemble de chemins récupérés de l'algorithme NETSCAN-PHASE 1
  $O = < C_1, C_2, \ldots, C_{nbchemins} >$
- Ensemble de trajectoires d'objets mobiles $TR = < TR_1, TR_2, \ldots, TR_{nbtrajectoires} >$
- Seuil σ -- seuil minimal de similarité

Sortie :

- Ensemble de clusters $Clus = \{Clus_1, Clus_{nbchemins}\}$

Algorithme :

1. Pour (i=0 ; i < nbchemins ; i ++)
2. Pour (j=0 ; j<nbtrajectoires ; j ++)
3. Calculer la séquence $Sc = C_i \cap Tr_j$ -- ensemble des sections communes
4. Calculer la somme des longueurs des sections commune

$$L_c = \sum_{\substack{nbde\ sec\ tions \\ communes}}^{i=0} longeur\ \ (S_c)$$

5. Mesurer la similarité entre C_i et Tr_j en utilisant la formule : $Sim = L_c / longeur\ (C_i)$
6. Si sim >= σ
7. Ajouter Sc à $Clus_i$
8. Ajouter $Clus_i$ à l'ensemble Clus
9. Retourner Clus

FIG.4 – *Algorithme NETSCAN-Phase2 : Clustering des trajectoires*.

6 Conclusion et Perspectives

Cet article se situe dans le cadre de la fouille de données spatio-temporelles. Plus précisément, il se focalise sur le clustering pour l'adapter aux objets mobiles contraints par le réseau routier. Nous avons proposé un algorithme de clustering en deux phases. La première

traite des sections de routes pour obtenir les chemins les plus denses sur le réseau et la seconde traite des trajectoires afin d'obtenir des classes de trajectoires similaires.

Comme futur travail, nous allons implémenter notre algorithme pour évaluer ses performances. Dans sa forme actuelle, l'algorithme de clustering de trajectoires parcourt la base de données autant de fois qu'il n'y a de chemins denses. Une solution plus optimale devra être mise en œuvre afin de réduire la complexité algorithmique, notamment en terme d'entrées / sorties. L'utilisation d'une solution par index spatial comme le TTR-tree proposé dans Wan et Zeitouni (2006) permettrait un gain significatif de performances.

Nous étudierons ensuite la prise en compte du temps, ainsi que d'autres mesures rattachées à la localisation des trajectoires. Pour cela, nous garderons et nous étendrons notre représentation symbolique par référence aux sections du réseau routier. Cela implique l'extension de la matrice de transition par la dimension temporelle ou de mesure. La similarité devra également être étendue et les algorithmes adaptés et optimisés.

Références

Ankerst M., M. M. Breunig, H.-P. Kriegel et J. Sander (1999) OPTICS: *Ordering Points to Identify the Clustering Structure*, In Proc. 1999 ACM SIGMOD Int'l Conf. on Management of Data, Philadelphia, Pennsylvania, pp. 46-60.

Chang J-W., R. Bista, Y-C. Kim et Y-K Kim (2007) *Spatio-temporal Similarity Measure Algorithm for Moving Objects on Spatial Networks*. ICCSA 2007, pp.1165-1178.

Chen L., M.T. Ozsu et V. Oria (2005) *Robust and Fast Similarity Search for Moving Object Trajectories*. In: ACM SIGMOD, pp. 491-502. ACM Press, New York.

Du Mouza C. (2005) *Patterns de mobilité*. Thèse, Chapitre 3 : Patterns de mobilité, p. 51-66.

Ester M., H.-P. Kriegel, J. Sander et X. Xu (1996) *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, In Proc. 2nd Int'l Conf. on Knowledge Discovery and Data Mining, Portland, Oregon, pp. 226-231.

Gaffney S. et P. Smyth, (1999) *Trajectory Clustering with Mixtures of Regression Models*, In Proc. 5th ACM SIGMOD Int'l Conf. on knowledge Discovery and Data Mining, San Diego, California, pp. 63-72.

Hadjieleftheriou M., G. Kollios, P. Bakalov, V. Trotras (2005) *Complex Spatio-Temporal Pattern Queries*. In Proc. of the 31st VLDB Conference.

Han J. et M. Kamber (2006) *Data Mining: Concepts and Techniques*, 2nd ed., Morgan Kaufmann.

Hwang J-R., H-Y. Kang et K-J. Li (2005) *Spatio-temporal Analysis Between Trajectories on Road Networks*. ER Workshops 2005, LNCS 3770, pp. 280-289.

Lee J-G, J. Han et K-Y. Whang (2007) *Trajectory Clustering: A Partition-and-Group Framework*. In Proc.SIGMOD'07, Beijing, China.

Lin B., J. Su (2005) *Shapes Based Trajectory Queries for Moving Objects*. GIS, pp. 21-30.

Lloyd S. (1982) *Least Squares Quantization in PCM*, IEEE Trans. on Information Theory, 28(2): 129-137.

Sakurai Y., M. Yoshikawa et C. Faloutsos (2005) FTW: *Fast Similarity Search Under the Time Warping Distance*. In: PODS, pp. 326-337.

Shim C-B et J-W Chang (2003) *Similar Sub-Trajectory Retrieval for Moving Objects in Spatiotemporal Databases.* In: Proc. of the 7th EECADIS, pp.308-322.

Tiakas E., A. N. Papadopoulos, A. Nanopoulos et Y. Manolopoulos (2006) *Trajectory Similarity Search in Spatial Networks.* In : Proc. of the 10th IDEAS, pp. 185-192.

Vlachos M., D. Gunopulos et G. Kollios (2002) *Robust Similarity Measures of Mobile Object Trajectories*. In: Proc. of the 13 th Intl. Workshop on DEXA, IEEE Computer Society Press, Los Alamitos pp. 721-728.

Vlachos M., G. Kollios et D. Gunopulos (2002) *Discovering Similar Multidimensional Trajectories*. In: Proc. Of the 18th ICDE. IEEE Computer Society Press, Los Alamitos pp. 673-684.

Wan T., K. Zeitouni (2005) *Modélisation d'objets mobiles dans un entrepôt de données*. EGC 2005: pp. 343-348.

Wan T., K. Zeitouni (2006) *Représentation et indexation d'objets mobiles contraints par le réseau dans un entrepôt de données*, 2ème Conférence Entrepôts de données et Analyse en ligne, EDA 2006, Revue Nouvelles Technologies de l'Information (RNTI), Cépaduès Edition, 2006, pp. 139-154.

Yanagisawa Y., J. Akahani, T. Satoch (2003) *Shape-Based Similarity Query for Trajectory of Mobile Objects*. In : Proc. Of the 4th Intl. Conf. On MDM, pp. 63-77.

Zeinalipour-Yazti D., S. Song Lin, D. Gunopulos (2006) *Distributed Spatio-Temporal Similarity Search.* CIKM, pp. 14-23.

Zhang T., R. Ramakrishnan, et M. Livny (1996) BIRCH: *An Efficient Data Clustering Method for Very Large Databases*. In Proc. ACM SIGMOD Int'l Conf. on Management of Data, Montreal, Canada, pp. 103-114.

Summary

In this paper, we proposed a new clustering density based method into moving object trajectories database. The proposed algorithm (NETSCAN) is two steps: the first one computes dense paths in the network while the second step clusters the moving object trajectories with the dense paths using an adapted similarity measure. Our work assumes time is relaxed constraint. The advantage of this framework is to discover groups of common sub-trajectories from a trajectory database.

Découverte de relations par croisement d'analyses

Lobna Karoui*, Marie-Aude Aufaure*

*Supelec, Plateau de Moulon, Gif-sur-Yvette
{Lobna.Karoui, Marie-Aude.Aufaure}@supelec.fr
http://www.supelec.fr/ecole/si/pages_perso/karoui.html

Résumé. L'extraction de relations est une tâche difficile. Dans cet article, nous nous intéressons à la découverte des relations entre les concepts ontologiques afin de construire une ontologie de domaine. Nos objectifs sont d'extraire des relations de différents types à partir des analyses de textes et des relations existantes dans la hiérarchie de concepts, d'étiqueter certaines relations et de pouvoir les valider automatiquement et d'en déduire de nouvelles relations. Pour cela, nous proposons un algorithme contextuel de découverte de relations entre concepts. Il est guidé par une analyse centrée autour du verbe, des analyses lexicales, syntaxiques et statistiques. Il est basé sur la structure du document pendant l'analyse statistique, une modélisation contextuelle riche qui renforce la sélection des termes concurrents, une utilisation des relations existantes dans la hiérarchie de concepts et une intersection entre les relations. Notre algorithme suggère des évaluations automatiques des relations à l'expert du domaine. Cette validation peut être approuvée ou rejetée par l'expert du domaine. Dans les deux cas, elle l'assiste et facilite sa tâche d'évaluation.

1 Introduction

Face à de grandes quantités de documents web, nous souhaitons extraire leurs connaissances pour le web sémantique. Généralement, ces connaissances sont représentées par des concepts et des relations qui les relient. Dans un travail précédent (karoui et al., 2007), nous avons défini une approche de découverte de concepts basée sur le contexte. Ces concepts sont utilisés pour construire une ontologie. Dans cette seconde partie de notre recherche, nous nous focalisons sur l'extraction des relations. Dans l'état de l'art, l'extraction des relations a été faite soit par une approche statistique, une approche linguistique ou une approche hybride. De plus, l'intérêt a été toujours porté sur un voire deux types de relations. A contrario, notre objectif est d'extraire des relations de différents types en combinant des analyses de textes et en considérant les caractéristiques des mots. Dans cet article, à partir de l'état de l'art, nous expliquons quels sont les problèmes des différentes approches d'extraction des relations. Ensuite, nous proposons les idées fondamentales de notre approche. Cette dernière est basée sur trois analyses à savoir une analyse lexicale, une analyse syntaxique et une analyse statistique qui prend en compte les caractéristiques pragmatiques et stylistiques d'un document. En appliquant une ou deux analyses, il y aura obligatoirement des relations oubliées. Afin d'éviter ceci, nous définissons un algorithme contextuel de découverte de relations qui combine différentes analyses pour définir des processus complémentaires qui assurent l'extraction de relations variées et pertinentes. Notre algorithme établit des opérations de croisements entre analyses afin de pouvoir valider certaines relations. Les relations valides,

comme celles non valides, seront présentées à l'expert du domaine pendant la phase d'évaluation mais en les séparant. Il prend en compte la structure des documents pendant l'analyse statistique (pour calculer la fréquence des mots), une modélisation contextuelle riche qui renforce la sélection des cooccurrents des termes, une analyse lexicale, une utilisation des relations existantes dans la hiérarchie de concepts (karoui et al., 2006) et une intersection entre les différentes relations extraites grâce à des analyses variées afin de faciliter l'évaluation faite par l'expert du domaine. Nos premières expérimentations ont généré des résultats satisfaisants.

2 Etat de l'art

L'extraction des relations peut servir lors de la constitution d'ontologies (Maedche, 2002), (Reinberger et al., 2004), l'enrichissement d'ontologies (Schutz et Buitelaar, 2005), (Magnini et al., 2005), etc. On peut, également, s'intéresser à extraire un seul type de relation comme les relations de causalités (Nazarenko, 1994), les relations d'hypéronimie (Hearst, 1992), les relations complexes comme celles n-aire qui impliquent n entités n'appartenant pas à la même phrase (McDonald et al., 2005), les relations taxonomiques et non taxonomiques (Maedche et Staab, 2000), etc. La majorité des approches d'extraction des relations ont exploité les patrons syntaxiques. Ces patrons sont composés de deux expressions nominales et une expression verbale dont l'ensemble constitue ce qu'on appelle SVO (subject-verb-object). Dans ce contexte, plusieurs stratégies ont été utilisées comme le fait d'identifier la similarité entre les termes et les verbes dans un patron en tenant compte du texte, d'appliquer un ensemble limité de patrons pour un processus d'apprentissage, d'extraire de nouveaux patrons en utilisant WordNet et en combinant une méthode statistique et syntaxique. Par exemple, l'approche proposée par (Yangarber et al., 2000) extrait les relations en généralisant un ensemble de patrons. Les entités dans les patrons sont groupées en paires afin de fournir des fréquences permettant d'appliquer des méthodes statistiques (chaque paire représente un patron). Ensuite, ces paires sont utilisées pour enrichir les mots oubliés dans les relations. Les patrons peuvent être aussi évalués en utilisant des mesures de probabilités. Sevenson et Greenwood (2005) ont utilisé un ensemble pertinent de patrons primitifs afin d'apprendre un autre ensemble de patrons à partir de textes et en appliquant la mesure de similarité entre les mots de wordnet. Certains patrons SVO sont produits par des analyses syntaxiques. Ceux similaires aux patrons existants seront ajoutés à l'ensemble des patrons de travail. Soderland (1999) propose une approche supervisée qui apprend des patrons d'extraction de relations à partir de phrases appartenant à des textes non annotés et grâce à un analyseur syntaxique. Le système présente à l'utilisateur un ensemble de nouvelles règles à partir d'un ensemble d'apprentissage. En combinant une méthode statistique et linguistique, Gamalho et al (2002) ont classifié les dépendances syntaxiques similaires afin d'extraire les relations sémantiques qui sont structurées hiérarchiquement par la suite. Maedche et Staab (2000) ont proposé une approche pour l'extraction des relations non taxonomiques. Ils ont utilisé des règles d'associations ainsi qu'une taxonomie existante. Pour cela, un premier algorithme trouve les paires d'entités lexicales ainsi que leurs pondérations dans le texte. Un second algorithme détermine les mesures de confiance et de support pour les relations entre les concepts. Seules les relations conceptuelles sont préservées. De même que Maedche, Schutz et Buitelaar (2005) utilisent des traitements statistiques et linguistiques. Dans la partie linguistique, l'analyse de la structure et la reconnaissance des concepts nommés est réalisée.

Dans la partie statistique, les auteurs calculent trois formules liées à l'ordre de pertinence, la référence des noms et verbes pertinents aux paires de prédicats et les scores de cooccurrences. Dans les travaux précédemment présentés, nous avons présenté ceux qui se basent sur la linguistique tout en utilisant parfois des méthodes statistiques. En tant qu'approche statistique, Reinberger et Spyns (2004) emploient principalement des méthodes statistiques basées sur la fréquence d'information dans les dépendances linguistiques afin de pouvoir extraire des relations entre les entités d'un corpus biomédical sans être amené à les nommer (relations). D'autres approches sont symboliques (Claveau et al., 2003) utilisent la programmation logique inductive. Cette méthode utilise les couples qualia extraits afin de produire des règles en généralisant des exemples.

En général, ces approches n'exploitent pas les relations existantes, précisément quand il s'agit d'enrichir ou de construire une ontologie. En travaillant dans un domaine spécifique, elles permettent de définir une méthode riche mais qui n'est pas systématiquement utilisable pour d'autres domaines. Ceci est du à la difficulté d'extraction des patrons et de leur adaptation à d'autres domaines. Nous avons aussi remarqué que la majorité des approches néglige la tâche d'étiquetage (Reinberger et al., 2004) et nécessite une phase manuelle importante (Sabou, 2004).

Les approches se basant sur les patrons limitent le contexte à une phrase. Cependant, la relation entre les mots ne se limite pas à une phrase mais peut la dépasser vers des paragraphes, etc. Concernant les approches centrées autour du verbe, elles supposent que toute relation est exprimée par un verbe ce qui est réducteur pour ce type d'approche (une relation n'est pas obligatoirement exprimée par un verbe). Nous signalons aussi l'intérêt fréquent à extraire un voire deux types de relations ou plutôt des relations générales sans pour autant définir leurs types. Dans la majorité des approches étudiées, c'est l'expert du domaine qui est amené à évaluer les relations extraites. A notre connaissance, aucune tentative d'automatisation de l'évaluation des relations n'a été réalisée.

Dans notre approche, nous définissons une méthode qui extrait différents types de relations, dépasse la phrase lors de la définition de contextes pour l'extraction des relations, combine les approches centrées autour du verbe et les autres afin d'éviter les oublis possibles de relations pertinentes et propose des alternatives d'évaluation des résultats afin d'aider l'expert du domaine.

3 Les idées fondamentales de notre approche

Dans les travaux précédents (karoui et al., 2007), nous avons défini un algorithme d'extraction d'une hiérarchie de concepts. Cette dernière contient des relations entre les classes de mots qui ont été évaluées auparavant. Par conséquent, extraire ces relations ne sera plus utile vu qu'elles existent déjà. C'est la raison pour laquelle, notre algorithme d'extraction de relations veillera à ce qu'elles soient effacées (les relations existantes).

3.1 Analyse lexicale

Cette analyse exploite des patrons pouvant générer des associations pertinentes de mots. Elle permet de découvrir à la fois des relations sémantiques implicites et explicites. Par exemple, en utilisant la relation de coordination avec la conjonction « et » nous allons découvrir une relation explicite qui peut exister dans l'analyse syntaxique. Par contre, en utili-

sant l'effet de la ponctuation, nous sommes capables de découvrir des relations implicites. Dans l'exemple suivant « hébergement : hôtel, gîtes, etc. », hôtel et gîte représentent une première relation. En considérant le signe de ponctuation ' :', nous constatons que le mot « hébergement » se trouve en relation avec le couple 'hôtel, gîte'. Le changement du signe de ponctuation peut révéler des changements au niveau de la compréhension du texte ainsi que son exploitation.

3.2 Analyse syntaxique

L'analyse centrée autour du verbe. Pour cette analyse, il s'agit de sélectionner les verbes puis leurs arguments dans les syntagmes verbaux (SV). Puis, nous extrayons deux types de relations : les relations explicites en cherchant les mots associés des SV dans les syntagmes nominaux SN et les relations implicites, en cherchant les mots associés des SV dans des contextes plus larges que la phrase. Les relations implicites sont le résultat d'une analyse syntaxico-contextuelle. En cherchant les relations autour du verbe mais en les sélectionnant à l'avance, nous assurons une évaluation progressive des relations produites. Autrement dit, en se focalisant sur les verbes pertinents (les plus fréquents dans le domaine ou désignés par l'expert du domaine) nous limitons le nombre d'associations pouvant être générés et facilitons l'évaluation du résultat final par l'expert. Par ailleurs, nous contribuons à la découverte de relations implicites pouvant être induites d'autres types d'analyses.
L'analyse globalement syntaxique. Cette analyse exploite tous les types de relations existant entre les unités textuelles sans pour autant se limiter à celles centrées autour du verbe. De ce fait, nous allons trouver des relations de coordination, d'objet, sujet, etc. Cette analyse extrait des relations explicites. Nos patrons (lexicaux ou syntaxiques) sont réutilisables et indépendant du domaine mais pas de la langue.

3.3 Analyse statistique

Selon l'hypothèse de Harris, une analyse distributionnelle des propriétés contextuelles des mots fait apparaître des classes de concepts et des relations entre les classes. Nous allons suivre son hypothèse, pour découvrir les relations de notre corpus. Dans l'analyse statistique, nous avons utilisé trois types de contextes à savoir le contexte structurel, le contexte fenêtre proximité et le contexte paragraphe. Pour le contexte structurel, nous allons exploiter les traitements qui ont servi à l'extraction des concepts en tirant profit de l'analyse contextuelle représentée grâce à notre hiérarchie contextuelle et la position du mot dans ce modèle. Le contexte de fenêtre est utilisé avec un degré de proximité pour bien définir les mots reliés. Par exemple, si nous considérons une fenêtre de 4 mots et les mots suivants : ' moyens de transport comme la voiture', pour le mot 'voiture', le fait que 'moyens' et 'transport' sont les plus proches à 'voiture' sera pris en compte dans le calcul de cooccurrences. Nous considérons, également, tous les mots à l'intérieur d'un paragraphe comme des voisins et calculons leurs cooccurrences. Cette définition de contexte 'paragraphe' est aussi importante puisqu'elle permet de produire des informations complémentaires avec les contextes de proximité et structurels. C'est pourquoi, nous utilisons tous ces contextes dans notre approche. Dans tous les cas, ce que nous utilisons est la distribution des mots dans leurs contextes ainsi que les relations entre ces derniers afin de calculer l'indice d'équivalence qui exprime le degré d'association des mots selon leurs distributions. Cette analyse nous assure une découverte des relations sémantiques implicites.

La définition d'un contexte structurel. L'existence d'une relation structurelle entre les éléments HTML peut révéler une relation sémantique implicite entre les termes associés, par exemple, les balises <h1> <p> ; <caption> <td> (titre d'un tableau cellule d'un tableau) ; etc. Nous distinguons deux types de lien structurel : un lien physique qui dépend de la structure du document HTML (entre la balise <h1> et la balise <p> associée) et un lien logique qui n'est pas visuel puisque les éléments ne sont pas nécessairement consécutifs (entre <TITLE_URL> (balise définie qui désigne le tire d'un lien hypertexte) et les titres du document référencé par exemple). Pour caractériser les liens entre les balises, nous avons défini deux notions : la « hiérarchie contextuelle » (H.C.) basée sur les balises HTML et la « cooccurrence par liaison ».

Une hiérarchie contextuelle est un modèle sous forme d'une hiérarchie de balises. Cette modélisation contextuelle des connaissances illustre les relations possibles dans les documents HTML, et entre eux, en considérant chaque balise comme un contexte. La cooccurrence est définie par le fait d'avoir deux mots dans le même contexte (paragraphe, texte, etc.). Dans notre étude, le contexte est variable et il est déduit de notre modèle contextuel (H.C). En respectant la structure de H.C, nous établissons des liaisons entre les termes si :

Les termes sont encadrés par la même balise bloc (TAB. 1 : Exemple 1). Dans ce cas, on parle de cooccurrence par voisinage et le contexte est fixé à la balise elle-même (<H1>).

Les termes sont encadrés par des balises qui à leurs tours sont reliées par un lien physique ou logique schématisé dans la hiérarchie contextuelle. Dans ce deuxième cas (TAB. 1 : Exemple 2), nous parlons de cooccurrence par liaison (balises non consécutives) et le contexte est l'association des deux balises (<title> + <h1>).

Exemple 1	Exemple 2
<H1> événement maritime </H1>	<TITLE> Catégories de logements et d'établissements d'hébergement </TITLE> <KEYWORDS> *** </KEYWORDS> <HYPERLINK> *** <TITLE_URL> *** <H1> Résidences de tourisme </H1> <P> un établissement touristique ayant certaines caractéristiques communes avec un hôtel…… </P>

TAB. 1 – Exemples de contextes d'utilisation

Nous présentons plus de détails concernant cette modélisation contextuelle dans (karoui et al., 2007). L'application du contexte générique en relation avec la structure html et les liens sémantiques existants entre les balises permet de représenter l'adaptabilité d'un terme dans le corpus et modélise un contexte dynamique. Nous avons utilisé plusieurs analyses afin de pouvoir s'adapter non seulement aux documents HTML (mal structuré) mais aussi à d'autres formats de documents (XML, Word, etc.) susceptible d'être utiliser dans nos prochaines expérimentations.

4 La découverte des relations

4.1 La notion de contexte

Le contexte est un ensemble de circonstances (situations) qui entourent l'objet d'étude et reflète son environnement concret. Il fournit un bagage d'information actif pour l'activité d'apprentissage et un support approprié pour l'interprétation sémantique.

Dans cette étude, l'objet d'étude est le mot, l'activité d'apprentissage est l'extraction de relations, l'interprétation sémantique est l'évaluation et l'étiquetage des relations extraites et

la définition de contexte est composée de l'association de différentes unités contextuelles qui appartiennent à différents degrés de granularité avec un degré de proximité intégré.

Pour l'extraction des concepts, notre objectif est de collecter les mots qui définissent le terme étudié en raffinant son contexte. Pour l'extraction des relations, nous souhaitons trouver les mots qui sont reliés au mot étudié. Donc, notre objectif est différent puisque nous cherchons des contextes qui contiennent ces mots reliés. Ainsi, la définition de contexte est différente. Pour l'extraction des relations, nous ne limitons pas le contexte. Au contraire, nous trouvons tous ceux capables de nous produire les mots relationnels. En utilisant les différents contextes définis, nous les catégorisons en quatre types: le contexte structurel, le contexte linguistique (centré autour du verbe, globalement syntaxique et lexical), le contexte documentaire (paragraphe) et le contexte fenêtre (avec un degré de proximité). Notre approche utilise toutes ces analyses afin d'extraire de nouvelles relations (en plus de celles existantes dans la hiérarchie de concepts) et de les valider automatiquement.

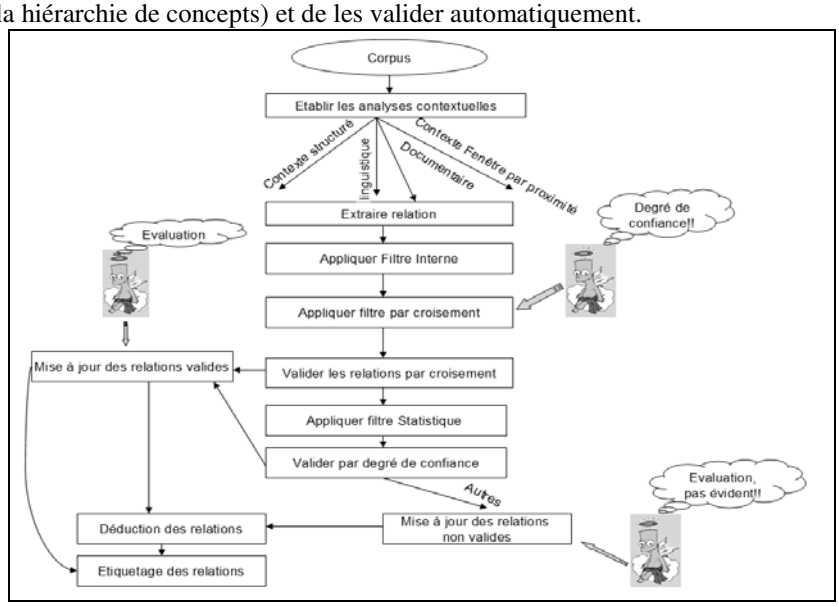


Figure.1. Approche de découverte des relations

4.2 L'algorithme contextuel de découverte des relations

Les détails de notre approche sont schématisés dans la Figure 1 et explicitement présentés dans l'algorithme (Figure 2).

0: Algorithme Découverte_Relation (In: Corpus, classes de mots, DC; Out: VR, NVR)
1: Appliquer l'analyse statistique basée sur le contexte structuré {/*Etape 0*/}
2: Appliquer l'analyse statistique basée sur le contexte fenêtre par proximité
3: Appliquer l'analyse statistique basée sur le contexte paragraphe
4: Appliquer l'analyse linguistique basée sur le contexte centré autour du verbe
5: Appliquer l'analyse linguistique basée sur le contexte globalement syntaxique
6: Appliquer l'analyse lexicale
7: Rassembler les relations extraites

```
8: Appliquer un filtre interne {/*Etape 1*/}
9: Rassembler les nouvelles relations

11: Appliquer le croisement {relations structurées, relations paragraphes} {/*Etape 2*/}
13: Appliquer le croisement {relations fenêtre par proximité, relations lexicales}
15: Sélectionner les relations valides retenues par le croisement

16: Pour toutes les relations non valides Faire {/*Etape 3*/}
17: Appliquer un filtre statistique
18: Si NO > 5% et FN > 0.0025 Alors
19: Rassembler les relations sélectionnées (SR)

20: SI DC >=50% Alors {/*Etape 4*/}
21: Rassembler les relations valides (VR)
22: Sinon si 25%<=DC <=50% Alors
23: Appliquer un filtre statistique fort
24: Si NO > 10% et FN > 0.005 Alors
25: Appliquer les relations valides (VR)
28: Sinon
29: Rassembler les relations non valides (NVR)
30: Fin Si
31: Sinon
32: Rassembler les relations non valides (NVR)
33: Fin SI
34: Sinon
35: Rassembler les relations non valides (NVR)
36: Fin si
37: Fin Pour

38: Pour VR et NVR Faire {/*Etape 5*/}
39: Appliquer la déduction des relations
40: Etiqueter les relations
41: Fin pour
42: FIN
```

Figure.2. Algorithme contextuel de découverte des relations

Notre algorithme contextuel de découverte de relations applique différents types d'analyses pour extraire et évaluer les relations. Il dépend de certains paramètres comme le degré de confiance (DC), NO est le pourcentage d'occurrences de mots dans le corpus (NO) et FN est la fréquence normalisée des mots dans le corpus (FN). Ces paramètres sont utilisés lors du filtre statistique ainsi que la validation. Le DC doit être défini par l'utilisateur vu qu'il explique sa confiance en l'application. Par contre, NO et FN peuvent être définis soit par l'expert du domaine, soit par le système en les déduisant de la valeur de DC ou par défaut (valeur définie par le concepteur du système). Dans le cas où le système est utilisé pour calculer les valeurs de NO et FN, si la valeur de DC est supérieure à 50% leurs valeurs (par défaut) seront maintenues, sinon elles seront multipliées par deux. Notre algorithme catégorise quatre types de relations extraites : valides, invalides, déduites et étiquetées. Une relation valide

est celle qui est récupérée après une opération de croisement entre analyses. Une relation invalide est celle qui n'a pas été retrouvé dans deux analyses.

Notre algorithme est composé de cinq étapes :

Etape0 permet d'appliquer les différentes analyses pour extraire les relations.

Etape1 applique un filtre interne pour éliminer les relations qui représentent les liaisons des mots à l'intérieur des classes validées. Ces relations ne sont plus intéressantes puisqu'elles existent dans la hiérarchie de concepts. A la fin de cette étape, nous obtenons un nouvel ensemble de relations à valider.

Etape2 applique un filtre par croisement des relations résultantes des différentes analyses. Nous proposons deux types de *croisements complets* (qui nécessitent que la relation existe dans les deux analyses pour qu'elle soit retenue) pour la première étape de validation :

- Un croisement au sein de l'analyse statistique. Ce croisement est fait entre les relations structurées et les relations paragraphes vu qu'une structure telle que définie dans notre démarche (contexte structurel) n'est pas systématiquement incluse dans un paragraphe. D'où l'intérêt de recueillir ces relations qui se trouvent dans les deux résultats de nos contextes de même nature.

- Un croisement hybride réservé pour les relations provenant de l'analyse fenêtre par proximité et celles des analyses syntaxiques et lexicales. Ce type de croisement est plus raffiné que le premier. Ce choix est du au nombre immense de relations produites par le contexte fenêtre que nous voulons exploiter en utilisant la force des analyses linguistiques.

Nous pouvons aussi faire des croisements entre d'autres analyses. Pour nos premières expérimentations, nous avons commencé par tester ces types de croisements. Les relations générées par ces croisements (apparues dans les deux contextes) seront validées automatiquement. Pour les autres, ils formeront l'ensemble des relations invalides et serviront dans la suite de l'algorithme.

Etape3 prend en compte l'ensemble des relations invalides et applique un filtre statistique. Ce dernier est fait en définissant la valeur de deux paramètres à savoir le nombre d'occurrences NO et la fréquence normalisée FN. Ces deux paramètres peuvent être définit par défaut suite à une étude empirique, par l'expert du domaine ou par le système. L'ensemble des relations dont les valeurs de NO et FN dépassent celles fixées forment l'ensemble des relations sélectionnées (RS).

Etape4 prend en entrée les relations sélectionnées pour établir l'opération de validation par degré de confiance. Pour cela trois cas se présentent : Si le degré de confiance est supérieur à 50%, dans ce cas les relations sélectionnées seront valides ; Si le degré de confiance est compris en 25% et 50%, l'algorithme renforce le niveau d'exigence en multipliant les valeurs de NO et FN par deux. Les relations qui sont retenues seront validées ; Si le degré de confiance est inférieur à 25%, les relations seront tous invalides.

Dans tous les cas, toutes les relations produites par notre algorithme seront présentées à l'expert du domaine mais avec des indications différentes pour démarquer celles valides de celles invalides.

Etape5 permet de déduire à partir des relations valides et invalides de nouvelles relations et essaye d'étiqueter à la fois les relations extraites et déduites. La déduction de nouvelles relations ou de relations plus riches se fait en croisant les relations globalement syntaxiques (riches vu qu'elles dépendent de règles syntaxiques) avec d'une part celles fenêtre par proximité et d'autre part celles paragraphe. Ce type de *croisement appelé partiel* est différent puisqu'il n'exige pas le fait que la relation soit présente dans les deux analyses mais plutôt qu'il y'ait un élément commun entre les deux relations appartenant aux deux analyses. C'est

ainsi que les relations déduites seront plus riches et plus intéressantes. La déduction n'est pas restreinte aux relations valides vu que celles qui ne le sont pas n'impliquent pas forcément une invalidité absolue. Donc en les utilisant dans cette partie, nous pouvons parfois en valider certaines par ce type de croisement.

L'étiquetage est la tâche qui désigne à une relation un nom. , Il est fait par le biais des relations produites par l'analyse centrée autour du verbe et appliquée aux relations valides et déduites. Cela se produit en essayant de trouver le verbe qui relie deux termes ou deux groupes de termes déjà reliés dans une relation produite par l'analyse fenêtre, structuré ou paragraphe. Le verbe provient de l'analyse centrée autour du verbe.

Notre algorithme permet non seulement d'extraire des relations mais aussi de les valider automatiquement, d'en déduire d'autres relations et de les étiqueter. Toutes ces tâches sont réalisées par les cinq analyses ainsi que leurs aspects complémentaires.

5 Expérimentations

Notre approche de découverte de relations s'est basée sur six analyses que nous avons appliquées à notre corpus de documents HTML. Ce corpus est formé de 565 documents en langue française relatifs au domaine du tourisme. Ainsi, nous avons pu extraire un nombre important de relations : relations centrées autour du verbe (2251) ; relations globalement syntaxiques (34439) ; relations lexicales (5793) ; relations paragraphe (72476) ; relations structurelles (16966) ; relations fenêtres (206010). Toutes les relations qui relient deux mots ou un ensemble de mots appartenant à une même classe déjà formée dans le travail précédent (karoui et al., 2007) ont été supprimées lors du filtre interne (Figure.2) vu qu'elles existent déjà et ne nous apportent aucune information supplémentaire. Concernant les relations restantes, nous avons choisi d'établir deux types de croisements à savoir un croisement entre les relations structurelles et paragraphes, et un second entre les relations fenêtres et lexicales. Le premier croisement nous a permis de retenir 372 relations (Hôtellerie/ hébergement, Réservation/hébergement, Camping/dormir). Quant au second croisement, nous avons pu avoir 268 relations (Catholicisme/christianisme, Ethnographie/paléontologie), sachant que dans les deux croisements nous avons supprimé certaines relations contenant des noms propres afin de minimiser le bruit. D'après notre algorithme, ces relations seront considérées automatiquement valides vu qu'elles se sont répétées dans deux analyses différentes.

En prévoyant un degré de confiance moyen inférieur à 50%, lors du filtre statistique, nous avons augmenté le niveau d'exigence pour éviter de valider des relations inutiles. Par exemple, pour les relations globalement syntaxiques, il s'agit de préserver les relations qui ont un nombre d'occurrences supérieur à 4% du total des relations extraites auparavant mais non validées encore. Ce taux d'exigence bien qu'étant renforcé reste relativement raisonnable. Donc, la relation globalement syntaxique que nous remarquons au moins 1378 fois pourra être validée automatiquement. Pour les relations statistiques de base (structurelle, fenêtre et paragraphe), il s'agit de tolérer les relations ayant une fréquence normalisée supérieure à 0.005 (double de 0.0025 vu le degré de confiance faible définit par l'utilisateur). Ces choix de départ que ce soit la fréquence normalisée ou le nombre d'occurrences sont le résultat d'une étude empirique mais pourront être changés par l'utilisateur ou l'expert du domaine. Après cette étape de filtre statistique, lors de nos expérimentations, nous n'avons pas pu retenir des relations valides sur celles lexicales, globalement syntaxique et centrée autour du verbe vu que la relation la plus récurrente ne dépasse pas les 20 fois ; ce qui est largement

loin de nos critères définis. Par contre, pour les relations fenêtres (Activité/sport, Nautique/sport, Patrimoine/histoire, Plonger/sport) nous avons obtenu 24818 relations validées selon notre algorithme et 15257 relations pour celles structurelles (Casino/divertissement, Festival/musique, Vigne/vignoble). Pour les relations paragraphe, le résultat des validations a été négatif. Les relations qui n'ont pas été validées tout au long de notre démarche seront les relations invalides. Celles-ci seront présentées à l'expert en cas de besoin.

Jusqu'à présent, nous avons utilisé toutes les relations au niveau des opérations de croisement complet (car c'est un croisement qui nécessite que la relation existe dans les deux analyses pour qu'elle soit retenue) sauf celles provenant d'une analyse globalement syntaxique et d'une analyse centrée autour du verbe. Les relations globalement syntaxiques ont la caractéristique d'être extrêmement riches vu qu'elles se basent sur des règles grammaticales. C'est pourquoi, elles n'ont pas été choisies pour la validation croisée. Vu leur importance sémantique, elles serviront au niveau de l'opération de déduction automatique de nouvelles relations. Ces relations seront projetées sur les relations statistiques de tous genres afin de pouvoir déduire des relations plus riches (portant sur plus que deux mots). Ainsi, si nous trouvons un mot en commun entre deux relations provenant de deux analyses différentes, nous formons la même relation qui groupera l'ensemble des entités sémantiques. Ce croisement, appelé partiel, implique la totalité des relations (même invalides) de type fenêtre (Patrimoine religieux + patrimoine architectural + patrimoine mondial de unesco + patrimoine naturel et culturel) et paragraphe (Hébergement en camping+ hébergement sur place + hébergement proximité+ hébergement en refuge + hébergement en chalet pittoresque + hébergement en auberge ou campement+ hébergement sous un tipi + hébergement en igloo).

Pour les relations centrées autour du verbe, elles permettront de nommer les autres relations. En les utilisant, nous avons projeté ces relations sur celles structurelles (Randonneur *animer* randonnée), fenêtre (Planche *avoir* voile) et paragraphe (*découvrir* paysage) afin de pouvoir trouver des mots communs. Dans ce cas, nous attribuons automatiquement le verbe de la relation centrée autour du verbe à l'autre type de relation.

Grâce à notre algorithme contextuel de découverte des relations, nous avons pu non seulement extraire les relations mais aussi se servir de la variété des analyses pour en valider certaines, en déduire de nouvelles et en nommer celles extraites et déduites. Cela a été fait d'une façon automatique mais pourra faire appel à l'expert du domaine lors de l'évaluation finale. Jusqu'à présent, l'algorithme a été implanté et certains testes ont été effectués. En perspective, nous souhaiterons présenter l'ensemble des résultats à l'évaluation manuelle pour la comparer à celle automatique (pour les relations valides), multiplier les testes afin de mieux définir les valeurs de FN et NO et appliquer des processus d'apprentissage.

6 Conclusion

L'extraction des relations reste encore une tâche difficile. Dans cet article, nous avons défini les idées fondamentales de notre approche qui combine une méthode centrée autour du verbe avec une analyse lexicale, syntaxique et statistique. Notre méthode, utilisée pour la langue française, peut être appliquée pour la langue anglaise mais avec une simple modification qui concerne le choix de l'analyseur syntaxique et les patrons définis par l'analyse lexicale. De plus, notre approche combine différentes analyses afin d'extraire différents types de relations. Elle définit deux types de croisements entre analyses. Un premier croisement intégral qui nécessite que les relations existent en intégralité en commun dans les deux analyses.

Un second croisement partiel utilisé pour la phase de déduction. Pour ce dernier croisement, il suffit que les deux relations aient un terme en commun pour que la déduction puisse être faisable. Elle est basée sur un intérêt particulier à la structure du document pendant le traitement statistique, une modélisation contextuelle qui renforce la sélection des co-occurrents de chaque terme, une utilisation des relations existante dans notre hiérarchie de concepts et un processus qui facilite l'évaluation de certaines relations produites. Comme perspectives, nous allons développer ces idées dans notre architecture de découverte des connaissances pour le web sémantique. Ensuite, nous allons expérimenter d'avantage ce travail en présentant des échantillons plus importants de relations aux experts de domaine. Dans le même contexte, nous allons utiliser ces relations pour évaluer et enrichir la hiérarchie de concepts.

Références

Berland, M. and Eugene Charniak (1999). *Finding parts in very large corpora*. In Proc. of the 37th Annual Meeting of the Association for Computational Linguistics.

Bruandet, M.F. (1999). *Domain Knowledge Acquisition for an Intelligent Information Retrieval System.* Strategies and Tools, in Expert systems applications EXPERSYS - 90, Grenoble, pp231-235.

Claveau, V., P. Sebbillot, C. Fabre and P. Bouillon (2003). *Learning Semantic Lexicons from a Part-of-Speech and Semantically Tagged Corpus using Inductive Logic Programming*. Journal of Machine Learning Research, special issue on ILP. J. CUSSENS & S. DZEROSKI , Eds. (2000). Learning Language in Logic. LNAI. Springer Verlag.

Gamallo, P., M. Gonzalez, A. Agustini, G. Lopes, and V. S. de Lima. 2002. *Mapping syntactic dependencies onto semantic relations*. ECAI Workshop on Machine Learning and Natural Language Processing for Ontology Engineering, Lyon, France.

Grefenstette, G. (1992). *Use of syntatic context to produce terms association list for text retrieval*. In Conference in Researche and Developement in Information Retrieval (SIGIR'92), Copenhagen, Denmarke, pages 89-97.

Haddad, M.H (2002). *Extraction et impact des connaissances sur les performances des systèmes de recherche d'information*. Thèse de doctorat, université joseph Fourier.

Hearst, M.A. (1992). Automatic acquisition of hyponyms from large text corpora. In proceedings of the fourteenth international conference on computational linguistics, Nantes, France, pages 539-545.

Karoui, L., Aufaure, M-A., Bennacer, N. (2007). «*Extraction Contextuelle de Concepts Ontologiques pour le Web Sémantique*», 18èmes Journées Francophones d'Ingénierie des Connaissances (IC'2007), Plate-forme AFIA, Grenoble.

Maedche, A.D. (2002). *Ontology Learning for the Semantic Web*, Kluwer Academic Publishers, Norwell, MA.

Maedche, A. and S. Staab (2000). *Discovering conceptual relations from text*. ECAI 2000.

Magnini, M. Negri, E. Pianta, L. Romano, M. Speranza, and R. Sprugnoli. 2005. *From Text to Knowledge for the Semantic Web: the ONTOTEXT Project*. SWAP 2005, Semantic Web Applications and Perspectives, Trento.

McDonald, R., F. Pereira, S. Kulick, S. Winters, Y. Jin, and P. White. 2005. *Simple Algorithms for Complex Relation Extraction with Applications to Biomedical IE*. 43rd Annual Meeting of the Association for Computational Linguistics, pp. 491-498.

Morin, E. (1999). *Using Lexico-Syntactic Patterns to Extract Semantic Relations between Terms from Technical Corpus*. In Proceedings, 5th International Congress on Terminology and Knowledge Engineering (TKE), 268–278.TermNet, Innsbruck, Austria.

Nazarenko, A. 1994. *Compréhension du langage naturel : le problème de la causalité*. Thèse de doctorat.

Reinberger, M.L., P. Spyns, and A.J. Pretorius. 2004. *Automatic initiation of an ontology*. On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, LNCS 3290, Napa, Cyprus, pp. 600-617.

Rijsbergen, C.V. (1979). Information Retrieval. Butterworths, London.

Sabou, M. (2004). *Extracting ontologies from software documentation: a semi-automatic method and its evaluatio*n. In: Proceedings of the ECAI-2004 Workshop on Ontology Learning and Population (ECAI-OLP).

Schutz A. and P. Buitelaar. 2005. *RelExt: A Tool for Relation Extraction from Text in Ontology Extension*. 4th International Semantic Web Conference (ISWC-2005), pp. 593-606.

Soderland, S. 1999. *Learning information extraction rules for semi-structured and free text*. Machine Learning, 34.

Soo-Guan Khoo, C. (1995). *Automatic identification of causale relations in text and their use for improving precision in information retrieval*. P.D thesis.

Stevenson, M. and M. Greenwood. 2005. *A Semantic Approach to IE Pattern Induction*. 43rd Meeting of the Association for Computational Linguistics (ACL-05), Ann Arbour, Michigan, p. 379-386.

Yangarber, R., R. Grishman and P. Tapanainen, 2000. *Unsupervised Discovery of Scenario-Level Patterns for Information Extraction*. 6th ANLP, Seattle, pp. 282-289.

Summary

Relation extraction is a difficult open research problem. In our research, we focus on extracting relations among the ontological concepts in order to build a domain ontology. For this, we define a contextual relation discovery algorithm that applies different textual analyses in order to extract, deduce, label and validate the domain relations. Our method is based on a rich contextual modelling that strengthens the term co-occurrence selection, a use of the existent relations in the concept hierarchy and a stepping between the various extracted relations to facilitate the evaluation made by the domain experts. Our main perspective is using these relations for the concept hierarchy evaluation and enhancement.

SWAR : Modèle de génération des règles d'associations sémantiques à partir d'une base d'associations

Thabet Slimani*, Boutheina Ben Yaghlane**
Khaled Mellouli**

*ISG de Tunis, 41 rue de la liberté, Bouchoucha, Bardo 2000, Tunisie
thabet.slimani@issatm.rnu.tn,
http://www.isg.rnu.tn/larodec/members
**IHEC Carthage, Carthage Présidence 2016, Tunisie
boutheina.yaghlane@ihec.rnu.tn, khaled.mellouli@ihec.rnu.tn,

Résumé. Dans le domaine du Web sémantique, la théorie des associations sémantiques permet de trouver des associations directes et/ou imprévues entre les élements d'une ontologie. Pour un meilleur apport sémantique, nous proposons de suivre un processus de fouille des règles associatives à partir d'un ensemble d'associations déjà découvertes. Dans cet article, nous présentons un nouveau modèle SWAR (Semantic Web Association Rule) qui suppose une transformation préalable des associations stockées et prend en entrée une requête paramétrée générique permettant d'orienter la découverte des règles. Les résultats expérimentaux ont été appliqués sur un exemple de données synthétiques.

1 Introduction

Certaines applications du domaine analytique nécessitent une approche d'extraction et de stockage des interactions entre les entités d'une ontologie. Les chemins connectant deux entités spécifiées sont considérés comme des associations sémantiques (Aleman-Meza et al. (2003)) (Anyanwu et al. (2005)) (Aleman-Meza et al. (2005)). Dans cet article, les données et les informations d'une association sémantique sont stockées dans une base d'association (ASB).

Associée à la croissance du nombre d'associations contenues dans une base d'associations, la capacité d'extraire la connaissance à partir de celle-ci pour l'aide à la décision devient de plus en plus importante et souhaitable. Cependant, la fouille de données est utile pour la transformation de vastes quantités de données sémantiquement riches (associations sémantiques) en connaissances utiles.

La découverte des règles d'associations sémantiques est confrontée à plusieurs défis qui se résument comme suit : premièrement, les entités d'une association possèdent une structure séquentielle (séquence d'entités connectée via une séquence de propriétés) ayant une présentation plus complexe qu'un attribut d'une base de données classique. Deuxièmement, les éléments d'une association (entités et propriétés) possèdent des positions contextuelles ayant une signification sémantique ce qui est inexistant au niveau d'un simple attribut de base de données ordinaire. Troisièmement, les données d'une association sémantique semblent beaucoup plus riches en sémantique que celles d'une base de données classique.

Pour répondre à ces défis, nous proposons de suivre un processus de fouille de données à partir de ces associations afin de dégager des règles associatives dont l'évaluation sera basée sur la confiance et le support.

Dans le but de rendre la fouille à partir des associations sémantiques réalisable, nous présentons un modèle qui aide à la découverte des règles d'associations sémantiques (SWAR). Les travaux recensés sur les règles d'associations ont démontré l'efficacité de la fouille des associations basée sur requête/contrainte (Lakshmanan et al. (1999)) (Ng et al. (1998)) (Srikant et al. (1997)) (Baralis et Psaila (1997)) (Dehaspe et Toivonen (1999)). Ces travaux sont applicables à notre modèle SWAR.

Cet article propose, dans un premier temps, une brève présentation des travaux liés aux règles d'association. La section 3 présente une description des associations et des règles sémantiques suivie de la proposition des contraintes spécifiées au niveau de la détection des règles sémantiques (section 4). La section 5 décrit les différentes phases du modèle SWAR pour la génération des règles sémantiques. Dans la section 6, l'algorithme SWARM est présenté. Les résultats expérimentaux sont détaillés dans la section 7. Enfin, la section 8 propose une conclusion et quelques perspectives de recherche future.

2 Travaux liés

Les travaux effectués sur les règles d'association dans le domaine de la fouille de données sont très nombreux. L'algorithme de base Apriori (Agrawal et Srikant (1994)) est l'algorithme le plus populaire, il a été développé pour la découverte des règles d'association dans des bases de données larges. Parmi les travaux qui touchent l'aspect sémantique au niveau des règles d'associations, nous pouvons citer le travail de (Jiang et Tan (2006)) qui décrit un processus de fouille des données à partir des méta-données RDF pour la génération des règles d'associations généralisées. Dans le travail de (Louie et Lin (2002)), il y a une présentation d'une approche proposant un nouveau modèle permettant la découverte des règles d'association orientées sémantique. Un autre travail permettant de traiter l'expansion sémantique des requêtes combinant des règles d'associations avec des ontologies et des techniques de recherche documentaire est décrit dans (Song et al. (2005)). Récemment, un travail sur l'intégration des règles d'associations et des ontologies pour une expansion des requêtes sémantiques a été proposé dans (Song et al. (2007)). L'apport de notre travail est remarqué au niveau de la fouille des règles à partir des associations déjà stockées dans une base d'associations, ce qui est différent des autres travaux qui travaillent directement sur des ontologies RDF. Nous précisons que le travail présenté dans cet article est une continuité de notre travail soumis au journal électronique de l'intelligence artificielle (Slimani et al. (2006)).

3 Description d'une association et d'une règle sémantique

Les associations sémantiques se rapportent à des relations complexes entre des entités, des propriétés et des concepts. Elles accordent des significations aux informations en les rendant compréhensibles et mesurables. Formellement, une *association sémantique* est une séquence de relations, connectant deux entités sources, représentée par une séquence de noeuds/entités (*SN*) connectée par une séquence de propriétés (*SP*). Une base d'associations est une base

de données composée par les attributs : "*AttrSource*", "*AttrCible*", "*SP*", "*SN*" et "*Assoc*" désignant, respectivement, les attributs de l'entité source, les attributs de l'entité cible, la séquence de propriétés, la séquence de noeuds/entités de l'association et l'expression de l'association. A titre d'exemple, l'association A = {US0$\xrightarrow{TakesCourse}$C0$\xrightarrow{Advisor}$ FP0$\xrightarrow{MasterDegreeFrom}$U0} est une association directe formée par la séquence des noeuds (*SN*) composée par quatre entités (une entité source ("US0"), une entité cible ("U0") et deux entités intermédiaires ("C0"et "FP0")) et une séquence de propriétés (*SP*) contenant trois propriétés ($\xrightarrow{TakesCourse}$, $\xrightarrow{Advisor}$, $\xrightarrow{MasterDegreeFrom}$). L'association A contient en total 7 éléments. Si l'association A est stockée dans ASB alors, "*AttrSource*"="US0", "*AttrCible*"="U0", "*SP*"= *SP*, "*SN*"=*SN* et l'attribut "*Assoc*"=A.

Une *règle d'association sémantique* se définie comme une implication de la forme A⟶B (A et B sont deux associations variables) satisfaisant trois conditions :

1. A$\subset\mu$, B$\subset\mu$ et A$\cap$B=$\emptyset$;
2. $\forall$ les ensembles d'associations E_A et E_B contenant respectivement les valeurs des associations A et B, il existe respectivement deux ensembles d'associations E_{AR} et E_{BR} contenant des entités sources similaires à E_A et E_B. La variable μ désigne l'ensemble résultant de l'union de E_{AR} et E_{BR} ($\mu = E_{AR} \cup E_{BR}$)
3. Les règles associatives sont générées à partir de la correspondance binaire entre E_{AR} et E_{BR}. Le support et la confiance d'une règle sont définis comme suit :

$$Support(A \longrightarrow B) = \frac{|Ass_{ab}|}{|\mu|} \quad (1)$$

$$Confiance(A \longrightarrow B) = \frac{|Ass_{ab}|}{|\mu_a|} \quad (2)$$

où $|\mu|$ est la cardinalité de μ, $|Ass_{ab}|$={entités | $\forall$ X$\in$(A$\cup$B)} représente la cardinalité de l'apparition d'une entité source contenue dans les associations de E_{AR} d'une manière similaire à une entité source dans E_{BR} avec un contexte bien défini, μ_a={entités | $\forall$X$\in$(A)} représente l'ensemble des associations contenues dans E_{AR} et $|\mu_a|$ représente la cardinalité de μ_a. Une règle d'association forte est caractérisée par un support>MinSup et une confiance>MinConf. Les seuils du support minimum (MinSup) et de la confiance minimale (MinConf) doivent être fixés d'avance par l'expert.

4 Contraintes spécifiées au niveau des règles sémantiques

La génération des règles associatives à partir d'une base ASB est une tâche complexe qui nécessite une étape primordiale de modélisation. Nous présentons dans ce qui suit le modèle SWAR (Semantic Web Association Rule) pour la génération des règles d'associations sémantiques. Dans ce modèle, nous définissons un ensemble de concepts permettant d'en faciliter la compréhension.

Soit l'expression suivante de la forme : ΔX = {ΘA : $\Theta B \Rightarrow \Theta X$} où ΘA et ΘB désignent deux associations sources variables et ΘX désigne une entité cible variable permettant de guider le processus de fouille des règles d'associations. En d'autres termes, ΥX=(ΔX, *VCT*,

MinSup, MinConf) dénote une requête paramétrée variable qui cherche toutes les règles d'associations fortes (support>MinSup et confiance>MinConf) dont *VCT* représente un contexte variable ayant comme valeur "*SN* " (séquence des noeuds), "*SP* " (séquence de propriétés) ou "*" (tous les éléments de l'association).

L'expression de la variable ΘA ou ΘB est représentée par l'écriture suivante : (<e_1 :p_1>,..., <e_n :p_n>) incluant les éléments d'une association variable, tel que e_i représente le nom d'une entité (une classe ou une instance de classe), p_i désigne la propriété issue de e_i et si e_i est terminal (dernière entité de l'association) alors p_i prend la valeur "nul".

Soit ΥX=(ΔX, *VCT*, MinSup, MinConf) une requête ayant comme paramètre ΔX et comme critère de recherche *VCT*. Nous définissons dans ce qui suit les contraintes du modèle SWAR : la première contrainte est spécifique aux séquences de propriétés. En cas de l'application de cette contrainte, la variable *VCT* est fixée à "*SP*". Ce type de contraintes est adopté pour obtenir un ensemble de règles selon la sémantique des opérations exécutées sur les entités qui forment l'association (exemple : la recherche des règles concernant la similarité des opérations exécutées par un internaute sur un site commercial aux opérations spécifiant l'intérêt pour l'achat d'un produit bien déterminé). La deuxième contrainte est spécifique aux séquences de noeuds. Avec cette contrainte, la variable *VCT* est fixée à "*SN*". Ce type de contraintes est adopté pour extraire un ensemble de règles concernant les entités contenues dans une association indépendamment des opérations sur lesquelles elles sont exécutées (exemple : la recherche des règles concernant la similarité de l'existence des articles acceptés dans une conférence donnée avec le même contenu que d'autres articles placés dans d'autres conférences). Enfin, la troisième contrainte concerne tous les éléments qui forment l'association. En cas de l'utilisation de cette contrainte, la variable *VCT* est fixée à "*". L'intuition derrière l'utilisation de cette contrainte est d'obtenir un ensemble de règles dont la comparaison concerne tous les éléments d'une association.

5 Modèle SWAR pour la génération des règles sémantiques

Le modèle SWAR pour la découverte des règles associatives sémantiques est basé sur les phases suivantes :

5.1 Phase-1 : Transformation des variables relatives à une requête de règles d'associations

Pour rendre la fouille des règles associatives faisable, il est indispensable de transformer chaque association dans ASB en une séquence de données stockée dans des matrices spécifiques. A titre d'exemple, pour le cas de la variable ΘA, Transformation (ΘA)=<M_e[e_1,...,e_n] ; M_p[p_1,...,p_n]> où e_i et p_i représentent, respectivement, les colonnes des entités stockées dans une matrice M_e et les colonnes des propriétés (rdfs :domain) stockées dans une matrice M_p.

Exemple d'une requête : ΥX = (ΔX, *, MinSup, MinConf) : Détection des règles fortes pour la recherche de la tendance des personnes, qui ont acheté des billets d'avions, de faire des attentats dans le même vol (Figure 1).

Soit ΔX = $\{\Theta A : \Theta B \Rightarrow \Theta X\}$ le paramètre de la requête ΥX. Les variables de la requête sont définies par les expressions suivantes :

- ΘA=(<PERSON :“Purchased”><Ticket :“ForFlight”> <Flight :“nul”>) ;
- ΘB= (<PERSON :“IsMemberOf”> <TerroristGroup :“nul”>) et
- ΘX=(<PERSON>) qui représente l’entité variable qui sera retournée comme résultat de la requête.

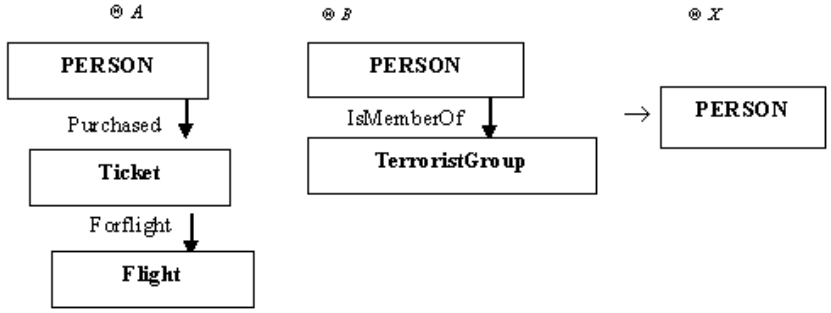


FIG. 1 – *Exemple de formulation d’une contrainte de requête.*

Afin de simplifier le travail d’appariement des associations dans ASB avec les entités dans M_e et les propriétés dans M_p, les variables ΘA et ΘB sont transformées comme suit : Transformation(ΘA)=<M_e[“PERSON”, “Ticket”,“Flight”] ; M_p[“Purchased”,“ForFlight”]>, et Transformation (ΘB)=<M_e[“PERSON”, “TerroristGroup”] ; M_p[“IsMemberOf”]> .

Ces transformations permettent de recupérer toute association dans ASB ayant un appariement exact avec M_e et M_p avec les attributs “*SP*”, “*SN*” ou les deux (*).

5.2 Phase-2 : Extraction de la séquence des associations relatives à un appariement exact avec les variables d’une requête de règles d’associations.

Soient ΘA et ΘB deux variables concernées par une requête de règles d’associations ayant comme valeurs E_A et E_B. Les deux ensembles E_A et E_B décrits dans la section 3 sont construits à travers un processus d’appariement. Ensuite, la recherche des règles d’association sera restrictive à la base d’association minimale ASB_{Min} (ASB_{Min} incluant les ensembles E_A et E_B) afin d’alléger le processus d’extraction. Dans le tableau 1, nous présentons un exemple illustratif d’une base ASB_{Min} (6 associations) déjà extraite à partir de la base d’associations initiale.

5.3 Phase-3 : Conversion des séquences d’associations trouvées dans la phase 2 sous format de la méthodologie rough set et génération des règles possibles

Nous avons opté pour le choix de la méthodologie *rough set* (Munkata (1998)) puisqu’elle retourne, à partir d’un sous ensemble optimal des attributs sources, assez d’information concernant des attributs cibles déjà prédéfinis. En adoptant cette méthodologie, l’univers Ω des attributs de la table binaire (tableau 3) est divisé en deux ensembles disjoints, un ensemble constituant les attributs de condition Ω_c et un deuxième ensemble constituant les attributs de décision

idAssoc	Associations extraites
A 1	SarahWhite1($\xrightarrow{Purchased}$)(3699)($\xrightarrow{ForFlight}$)(AA205)
A 2	SarahWhite1($\xrightarrow{Purchased}$)(3697)($\xrightarrow{ForFlight}$)(AA203)
A 3	SarahWhite2($\xrightarrow{Purchased}$)(3698)($\xrightarrow{ForFlight}$) (AA203)
A 4	SarahWhite3($\xrightarrow{Purchased}$)(3699)($\xrightarrow{ForFlight}$)(AA206)
A 5	SarahWhite10($\xrightarrow{IsMemberOf}$)(Al-Qaeda)
A 6	SarahWhite5($\xrightarrow{IsMemberOf}$)(Al-Qaeda)

TAB. 1 – *ASB_{Min} : exemple de base d'associations minimale contenant des associations extraites après l'exécution de la phase2.*

Ω_d. Les valeurs de l'attribut de condition sont contenues dans l'ensemble E_{AR}={A1, A2, A3} et les valeurs de l'attribut de décision sont contenues dans l'ensemble E_{BR}={A4, A5} (Tableau 2). Ces valeurs ne sont pas obtenues arbitrairement, mais à partir des associations contenues dans le tableau 1. En effet, le tableau 2 contient deux sous-ensembles d'associations (E_{AR} et E_{BR}) correspondants aux associations provenant de ASB ayant un appariement correct au niveau de leurs entités sources par rapport aux entités sources e1 et e'1, respectivement, contenues dans les associations des ensembles E_A et E_B. A la fin de la phase 3, la base d'association minimale (ASB_{Min}) devient comme représentée par le tableau 2.

idAssoc	Associations extraites
A 1	SarahWhite1($\xrightarrow{WorksFor}$)(MirageCorporation)($\xrightarrow{ElectedLeader}$)(SarahWhite0)
A 2	SarahWhite1($\xrightarrow{WorksFor}$)(AperInc)($\xrightarrow{ElectedLeader}$)(ZackaryBlack)
A 3	SarahWhite2($\xrightarrow{WorksFor}$)(AperInc)($\xrightarrow{ElectedLeader}$)(ZackaryBlack)
A 4	SarahWhite5 ($\xrightarrow{WorksFor}$)(MirageCorporation)($\xrightarrow{ElectedLeader}$)(SarahWhite0)
A 5	SarahWhite10($\xrightarrow{WorksFor}$)(AperInc)($\xrightarrow{ElectedLeader}$)(ZackaryBlack)

TAB. 2 – *ASB_{Min} : exemple d'une base d'associations minimale contenant des associations extraites après l'exécution de la phase 3.*

idAssociation	A4(SarahWhite5)	A5(SarahWhite10)
A1(SarahWhite1)	1	0
A2(SarahWhite1)	0	1
A3(SarahWhite2)	0	1

TAB. 3 – *Exemple d'une table binaire.*

Pour générer des règles associatives il est primordiale de construire une table binaire (tableau 3) en se basant sur la nouvelle ASB trouvée. Le remplissage de la table binaire est basé sur le principe suivant : les associations contenues dans les ensembles E_{AR} et E_{BR} doivent être comparées ensemble, afin de dégager, selon le critère spécifié, les associations similaires (comparaison à partir de la deuxième entité de chaque association). Cependant, la table binaire doit être structurée comme une matrice binaire (produit cartésien des associations dans l'ensemble E_{AR} et l'ensemble E_{BR}). Les valeurs binaires placées dans la table binaire sont représentées par 1 en cas d'un appariement correct et par 0 dans le cas contraire.

Les règles possibles (tableau 4) correspondantes à l'exemple présenté dans la figure 1 sont : A1→A4, A1→A5, A2→A4, A2→A5, A3→A4, A3→A5.

idRègle	Règles	Supp	Conf
R1 (A1→A4)	SarahWhite1→SarahWhite5	$0,33$	1
R2 (A1→A5)	SarahWhite1→SarahWhite10	0	0
R3 (A2→A4)	SarahWhite1→SarahWhite5	0	0
R4 (A2→A5)	SarahWhite1→SarahWhite10	$0,33$	$0,75$
R5 (A3→A4)	SarahWhite2→SarahWhite5	0	0
R6 (A3→A5)	SarahWhite2→SarahWhite10	$0,33$	$0,75$

TAB. 4 – *Règles possibles.*

Une règle d'association X→Y a comme support *s* si *s*% des lignes dans la table binaire contiennent X∪Y (compter les valeurs 1 de la table binaire ou X est associé avec Y). Une règle apparaît dans la table binaire avec une confiance *c* si *c*% des lignes dans cette table qui contiennent X contiennent aussi Y. A titre d'exemple, le support de la règle R1 est obtenu par l'expression suivante : Supp (R1)=0,33 (où 0,33=1/3 représente la valeur de l'apparition de l'entité source (SarahWhite1) contenue dans A1 et A2 avec la valeur 1 par rapport au nombre total des lignes de la table binaire) et la confiance de la règle R1 est obtenue par l'expression suivante : Conf(R1)=supp (R1)/conf(X)= 0,33/0,33=1 (où la valeur conf(X)=0,33 représente le pourcentage (support) de l'apparition de l'entité source contenue dans l'association A1 par rapport à la colonne de l'association A4 contenue dans la table binaire).

Dans le cas de la requête ΥX, si l'utilisateur choisit comme MinConf=0.7 et un MinSup=0.3, alors les règles fortes à générer par notre méthode sont R1, R4 et R6. Par exemple, la règle R1 signifie qu'on a une certitude de 100% que le voyageur "SarahWhite1" est en relation avec des personnes reliées avec des groupes terroristes et par conséquent, ce voyageur a tendance de faire des attentats dans le vol correspondant à son ticket.

6 Algorithme SWARM (Semantic Web Association Rule Mining)

Etant donnée une base d'associations (ASB) composée de *n* associations. L'idée principale de notre algorithme consiste à balayer, dans une première étape, la base d'associations (selon la contrainte de la requête : *SP* ou *SN*) afin de décomposer les associations en des matrices. Deux matrices sont consacrées pour contenir les données des associations : matrice des propriétés (M_p) et matrice des entités/objets (M_e). Chaque matrice possède comme dimension (*n*x*m*) où *n* désigne le nombre d'associations contenues dans ASB (ayant un appariement correct avec le paramètre spécifié par la requête) et *m* désigne le nombre de colonnes (propriétés ou entités) qui forment la matrice M_p ou M_e). Ensuite, l'algorithme SWARM procède de la même manière que les phases décrites dans la section 5.

Algorithme SWARM
Début
Discrétisation des données :ΘA,ΘB, ΘX, M_e, M_p,
Etape 1 : Extraction des ensembles E_A et E_B
Etape 2 : Extraction de E_{AR} et E_{BR}
Etape 3 : Construction de la table binaire
Pour chaque association i$\subset E_{AR}$
$C_1=F_1$; // *placer dans* C_1 *toutes les associations de* E_{AR}
i=i.next ;
FinPour
$C_k=C_1$;
Tant que ($C_k \neq \emptyset$) faire
Pour chaque entité ***es***$\subset$Ck // ***es*** *désigne l'entité source d'une association appartenant aux lignes de la table binaire*
Pour chaque association ***ec***$\subset E_{BR}$ // ***ec*** *désigne l'entité cible d'une association appartenant aux colonnes de la table binaire*
s.count=Appariement (***es***,***ec***) ;
F_k={c$\in C_k$|s.count >= MinSup} ;
FinPour
$F_k=F_k \cup F_{ec}$; // *itemsets candidats*
FinPour
$C_k=F_k$;
Assoc={A$\subset F_k$|c.count >= Minconf}
FinTantque
Fin

---

7 Résultats expérimentaux

L'objectif de ce travail est de proposer une nouvelle méthode de génération des règles associatives sémantiques. Pour ce faire, nous avons développé un prototype pour évaluer notre

travail. Dans nos expérimentations, nous avons utilisé une ASB contenant des associations générées à partir des ontologies OWL Lite (Dean et al. (2003)). L'architecture du modèle SWAR est définie par la Figure 2. Dans cette architecture, nous présentons les différents modules responsables du fonctionnement de notre application. Pour évaluer la performance de la méthode proposée, nous avons conduit un ensemble d'expériences sur des données synthétiques.

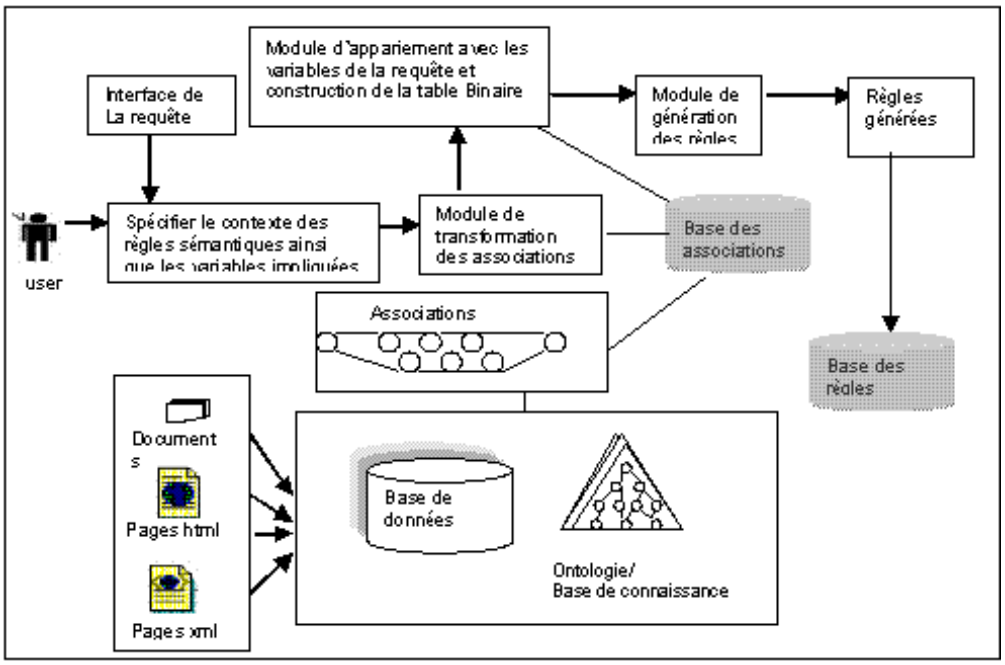


FIG. 2 – *Architecture du modèle SWAR pour la génération des règles sémantiques.*

Etude des performances : Trois ensembles d'expériences ont été exécutés sur des données synthétiquement produites. La machine utilisée est munie d'une fréquence d'horloge de 2,6 gigahertz et de 500 Mo de mémoire centrale.

Nous avons appliqué des expérimentations sur des bases d'associations contenant, respectivement, 1700, 3000 et 5000 associations. Les fréquences de support minimum et de la confiance minimale ont été fixées respectivement à 20% et 80%. Par exemple, les résultats appliqués sur l'exemple de 1700 associations (Figure 3) donnent 234 règles fortes dont la variable condition est formée par un seul élément, 22 règles pour des attributs dont la variable condition est formée par deux éléments et aucune règle pour toute autre combinaison (dans le cas où l'utilisateur s'intéresse à des règles fortes au niveau des propriétés).

Les expérimentations schématisées dans la figure 3 montrent que le temps d'exécution correspondant au critère spécifié par l'utilisateur pour tous les éléments d'une association présente un temps d'exécution plus élevé par rapport aux autres critères, cela revient à la taille élevée des matrices lors de la transformation des associations.

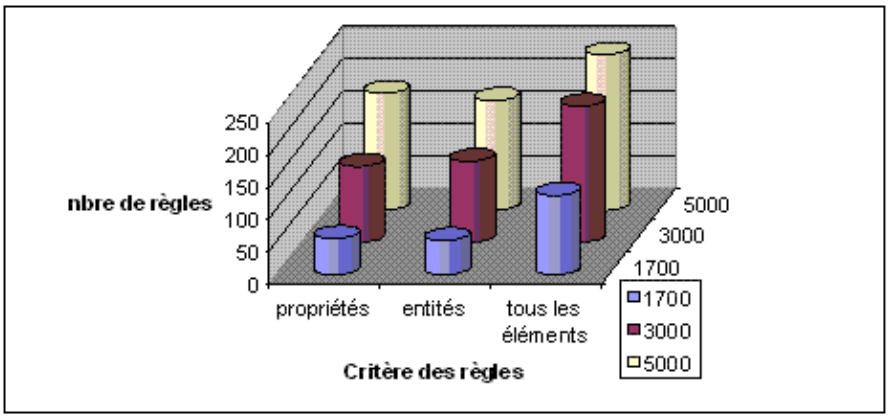


FIG. 3 – *Graphe comparatif du temps d'exécution appliqué sur trois exemples contenant un nombre d'associations différent.*

Pour accélérer le temps d'exécution, nous avons développé l'algorithme BSWARM (Binary Semantic Web Association Rule Mining) dont le principe de génération des associations s'inspire de notre algorithme BARM (Binary Association Rule Mining)(Slimani et al. (2004)).

Nous rappelons que BARM tente à diminuer le temps de calcul résultant des balayages multiples de la base de données, pour ce faire, il utilise la structure Peano Tree (Ptree) permettant de convertir la base de données en un fichier stocké sur disque et contenant des données binaires sur lesquelles nous pouvons faire des opérations binaires (*ANDing de Ptrees*). Avec ces opérations, le support de chaque itemset candidat peut être obtenu directement sans parcourir la base de données. Ce même principe est adopté par notre algorithme BSWARM où les associations contenues dans ASB sont transformées en des séquences de bits binaires, selon les paramètres de la requête, permettant ainsi de faciliter leurs comparaisons avec les paramètres de la requête sans recourir à des opérations coûteuses de construction des matrices de données.

Le graphe de la figure 4 montre une comparaison des résultats appliqués sur une ASB de 1700 associations en adoptant l'algorithme SWARM et BSWARM (à travers une variation de la valeur du support minimum dans l'intervalle [10%, 20%]). Les résultats dégagés montrent une accélération remarquable au niveau du temps d'exécution en adoptant l'algorithme BSWARM par rapport à l'algorithme SWARM.

8 Conclusion

Dans ce travail nous avons présenté une nouvelle méthode de détection des règles d'associations sémantiques stockées dans une base d'associations découvertes à partir des ontologies OWL Lite. Ainsi, la découverte des relations implicites entre les entités sources de deux associations permet de supporter la prise de décision. Les résultats produits montrent l'utilité de ce travail dans divers applications du monde réel. Néanmoins, le temps d'exécution est un peu élevé à cause des balayages multiples de la base d'association. Nous avons résolu ce problème

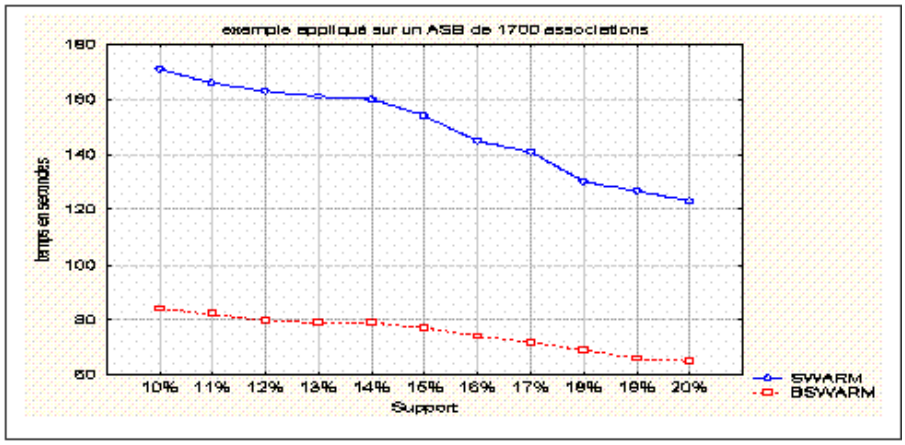


FIG. 4 – *Graphe comparatif des algorithmes SWARM et BSWARM pour des règles générées avec un support minimum qui varie dans l'intervalle 0.1 et 0.2.*

par le développement de l'algorithme BSWARM inspiré de notre algorithme BARM. Ainsi, nous proposons dans un prochain travail d'appliquer cette approche sur des données issues d'une base d'associations réelle.

Références

Agrawal, R. et R. Srikant (1994). Fast algorithms for mining association rules. In *20th Int. Conf. Very Large Data Bases (VLDB)*.

Aleman-Meza, B., P. Burns, M. Eavenson, D. Palaniswami, et A. Sheth (2005). An ontological approach to the document access problem of insider threat. In *IEEE International Conference on Intelligence and Security Informatics*, Atlanta, Georgia, USA, pp. 486–491.

Aleman-Meza, B., C. Halaschek, I. Arpinar, et A.Sheth (2003). Context-aware semantic association ranking. In *First International Workshop on Semantic Web and Databases.*, Berlin, Germany, pp. 33–50.

Anyanwu, K., A. P. Sheth, et A. Maduko (2005). Sem-rank : Ranking complex relationhip search results on the semanticweb. In ACM (Ed.), *14th International World Wide Web Conference.*, Chiba, Japan, pp. 117–127.

Baralis, E. et G. Psaila (1997). Designing templates for mining association rules. *Journal of Intelligent Information Systems* (9), 7–32.

Dean, M., D. Connolly, F. Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. F. P. Schneider, et L. Stein (2003). Owl web ontology language 1.0 reference. Technical report, W3C Working Draft.

Dehaspe, L. et H. Toivonen (1999). Discovery of frequent datalog patterns. In *Data Mining and Knowledge Discovery.*

Jiang, T. et A. Tan (2006). Mining rdf metadata for generalized association rules : Knowledge discovery in the semantic web era. In *Proceedings of the 15th International Conference on World Wide Web.*, Edinburgh, Scotland, pp. 951–952. ACM Press, New York, NY.

Lakshmanan, L., R. Ng, J. Han, et A. Pang (1999). Optimization of constrained frequent set queries with 2-variable constraints. In *the ACM SIGMOD Intl. Conf. on Management of Data.*, pp. 157–168.

Louie, E. et T. Lin (2002). Semantics oriented association rules. In *Fuzzy Systems, Proceedings of the 2002 IEEE International Conference on fuzzy systems. USA*, Volume 2, pp. 956–961. (Honolulu, HI).

Munkata, T. (1998). Rough sets. In *Fundamentals of the New Artificial Intelligence*, New York : Springer-Verlag., pp. 140–182.

Ng, R., L. Lakshmanan, J. Han, et A. Pang (1998). Exploratory mining and pruning optimizations of constrained association rules. In *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data.*, pp. 13–24.

Slimani, T., B. B. Yaghlane, et K. Mellouli (2004). Approche binaire pour la génération de fortes règles d'association. In *Proc of EGC 04, extraction et gestion de connaissances, Clermont-Ferrand, France.*, pp. 329–340.

Slimani, T., B. B. Yaghlane, et K. Mellouli (2006). Nouvelle approche pour la génération des associations à partir d' une base de connaissance. *Proposition 68 au Journal électronique d'intelligence artificielle, http ://jedai.afia-france.org/detail.php/?PaperID=68*.

Song, M., I. Song, X. Hu, R. Allen, et B. Robert (2007). Integration of association rules and ontologies for semantic query expansion. In *Data and Knowledge Engineering.*, Volume 63, issue 1 of *1*, pp. 63–75.

Song, M., I. Song, X. Hu, et R.Allen (2005). Semantic query expansion combining association rules with ontologies and information retrieval techniques. In *Lecture Notes in Computer Science, Springer Berlin / Heidelberg.*, pp. 326–335.

Srikant, R., Q. Vu, et R. Agrawal (1997). Mining association rules with item constraints. In *Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining.*, pp. 67–73.

Summary

In the field of semantic Web, the theory of semantic associations brings new unexpected relationships. Nevertheless, for a better semantic contribution, it is essential to follow a mining process starting from these associations to discover deeper semantics. In this paper we propose a new algorithm, called SWARM (Semantic Web Association Rule Mining), which aims to generate semantic rules concerning associations previously stored in association base. The experimental results have been applied to an example of synthetic data.

Fouilles archéologiques : à la recherche d'éléments représentatifs

Cyril De Runz*, Frédéric Blanchard*
Eric Desjardin*, Michel Herbin*,**

*CReSTIC-SIC, IUT de Reims Châlons Charleville,
Rue des Crayères, BP 1035, Reims Cedex 2, France
{cyril.de-runz,frederic.blanchard,eric.desjardin}@univ-reims.fr,
http://crestic.univ-reims.fr
**Antenne CReSTIC-Châlons, chaussée du port,
BP 541, 51012 Châlons-en-Champagne Cedex, France
michel.herbin@univ-reims.fr

Résumé. Définir les éléments les plus représentatifs au sein d'un SIG archéologique est une question d'actualité. En effet, déterminer l'élément qui représente le mieux un ensemble de fouilles archéologiques est important pour la valorisation de ces fouilles. Nous avons développé au sein du CReSTIC-SIC une méthode statistique de sélection de l'élément le plus représentatif d'un échantillon. La notion -quantitative- de représentativité est basée sur la transformation par rangs des dissimilarités entre éléments. Nous avons appliqué cette méthode statistique à la recherche de l'élément le plus représentatif au sein de données issues de fouilles, dont les caractéristiques sont représentées par des ensembles flous convexes et normalisés, sur les rues de la ville de Reims à l'époque romaine. Dans ce cadre, nous avons donc utilisé une métrique classique entre ensembles flous en tant que dissimilarité pour déterminer le tronçon de rue le plus représentatif de Reims à cette époque.

1 Introduction

Les Systèmes d'Information Géographique (SIG) permettent de stocker et de visualiser à la fois les objets spatiaux et les informations associées. Au sein d'un SIG, il est intéressant de distinguer les objets les plus représentatifs (relativement à des critères fixés) de l'ensemble des objets stockés. Par exemple, si l'on recherche quel est le boulevard le plus architecturalement représentatif des boulevards parisiens, le résultat serait vraisemblablement un boulevard de type haussmanien. En archéologie, dans l'optique de valorisation des éléments découverts durant des fouilles, il s'avère intéressant de déterminer celui qui représente le mieux les éléments découverts en terme de localisation, de période d'activité, de forme. Nous disposons dans le cadre du projet SIGRem (de Runz et al. (2007a)) d'une Base de Données Géographiques (BDG) intitulée *BDRues* sur les tronçons de rues de Durocortorum (Reims à l'époque Romaine). Nous

cherchons dans ce cadre le tronçon de rue le plus représentatif spatio-temporellement. Au sein du groupe SIC (Signal Image et Connaissance) du CReSTIC (Centre de Recherche en Sciences et Technologies de l'Information et de la Communication), nous avons défini une statistique (Blanchard (2005)) pour déterminer l'élément le plus représentatif d'un échantillon de données. Nous proposons dans cette communication d'utiliser cette statistique sur $BDRues$ afin de déterminer les tronçons de rues les plus représentatifs en fonction de leurs périodes d'activité, de leurs localisations et de leurs orientations.

La définition de la notion de représentativité d'un élément au sein d'un échantillon de données, utilise les statistiques de rangs. Les statistiques non paramétriques connaissent depuis quelques années un regain d'intérêt (David et Ngaraja (2003)) et leur utilisation en analyse de données permet notamment de s'affranchir de l'hypothèse de normalité et apporte une robustesse vis à vis des données aberrantes (Galambos (1975); Barnett (1976)). Ces statistiques amplement utilisées dans des domaines tels que le traitement d'images (Lukac et al. (2006); Vautrot et al. (2006)) ne le sont que peu dans l'exploitation des SIG archéologiques pour la valorisation. Nous proposons ici d'utiliser le Vecteur de Meilleur Rang Moyen (VMRM) (de Runz et al. (2007d)), qui extrait l'élément de représentativité maximale d'un ensemble de données.

La définition quantitative de représentativité, utilisée par le VMRM nécessite de disposer d'un indice de dissimilarité entre éléments. Cette dissimilarité est liée à la description des données. Or comme nos données archéologiques sont incertaines dans leurs localisations, orientations, et datations, nous avons précédemment (de Runz et al. (2007b,c)) modélisé, dans $BDRues$, ces différentes caractéristiques par des ensembles flous convexes et normalisés. C'est pourquoi, nous proposons d'utiliser une métrique classique (Grzegorzewski (1998)) entre ensembles flous convexes et normalisés comme dissimilarité afin de déterminer le VMRM dans $BDRues$.

Après avoir présenté le principe théorique de Vecteur de Meilleur Rang Moyen, son utilisation dans le contexte des rues de Durocortorum est proposée. Une discussion et des perspectives sont enfin exposées avant de conclure.

2 Le vecteur de meilleur rang moyen

Considérons un échantillon de données multidimensionnelles $S = \{x_1, x_2, \ldots, x_n\}$ dans un espace E de dimension $p \in \mathbb{N}$. On suppose que l'on dispose, sur cet échantillon, d'un indice de dissimilarité. Autrement dit, on suppose que l'on dispose d'un moyen de quantifier la dissimilarité entre deux éléments quelconques de notre échantillon S. On notera $\delta(x_i, x_j)$ la dissimilarité entre x_i et x_j $(i, j \in [1..n])$. La distance euclidienne est un exemple d'indice de dissimilarité.

2.1 Statistiques de rangs marginales

On notera $X_1, X_2, ..., X_n$ les variables (vecteurs) aléatoires dont les éléments de l'échantillon S sont les observations. Les statistiques d'ordres associées sont les $X_{(1)}$, $X_{(2)}, \ldots, X_{(n)}$ triés par ordre croissant (et on notera $x_{(1)}, x_{(2)}, \ldots, x_{(n)}$ les observations ordonnées associées). Par définition, les statistiques d'ordres sont donc intrinsèquement

liées à la façon dont sont triées les variables aléatoires. Dans le cas multidimensionnel, le tri n'est pas trivial, on se reportera à Barnett (1976) pour une étude des différentes techniques pour trier des vecteurs.

Considérons maintenant les n classements (i.e. les n tris) obtenus en utilisant les dissimilarités par rapport à chaque x_i. Autrement dit, pour chaque élément x_i, nous classons l'ensemble de l'échantillon par ordre de dissimilarité croissante avec x_i. Soit $Rg_{x_i}(x_j)$ le rang de la donnée x_j dans le classement par dissimilarité croissante à x_i. La valeur de $Rg_{x_i}(x_j)$ représente ainsi la position de x_j dans le classement des données les plus similaires à x_i. Par exemple, $Rg_{x_i}(x_j) = k$ ($k \in [1..n]$) signifie que x_j est la k-ième donnée de S la plus similaire à x_i, c'est à dire $x_{(k)}$ dans l'ordre induit par la donnée x_i.

On obtient donc, sur l'ensemble de l'échantillon, n classements des données.

2.2 Rang moyen d'une donnée

On calcule ensuite, pour chaque donnée x_j de S, la moyenne des rangs $Rg_{x_i}(x_j)$ qu'elle a obtenus au cours de ces n classements. On note ce rang moyen : $\overline{Rg}(x_j)$ et on a :

$$\overline{Rg}(x_j) = \frac{1}{n} \times \sum_{i=1}^{n} Rg_{x_i}(x_j).$$

Le rang moyen est un critère qui nous permet alors d'évaluer le potentiel d'une donnée à représenter l'échantillon auquel elle appartient. En effet, cette valeur moyenne traduit la façon dont une donnée est *la plus similaire* à l'ensemble des autres. On appelle cette notion la *représentativité d'une donnée dans son échantillon.*

2.3 Statistique de meilleur rang moyen

Nous terminons notre processus par la recherche dans l'échantillon, de la donnée ayant le plus petit rang moyen, c'est à dire la donnée de l'échantillon la plus représentative dudit échantillon.

Finalement, nous avons donc, au cours de ces étapes, défini une statistique exprimée comme une fonctionnelle des statistiques de rangs marginales et notée *VMRM* (Vecteur de Meilleur Rang Moyen) :

$$VMRM : (X_1, X_2, \ldots X_n) \mapsto \underset{X_i, i=1..n}{argmin} \left(\overline{Rg}(X_j)\right)$$

Cette statistique associe à un échantillon l'élément qui le représente le mieux. Cette notion de meilleur représentant d'un échantillon rejoint, d'un point de vue sémantique, la notion de représentant de classes en classification automatique des données. De plus, au même titre que la médiane, notre statistique est un estimateur robuste de position de l'échantillon. En effet, la donnée de meilleur rang moyen est un élément typique et représentatif de l'échantillon.

Dans la partie suivante, nous abordons l'utilisation du VMRM dans *BDRues* afin de déterminer les tronçons de rues trouvés de Durocorturum les plus représentatifs en terme de localisation, de période d'activité, d'orientation et des trois cumulées.

3 Tronçons de rues romaines les plus représentatifs

Trouver l'objet représentant le mieux un ensemble d'objets issus de fouilles archéologiques peut avoir un intérêt fort pour la valorisation du travail de fouilles en archéologie. Cet objet représente l'objet le plus classique trouvé au cours des fouilles en fonction des critères choisis. Dans cette partie, nous présenterons d'abord la BDG dédiée aux éléments de rues romaines à Reims, *BDRues*, qui nous servira de base de travail. Ensuite, notre travail se penchera sur la ou les mesures de dissimilarité choisies. Enfin nous étudierons les résultats issus de ce travail.

3.1 A propos de *BDRues*

Les données archéologiques sont des données spatio-temporelles, ce qui diffère des cas classiques des données géographiques. Quelques études, telles que Dragicevic et Marceau (2000), s'approchent conceptuellement de notre cadre de travail. Dans la base de données sur les rues de Durocortorum, les tronçons de rues sont caractérisés par des points ayant une orientation et une période d'activité.

La datation de la période d'activité des objets est généralement issue d'interprétations ou d'estimations dépendantes de l'environnement de la découverte (lieux de fouilles, stratigraphie, comparaison aux objets se situant dans la même pièce...). De plus, la codification linguistique de périodes temporelles n'a pas toujours la même représentation. Par exemple l'estimation du début du Bas Empire varie selon les experts entre 193 et 284 après J.C. Elle est donc largement incertaine et imprécise. Le géoréférencement est lui aussi sujet à de l'imprécision et/ou de l'incertitude liées à différents facteurs : positionnement du point de fouilles, position par rapport à la route, référentiel utilisé, mouvement de terrain. L'orientation de la route est aussi à redéfinir dans ce cadre. En effet, l'orientation est notamment dépendante de la technique d'estimation utilisée à l'époque de la fouille.

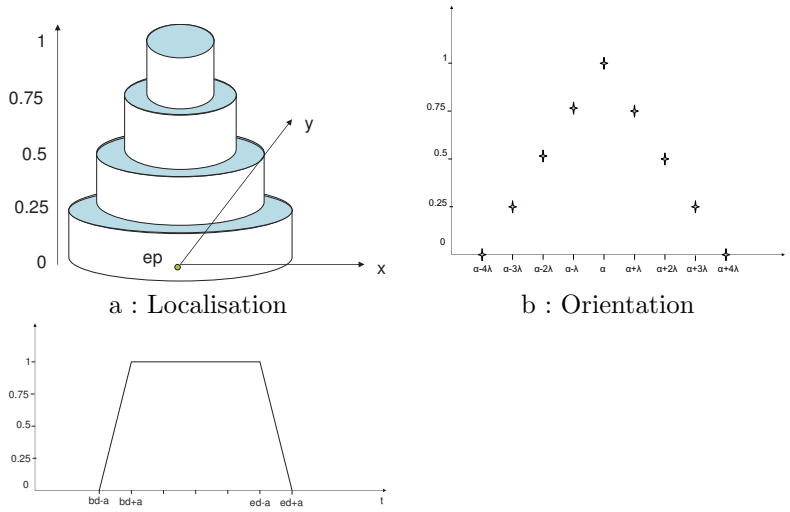
Nous représentons les orientations, les périodes d'activité, et les localisations par des ensembles flous convexes et normalisés soit respectivement par des nombres flous, des intervalles flous et ensembles flous spatiaux (2D). On peut ainsi prendre en compte cette incertitude (voir Figure 1).

Afin de pouvoir obtenir les vecteurs de meilleurs rangs moyens en terme de localisation, orientation, période d'activité et des trois conjuguées, nous devons définir les mesures de dissimilarités associées.

3.2 Détermination de la dissimilarité

Nous proposons d'utiliser une distance classique (Grzegorzewski (1998)) entre nombres et/ou intervalles flous comme mesure de dissimilarité. Soit F et G deux nombres et/ou intervalles flous, soit $F_{\alpha-}$ (resp. $G_{\alpha-}$) et $F_{\alpha+}$ (resp. $G_{\alpha+}$) les bornes inférieure et supérieure de l'α-coupe F_α de F (resp G_α de G), alors la distance entre F et G est obtenue par :

$$D(F,G) = \int_0^1 |F_{\alpha-} - G_{\alpha-}| + |F_{\alpha+} - G_{\alpha+}| d\alpha.$$



a : Localisation

b : Orientation

c : Période d'activité

Fig. 1 – *Modèles flous pour la localisation, l'orientation et les périodes d'activité des rues romaines*

Nous utiliserons cette mesure pour le calcul de la dissimilarité d'orientations (D_{orien}) et de périodes d'activité entre éléments (D_{date}).

Pour le calcul de la dissimilarité de localisation, en raison du caractère cylindrique de la fonction d'appartenance des ensembles flous spatiaux associés aux données, nous calculons la mesure de dissimilarité D_{loc} à partir de leurs projections floues sur le plan passant par les centres des localisations (voir Figure 2).

A l'instar des mesures de dissimilarités pour le calcul des VMRM en terme de localisation, d'orientation ou de période d'activité, nous avons besoin de prendre une mesure de dissimilarité pour l'ensemble des caractéristiques. Afin de l'obtenir, nous normalisons la dissimilarité liant un objet à un autre par la dissimilarité maximale du premier objet à l'échantillon. Nous effectuons cette normalisation des dissimilarités pour toutes les caractéristiques. La dissimilarité résultant de la moyenne de ces dissimilarités normalisées sera considérée comme la dissimilarité globale. Ainsi, soit deux éléments X et Y de BDRues alors :

$$D_{global}(X,Y) = \frac{1}{3} \times \sum_{i \in \{loc, orien, date\}} \frac{D_i(F,G)}{\max\limits_{J \in BDRues} D_i(F,J)}$$

Nous utilisons cette dissimilarité afin d'extraire le VMRM global.

3.3 Résultats

En calculant le VMRM pour la localisation, pour l'orientation, pour la période d'activité et les trois conjuguées nous obtenons les plans de la Figure 3.

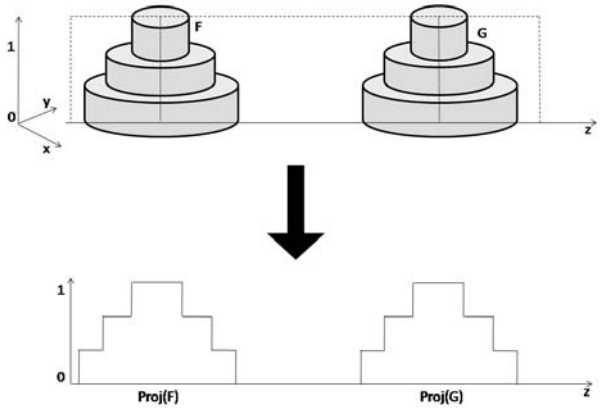


FIG. 2 – *Projections pour le calcul de dissimilarité des localisations*

On peut s'apercevoir que les tronçons représentatifs sont différents (excepté pour le VMRM global et le VMRM date) en fonction de la recherche effectuée, et en cela reflète bien l'influence de chacune des caractéristiques des données archéologiques. Ainsi, nous observons que le VMRM Global n'a pas la même orientation que le VMRM Orientation. Cela est dû au fait que le nombre de tonçons de rues dont l'orientation est proche de celle du VMRM Global est presque égal à celui des tonçons de rues dont l'orientation est proche de celle du VMRM Orientation (17 contre 16). Enfin, c'est la période d'activité des tonçons de rues qui a entrainé le décalage au centre du VMRM Global par rapport au VMRM Localisation. Si nous ne regardons que le VMRM Global, il est celui qui à la fois : est au centre, a l'une des deux orientations principales et a une période d'activité (voir Figure 4) qui se situe dans l'âge d'or de la période romaine de Reims (début du gallo-romain - fin du Bas-Empire).

4 Discussion et conclusion

L'extraction et la visualisation d'éléments représentatifs au sein d'un SIG sont importantes pour la valorisation des données et plus encore dans le cadre de données de fouilles archéologiques. Les données de fouilles sont incertaines, nous avons donc choisi de les représenter par des ensembles flous. Nous avons dans des précédents travaux défini une statistique de rangs permettant d'extraire d'un échantillon de données, l'élément de meilleur rang moyen. Nous l'avons utilisée dans le cadre des données de fouilles sur les rues de Reims à l'époque romaine pour en extraire les tronçons trouvés de rues romaines les plus représentatifs en terme de date, de localisation et/ou d'orientation. Ce travail constitue la première étape d'extraction d'éléments caractéristiques d'un échantillon de données issues de fouilles archéologiques.

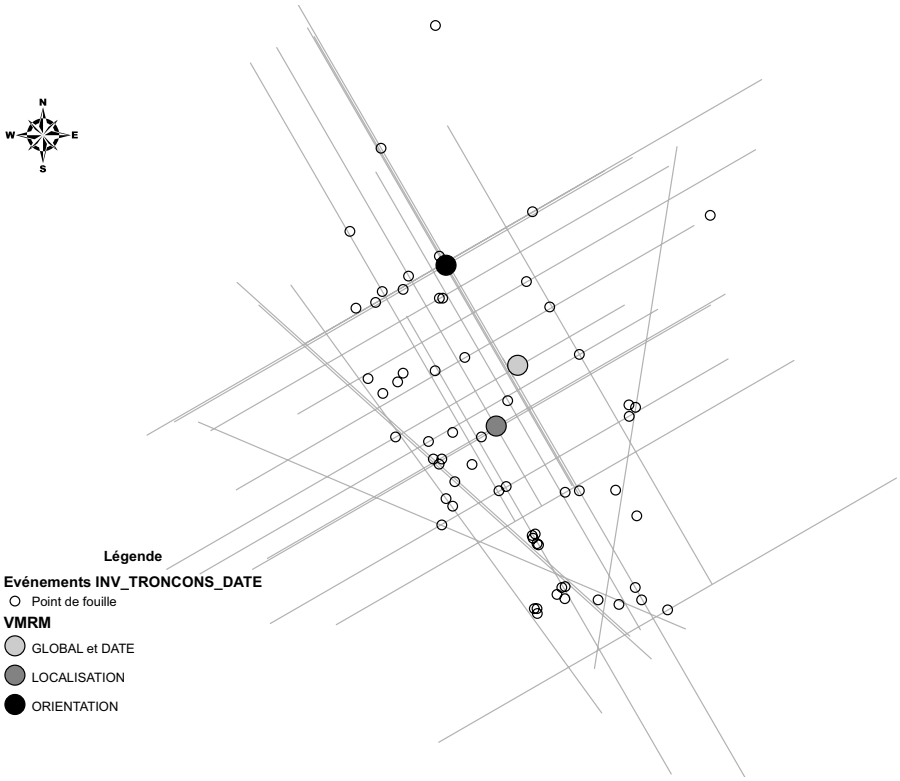


FIG. 3 – *Différents VMRM pour BDRues*

En effet, l'élément le plus représentatif d'un jeu de données est l'élément le plus similaire aux autres, mais pas forcément le plus emblématique. Ainsi, bien que, pour le nombre de visiteurs, la Tour Eiffel soit l'élément le plus emblématique des monuments de Paris, il ne sera vraisemblablement pas le plus représentatif. Dans de futurs travaux, nous nous attacherons donc à la détermination de techniques permettant d'extraire d'autres éléments typiques en fonction de leurs propriétés.

Remerciements

Nous tenons à remercier le Service Régional d'Archéologie de Champagne-Ardenne et le centre rémois de l'Institut National de Recherche en Archéologie Préventice pour nous avoir permis d'accéder à leurs données. Nous tenons de même à souligner la contribution de Dominique Pargny, ingénieur d'études au laboratoire GEGENA, et de Frédéric Piantoni, Maître de Conférences au laboratoire HABITER, au projet SIGRem, porté par l'Université de Reims Champagne-Ardenne, sur lequel se base ce travail.

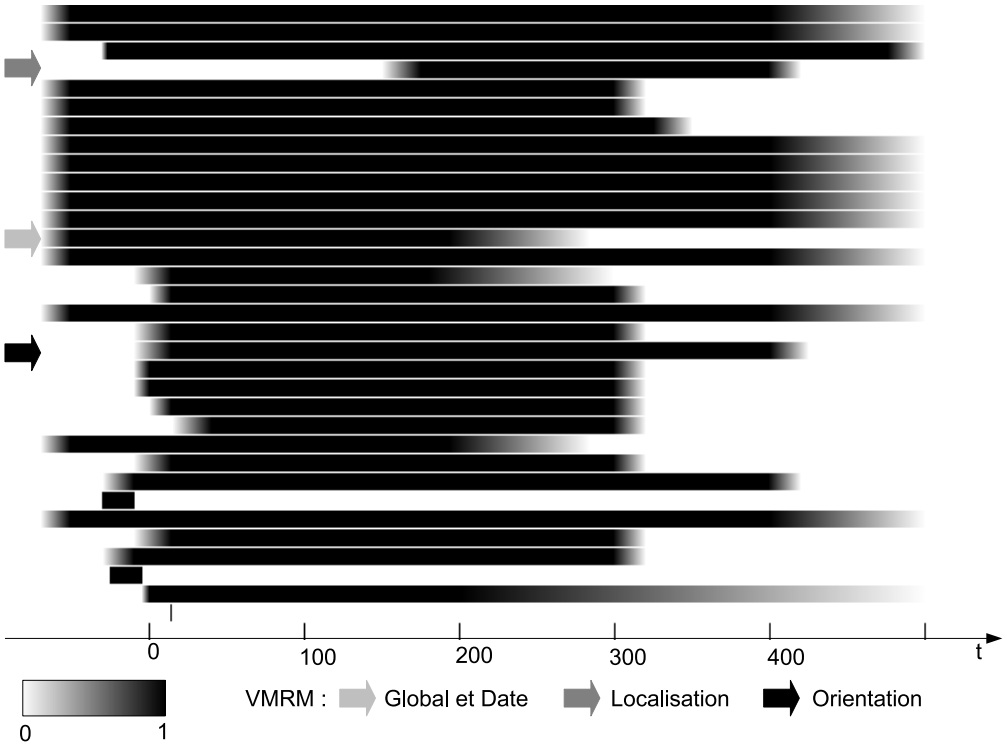


FIG. 4 – *Représentation des périodes d'activités des objets de BDRues*

Références

Barnett, V. (1976). The ordering of multivariate data. *Journal of the Royal Statistical Society, Series A (General) 139*(3), 318–355.

Blanchard, F. (2005). *Visualisation et classification de données multidimensionnelles. Application aux images multicomposantes.* Thèse de doctorat, Université de Reims Champagne-Ardenne, France.

David, H. A. et H. N. Ngaraja (2003). *Order Statistics* (Third ed.). Wiley.

de Runz, C., E. Desjardin, M. Herbin, D. Pargny, F. Piantoni, et F. Berthelot (2007a). Simulation de cartes et prédictivité à partir de données archéologiques traitées par sig. *Culture et Recherche* (111), 35–35.

de Runz, C., E. Desjardin, F. Piantoni, et M. Herbin (2007b). Management of multimodal data using the fuzzy hough transform : Application to archaeological simulation. In *First International Conference on Research Challenges in Information Science*, Maroc, Ouarzazate, pp. 351–356. Colette Rolland, Oscar Pastor and Jean-Louis Cavarero.

de Runz, C., E. Desjardin, F. Piantoni, et M. Herbin (2007c). Using fuzzy logic to manage uncertain multi-modal data in an archaeological gis. In *International Symposium on Spatial Data Quality*, Pays-Bas, Enschede.

de Runz, C., M. Herbin, F. Blanchard, L. Hussenet, V. Vrabie, et P. Vautrot (2007d). Le vecteur de meilleur rang moyen : une statistique pour l'analyse de données multidimensionnelles - application au filtrage d'images couleurs. In *GRETSI*, France, Troyes.

Dragicevic, S. et D. Marceau (2000). An application of fuzzy logic reasoning for gis temporal modeling of dynamic processes. *Fuzzy Sets and Systems 113*, 69–80.

Galambos, J. (1975). Order statistics of samples from multivariate distributions. *Journal of the American Statistical Association 70*(351), 674–680.

Grzegorzewski, P. (1998). Metrics and orders in space of fuzzy numbers. *Fuzzy Sets and Systems 97*, 83–94.

Lukac, R., B. Smolka, K. Plataniotis, et A. Venetsanopoulos (2006). Vector sigma filters for noise detection and removal in color images. *J. Vis. Commun. Image R. 17*, 1–26.

Vautrot, P., L. Hussenet, et M. Herbin (2006). A robust filtering method using owa filters : Application to color images. In *3rd European Conference on Colour in Graphics, Imaging and Vision*, University of Leeds, UK.

Summary

In Geographical Information Systems devoted to archeology, the selection of the most representative element of the Geographical Database is important to increase the value of city excavations. We developed, in the CReSTIC-SIC, a statistical method to select the most representative element of a dataset. This method is based on ranking statistics, but also on a dissimilarity measure between data to determine ranks. We apply it on the extraction of the most representative elements of excavation data. In the GIS about the Roman streets in Reims, the features of the information are modeled by convex and normalized fuzzy sets; hence we use a classical metric between convex and normalized fuzzy sets to measure the dissimilarity. This allows us to extract the most representative Roman street in Reims.

Détection d'intrusions : de l'utilisation de signatures statistiques

Payas Gupta *, Chedy Raïssi **, Gérard Dray **, Pascal Poncelet **, Johan Brissaud ***

*LNMIIT - Jaipur (Raj.) 302015 - India
payasgupta@gmail.com
**LGI2P - École des Mines d'Alès, Parc Scientifique G. Besse, 30035 Nîmes, France
Chedy.Raissi@ema.fr, Gerard.Dray@ema.fr, Pascal.Poncelet@ema.fr
***BeeWare SA
Le Millenium, Bâtiment B, 501 rue Denis Papin 34000 Montpellier
jbrissaud@bee-ware.net

Résumé. Garantir la sécurité des serveurs web devient un enjeu majeur pour les entreprises ou les organisations et il devient de plus en plus difficile de détecter parmi les différentes requêtes celles qui correspondent à un comportement normal de celles qui correspondent à un comportement malveillant. Même s'il existe des Systèmes de Détection d'Intrusions (SDI), ces derniers ne sont malheureusement pas ou plus adaptées aux nouvelles attaques. Motivés par ce constat, les chercheurs de la communauté fouille de données s'intéressent de plus en plus à la détection de fraudes dans les réseaux. Dans cet article nous proposons une nouvelle approche de détection de fraudes basée sur un paramétrage automatique du comportement des requêtes normales et de la distribution de valeurs d'attributs. Les connaissances extraites représentées sous la forme de signatures statistiques peuvent alors être utilisées pour rechercher efficacement des comportements malveillants dans le flot de requêtes. Enfin, de manière à prendre en compte les nouveaux services et minimiser les fausses alarmes, nous maintenons de manière incrémentale les signatures. L'approche proposée a été expérimentée sur des jeux de données réelles et a montré une précision de plus de 99.98% pour la prédiction de requêtes valides et un taux de faux positifs de moins de 9.88%.
Keywords: Détection d'intrusions, signatures statistiques, approche incrémentale.

1 Introduction

Le déploiement des ordinateurs et des réseaux a considérablement augmenté les risques causés par les attaques sur les systèmes informatiques qui deviennent un réel problème pour les entreprises et les organisations. Par exemple, une étude récente du National Institute of Standards and Technology a montré que les dommages, pour les compagnies américaines, étaient estimés à plus de 59,6 millions de dollars par an. Alors qu'auparavant de nombreuses attaques se focalisaient sur les serveurs Web car ils étaient souvent mal configurés ou mal maintenus, les attaques les plus récentes profitent des failles de sécurité des services ou applications Web qui sont plus vulnérables. Pour pallier ce problème, de nouvelles approches appelées Systèmes de Détection d'Intrusions (SDI) ont fait leur apparition. Ces outils, installés sur les réseaux, ont pour objectif d'analyser le trafic de requêtes et de détecter des comportements malveillants. Ils peuvent être classés en deux grandes catégories (*e.g.* McHugh et al. (2000); Proctor (2001)) : les *systèmes de détection d'anomalies* et les *systèmes de détection d'abus*.

Le principe des approches de détection d'abus consiste à appliquer des techniques d'apprentissage sur des attaques connues de manière à en définir leurs signatures. Ensuite, à l'aide d'expressions régulières

ou de correspondance de motifs ces dernières sont utilisées pour reconnaître les attaques dans les flots de requêtes. Si ces approches sont efficaces pour reconnaître les attaques passées, elles sont malheureusement mises en défaut lorsque de nouvelles attaques interviennent et certains logiciels profitent de cette faille pour passer outre ces systèmes de détection Fogla et Lee (2006). D'un autre côté, les systèmes de détection d'anomalies s'intéressent à l'analyse des comportements normaux, *i.e.* les comportements valides sur le site, et cherchent à les caractériser. Ils considèrent ainsi qu'une intrusion correspond à un comportement qui dévie de la norme. Traditionnellement, les motifs sont obtenus à l'aide de techniques d'apprentissages supervisés ou non. Par exemple, dans Giacinto et al. (2006), les auteurs utilisent un ensemble d'apprentissage non étiqueté et classent les requêtes en attaque ou non. Les travaux de Cohen et al. (2004) considèrent un ensemble de données étiquetées "attaque" ou "non attaque" et appliquent des algorithmes de clustering pour déterminer les différentes classes.

Par rapport aux détections d'abus, ces systèmes ont l'avantage d'être moins dépendants des attaques extérieures. Par contre, ils ne sont pas capables de prendre en compte les comportements non prévus ou les évolutions des applications du serveur Web et peuvent ainsi engendrer un grand nombre de fausses alarmes. Pour résoudre ce problème, de nouvelles approches ont été proposées et tentent de maintenir les signatures de manière automatique (*e.g.* Esposito et al.; Li (2005); Yeung et Ding (2003)). Des techniques basées sur la logique floue, les algorithmes génétiques ou les réseaux de neurones sont fréquemment utilisées (*e.g.* de Sá Silva et al. (2007); Saniee et al. (2007); Newsome et al. (2005)) mais, à cause de la structure complexe des requêtes, elles sont généralement difficiles à mettre en œuvre et sont souvent pénalisées par des temps de réponses prohibitifs. Aussi, l'évolution des signatures se résume très souvent à l'intervention d'un expert du domaine et les changements sont lents et coûteux Adeva et Atxa (2007); Paxson (1999).

Il est donc nécessaire de proposer des systèmes qui soient capables d'apprendre automatiquement les signatures des requêtes valides mais qui soient aussi capables de les maintenir afin de réduire le nombre de fausses alarmes Pietraszek et Tanner (2005). Dans cet article, nous proposons un SDI basé sur la méthode de détection d'anomalies. Dans ce contexte, nous souhaitons caractériser et modéliser les comportements typiques de serveurs Web Pereira et al. (2007). Notre objectif est de nous focaliser plus particulièrement sur la détection d'attaques nouvelles et sur les modifications apportées aux anciennes attaques qui ne peuvent pas être détectées par les SDI actuels. Aussi, nous considérons dans la suite de cet article que les anciennes attaques peuvent être traitées par un SDI traditionnel. Etant donné que chaque serveur Web possède ses propres caractéristiques ou ses propres usages, nous déterminons, pour un serveur, les caractéristiques qui correspondent à des requêtes valides, *i.e.* nous modélisons les données valides sous la forme d'attributs qui pourront être utilisés pour générer des signatures. Ces signatures seront alors utilisées pour détecter les comportements anormaux sur le serveur Web. Bien entendu, cette approche peut être utilisée pour détecter les attaques de n'importe quel serveur mais le jeu d'apprentissage d'un serveur ne peut pas être utilisé pour tester les requêtes d'un autre serveur.

Le reste de l'article est organisé de la manière suivante. Dans la section 2 nous discutons les travaux antérieurs. Une présentation générale de notre approche est proposée dans la section 3 et poursuivie par une description des algorithmes dans la section 4. La section 5 décrit quelques unes des expériences réalisées sur des jeux de données réels. La conclusion est proposée dans la section 6.

2 Travaux antérieurs

Notre proposition est proche des méthodes de détections d'anomalies. Le premier SDI basé sur les anomalies a été introduit par D.E. Denning (1987) et de nombreux travaux ont été réalisés dans ce domaine. Les différences essentielles sont liées à la manière de modéliser les données, *i.e.* de caractériser les requêtes valides. Proche de nos préoccupations, des travaux récents se sont intéressés à la détection d'intrusions dans des applications Web. Dans Wagner et Dean (2001), les auteurs montrent comment une

analyse statique des applications Web peut être utilisée pour dériver de manière automatique un modèle du comportement des applications. Cette technique améliore l'approche de Denning (1987) en offrant un haut degré d'automatisation, une protection contre une grande classe d'attaques basées sur des erreurs de programmation, et l'élimination de fausses alarmes. Elle est cependant très dépendante des langages de programmation utilisés pour développer les applications Web. Dans Robertson et al. (2006), les auteurs précisent les limitations des systèmes de détections d'anomalies utilisant des techniques de caractérisation et de généralisation. Ils proposent dans Kruegel et al. (2005), de représenter le comportement des requêtes normales par des attributs. Ces derniers sont générés de manière à distinguer les requêtes valides de requêtes invalides et nécessitent que l'ensemble d'apprentissage soit uniquement composé de requêtes valides. Même si notre proposition utilise également des attributs pour décrire des requêtes normales, elle ne nécessite pas de spécifier des seuils de valeurs qui sont souvent difficiles à déterminer. En outre, à partir de ces attributs nous générons un ensemble de signature réduit à partir duquel nous pouvons analyser les nouvelles requêtes du serveur.

D'autres travaux (*e.g.* Lee et al. (2001)) ont proposé des solutions pour générer en temps réel les signatures mais malheureusement les expérimentations montrent qu'elles produisent un trop grand nombre de fausses alarmes. Pour limiter les effets des fausses alarmes produites par les méthodes d'anomalies de détection, nous introduisons une nouvelle approche qui permet non seulement de générer des signatures de comportement de requêtes valides mais qui en plus permet de maintenir de manière incrémentale ces signatures quand cela est nécessaire. Bien que le modèle présenté dans cet article soit spécifique à des logs de serveur Web, il peut être adapté facilement à d'autres types de comportements inattendus, *e.g.* logs de serveurs Proxy. La seule différence réside dans le choix des attributs qui caractérisent des requêtes valides.

3 Le système de détection d'anomalies

Chaque serveur possède ses propres caractéristiques, *i.e.* les requêtes parvenant à un serveur sont usuellement similaires. Bien entendu, cela peut varier d'un serveur à un autre. En utilisant les caractéristiques d'un serveur Web, cette approche a pour objectif de modéliser les requêtes par différents attributs qui permettent ensuite de générer des signatures statistiques.

3.1 Description des données

Notre approche de détection d'anomalies analyse les requêtes HTTP telles quelles sont stockées par la plupart des serveurs Web (e.g., Apache (2007)). Ces requêtes respectent bien évidemment le format des caractères défini par le RFC 1738 Berners-Lee et al. (1994) : un encodage des caractères est possible à l'aide du caractère % suivi du code ASCII du caractère à coder en notation hexadecimal. Par souci de simplification, dans cet article, nous nous intéressons aux requêtes qui utilisent des arguments pour passer des valeurs aux programmes situés du côté du serveur. Plus formellement, nous considérons par la suite comme entrée du processus un ensemble ordonné d'Uri $U = \{u_1, u_2, ...u_n\}$ où chaque Uri u_i peut être exprimé sous la forme du chemin à la ressource demandée (*path*) et d'une chaîne requête optionnelle (*query*). Cette chaîne est utilisée pour transmettre des valeurs d'arguments à la ressource demandée et elle est identifiée par le caractère ' ?'. Plus formellement, une query q est une liste ordonnée de tuple $< (a_1, v_1), (a_2, v_2), ... (a_n, v_n) >$ où a_i correspond à un nom d'attribut et v_i correspond à la valeur de cet attribut.

L'exemple suivant illustre un exemple d'entrée dans un log simplifié de serveur Web.

192.233.57.105 - jean [15/Sep/2007 :23 :59 :59 - 0800] "GET /scripts/access.cgi ?user=jean&cred=admin" 200 2123

La partie $/scripts/access.cgi$ correspond au path et la partie query est composée de $< (user, jean), (cred, admin) >$.

Nous utiliserons, par la suite, les termes Uri pour désigner la partie *path* (*e.g.*$/scripts/access.cgi$) et *Query* pour la partie valeur des arguments de la requête (*e.g.*$jean, admin$).

3.2 Extraction des Uri et des arguments de la requête

Le principe général de notre système de détection est de détecter à partir d'un ensemble de données valides, les attributs qui peuvent les caractériser et ce d'une manière statistique rapide. Pour chaque requête du fichier log valide, nous séparons la partie Uri de la partie Query.

En outre chaque *Query* est elle-même décomposée en autant de partie qu'il existe d'arguments dans la requête. Etant donné qu'une attaque peut intervenir sur différentes parties d'une requête, notre objectif dans ce prétraitement est de séparer chaque élément de manière à ce qu'il ne puisse pas influencer les autres. Il n'existe donc pas de liaisons entre les éléments et les traitements peuvent donc être réalisés séparément.

3.3 Extraction des attributs

En fait, le choix des attributs est important dans la mesure où ils vont permettre de différencier des requêtes valides de requêtes invalides. Ils doivent donc être choisis pour produire le moins possible de faux positifs ou de faux négatifs. Certains attributs sont calculés sur la chaîne originale et d'autres sur la chaîne décodée. Une chaîne est décodée récursivement jusqu'à ce qu'elle ne puisse plus l'être (C.f. Tableau 1). Ce traitement permet d'aider à identifier des attaques qui utilisent un encodage, double ou de niveau supérieur, pour envoyer des attaques Mangarae et Morganti (2007).

Bien entendu, ces attributs sont spécifiques d'un serveur Web et différents tests sont nécessaires pour en retenir l'ensemble pertinent. Nous présentons ci-dessous quelques-uns des attributs que nous avons retenus pour les expérimentations finales (la sélection ayant été faite avec l'aide d'un expert dans le domaine de la securité informatique) en explicitant les raisons de nos choix.

1. *Nombre de caractères %00 (NULL)* : calculé sur la chaîne originale extraite des parties Uri et Query, cet attribut est utile pour détecter les attaques qui utilise le caractère NULL pour ignorer les caractères suivants. Par exemple, dans la chaîne $location = /etc/passwd\%00dfgdf$ le navigateur ignorera *dfgdf*. Lors des expérimentations, cet attribut absent des requêtes valides apparaissait dans plus de 20% des attaques.
2. *Nombre de caractères codés* : indique, dans la chaîne, le nombre de caractères codés.
3. *Nombre de caractères codés qui ne nécessitent pas d'encodage* : indique les caractères qui sont encodés soit par l'utilisateur soit par le navigateur pour la transmission.
4. *Nombre de caractères codés au moins deux fois*

	Chaîne
Chaîne originale	SELECT%2B%252A% 2BFROM%2B%2560admin%2560
Après le 1^{er} décodage	SELECT+%2A+FRO M+%60admin%60
Après le 2^{nd} décodage	SELECT * FROM 'admin'

TAB. 1 – *Un exemple d'encodage*

Les attributs 2-4 sont calculés sur la chaîne originale, *i.e.* sans décodage. Ils sont très informatifs car ils caractérisent les principes d'encodage utilisés sur un site. Dans un comportement normal le nombre de caractères encodé est traditionnellement faible dans une requête à part pour des caractères régionaux particuliers : *è*, *â*, *é*, *õ* etc.).

	Chaîne
Chaîne Originale	%2573%2565%256C%2 565%2563%2574%2B% 252A%2B%2566%2572 %256F%256D%2B%25 60%2561%256D%2569 %256E%2560
Après le 1^{er} décodage	%73%65%6C%65%63% 74+%2A+%66%72%6F %6D+%60%61%6D%6 9%6E%60
Après le 2^{nd} décodage	select * from 'admin'

TAB. 2 – *Un exemple d'encodage*

Les Tableaux 1 et 2 illustrent les raisons pour lesquelles un grand nombre d'attaques encodent les données. En effet, via cet encodage, les attaques peuvent passer au travers de nombreux SDI qui se limitent au premier ou au second niveau.

Les autres paramètres retenus correspondent principalement aux nombres de caractères spéciaux, d'espaces, de lettres de l'alphabet, de chiffres, ... Ces paramètres sont calculés sur la chaîne décodée.

3.4 Génération des valeurs des attributs

Après l'étape d'extraction des attributs, les valeurs des attributs sont calculées pour chacune des entrées obtenues dans la section 3.2. Ainsi, par exemple, pour la requête suivante :

$/\%63alendrier\%27\%3B\%2A/../../?var = 3649439\%5C4\%7C\%20\%258A8\&res = qsddff$

et après décomposition des différentes parties, nous obtenons :

Uri	$/\%63alendrier\%27\%3B\%2A/../../$	0 4 1 0 1 21 1 1 1 4 0 0 0 2 ... 0 0 0
Query-val$_1$	$3649439\%5C4\%7C\%20\%258A8$	0 4 0 1 2 13 0 0 0 0 1 0 0 0 ... 1 1 0
Query-val$_2$	$qsddff$	0 0 0 0 0 6 0 0 0 0 0 0 0 0 ... 0 0 0

Dans cet exemple, la valeur 4 dans "0 4 0 1 2 13 0 0 0 0 1 0 0 0 ... 1 1 0" correspond à l'attribut "Nombre de caractères codés".

3.5 Génération de la distribution

Après avoir calculé les valeurs de tous les attributs pour chacune des parties de la requête, nous calculons pour chacun des attributs la moyenne (μ) et l'écart type (σ) et ceci de manière séparée pour l'Uri et les valeurs d'arguments de la requête[1]. Ainsi, pour chaque attribut, nous pouvons déterminer son intervalle de validité : de μ-σ à μ+σ comme illustré dans le tableau 3.

3.6 Création des signatures

Après avoir générés les différentes valeurs de distribution pour l'ensemble des données valides, le même jeu de données est réutilisé et pour chaque requête nous examinons si la valeur de chaque attribut de la requête appartient à son intervalle de validité. Si elle appartient à l'intervalle choisi, elle est codée par *1* autrement par *0*. Ce codage correspond à la signature de la requête valide et cette procédure est appliquée à toutes les requêtes du jeux de données (C.f. tableau 4).

Nous ne retenons au final que les signatures uniques qui seront codées sous la forme de vecteurs de bits pour des raisons d'efficacité. A l'aide de ces signatures, les nouvelles requêtes sur le serveur Web peuvent être testées pour rechercher des comportements malveillants, *i.e.* qui n'appartiennent pas à l'ensemble des signatures.

[1]Dans cet article, par souci de simplification, nous considérons que les distributions pour les requêtes non-malveillantes sont normales.

Partie Uri		**Partie Query Argument**	
μ-σ	μ+σ	μ-σ	μ+σ
0.1540	0.1658	-0.5233	0.5371
-0.0032	0.0032	-0.1316	0.1334
-0.0007	0.0007	-0.2104	0.2136
-0.0413	0.0451	-0.0278	0.029
22.4660	49.2036	-1.6920	19.1362

TAB. 3 – *Un exemple d'intervalle de validité pour les 5 premiers attributs.*

00110110101111...1100
10110101111110...1110
10110101111111...1100
10110101111110...0000
10110101111111...0100
10110101111111...1000
11111111111111...1111

TAB. 4 – *Un exemple de signatures valides*

3.7 Analyse des signatures

La génération des signatures est très importante car elles caractérisent les requêtes normales dans le modèle. Elles montrent que certains paramètres des requêtes n'appartiennent pas à l'intervalle de distribution et sont donc codées avec *0*. Ceci n'implique cependant pas que la requête soit invalide. Il n'est pas nécessaire pour les valeurs de tous les paramètres d'une requête valide qu'elles apparaissent complètement dans leur intervalle. Supposons que nous n'ayons pas de signatures et que nous calculions simplement l'intervalle. Ainsi, lorsqu'une nouvelle requête arrive nous calculons ses valeurs d'attributs et vérifions si elles appartiennent ou non à la distribution. Si elles n'appartiennent pas à l'intervalle de distribution, nous augmentons son poids. Cependant comme nous l'avons précisé précédemment, il n'est pas nécessaire pour une requête valide de toujours satisfaire tous les intervalles des paramètres. Dans ce cas, cela produirait de nombreuses fausses alarmes car nous ne donnons pas d'importance à ces valeurs qui sont hors de l'intervalle. Ces signatures sont appelées Signatures Statistiques car, basées sur la distributions de la valeur des attributs, elles sont générées et représentent les caractéristiques du serveur Web.

4 Algorithmes

Le processus général est divisé en deux phases principales : la *phase de paramétrage* et la *phase de test*. Au cours de la première phase, le SDI est exécuté avec des requêtes valides et les signatures correspondantes au serveur Web sont générées. La phase de test du SDI examine toutes les nouvelles requêtes venant du serveur Web et sépare les requêtes valides des invalides. Une alarme est déclenchée quand le système détecte une requête supposée être une attaque ou une requête invalide par rapport aux comportements modélisés. La troisième étape de notre approche correspond à la maintenance des signatures.

4.1 Phase de paramétrage

La phase de paramétrage est une étape importante pour chaque SDI car les résultats obtenus dans la phase de test dépendent de la manière dont le modèle du SDI a été construit et du type de données utilisées. Comme nous l'avons vu dans les sections précédentes, nous réduisons les requêtes aux parties Uri et Query aussi l'algorithme ne considère que ces deux composantes mais peut aisément être étendu à tous les champs de la partie entête de la requête. Considérons l'ensemble L composé de requêtes valides R. μ_i et σ_i sont les moyennes et écarts types pour le i^{eme} attribut de toutes les requêtes de la phase de paramétrage. Afin de s'adapter à un principe de paramétrage *on-line*, les valeurs μ_i et σ_i sont obtenus de manière incrémentale à l'aide des équations 1 et 2.

$$\mu_1 = x_1, \qquad \mu_{k+1} = \frac{k}{k+1}\mu_k + \frac{1}{k+1}x_{k+1} \tag{1}$$

$$\sigma_1 = 0, \qquad \sigma_{k+1} = \sqrt{\frac{k}{k+1}\sigma_k^2 + \frac{k}{(k+1)^2}(\mu_k - x_{k+1})^2} \tag{2}$$

Dans la Fonction 1, à chaque fois qu'une nouvelle requête est traitée, les valeurs correspondantes aux attributs sont calculées et les μ et σ associés sont mises à jour. Ceci est généralisé à toutes les requêtes R de l'ensemble des données valides L, *compute_signature(type)* est utilisé pour générer, de manière séparée pour les Uri et les Query, les signatures.

Fonction `1 Parametering_phase(`L`)`

Data : L=Ensemble valide de requêtes R.
Result : Signatures pour Uri et Query.
begin
- **for** $i \leftarrow 1$ **to** $no\_of\_attributes$ **do**
  - $\mu_i \longleftarrow 0, \sigma_i \longleftarrow 0$
- **while** $\exists R \in L$ **do**
  - **if** $Uri \in R$ **then**
    - U $\longleftarrow$ extract Uri
    - compute attribute values
    - compute_mu_sigma(U)
  - **if** $Query \in R$ **then**
    - Q $\longleftarrow$ extract value from the query arg
    - **while** $\exists Q$ **do**
      - compute attribute values
      - compute_mu_sigma(Q)
- compute_signature(U)
- compute_signature(Q)

end

4.2 Phase de test

Dans cette phase, la Fonction 2 extrait de la nouvelle requête, les composantes Uri et Query et la valeur de tous les attributs est calculée. Si la valeur d'un attribut appartient à son intervalle calculé dans la phase de paramétrage, *i.e.* (μ-$\sigma \leq$ val $\leq \mu$+σ), il est codé *1* autrement *0*. Ce processus est répété pour tous les attributs et cela forme la signature de la partie Uri et Query.

A la fois pour les parties Uri et Query, la nouvelle signature est comparée avec l'ensemble des autres signatures obtenues lors de la phase de paramétrage. Pour chaque signature, le nombre de bits différent est compté et la somme est divisée par le nombre d'attributs. Ce nombre correspond au poids ($0 \leq$ weight ≤ 1) donné à la requête pour une signature. Ceci est réalisé pour toutes les signatures issues de la phase de paramétrage. Finalement, le minimum de tous les poids est calculé. Dans le cas de l'Uri ce nombre minimum montre de combien la nouvelle signature de l'Uri est différente des signatures statistiques. Par contre, dans le cas de la composante Query, cette procédure est répétée pour chaque argument et le maximum de tous les poids minimum est calculé à chaque fois pour décrire la différence par rapport aux signatures statistiques.

Poids de la partie Uri	**Poids de la partie Query**
0.166667	
0.166667	
0.000000	
0.000000	0.000000
0.000000	0.055556

TAB. 5 – *Un exemple de tableau de valeurs de poids*

De manière à illustrer pourquoi nous retenons le maximum de tous les poids minimum dans la composante Query, considérons le cas suivant. La composante Query contient trois arguments, les deux premiers sont valides et le dernier correspond à une attaque. Pour chacun des arguments, nous recherchons la correspondance la plus proche par rapport aux signatures statistiques et nous calculons le poids associé. Pour les deux premiers arguments, ce poids sera de 0 (correspondance parfaite) et pour le troisième, le poids aura une valeur supérieure à 0. Ainsi, quand nous considérons le maximum de tous les poids nous savons exactement dans quelle partie de la requête l'attaque a eu lieu. Le tableau 5 illustre les poids obtenus pour cinq requêtes. Si la requête ne contient pas de paramètres alors la valeur est évidemment nulle, *e.g.*les trois premières requêtes dans le tableau. Pour chaque requête, s'il existe un poids supérieur à zéro alors cela indique que la requête complète ne correspond pas aux signatures qui ont été générées et il peut s'agir d'une attaque.

4.3 Mise à jour des signatures

Comme nous pouvons le constater à la fin de la Fonction 2, l'approche peut être implémentée de manière incrémentale et les signatures peuvent être mises à jour de manière très efficace. Pour maintenir les connaissances de manière incrémentale, les valeurs des attributs et les μ et σ de tous les attributs sont stockés séparément. Quand une nouvelle requête arrive, ses valeurs d'attributs ainsi que les poids correspondants sont calculés. Nous avons vu dans la section précédente que les valeurs de poids supérieures à zéro pouvaient correspondre à des requêtes d'attaque et qu'une alarme était déclenchée. Si la requête est réellement une attaque, il s'agit alors d'une véritable détection. Par contre, s'il s'agit d'une requête valide, alors il s'agit d'une fausse alarme et les signatures doivent être mises à jour.

Pour cela, les valeurs de tous les attributs de la nouvelle requête *NR* sont ajoutées aux valeurs précédemment obtenus lors de la phase de paramétrage *R*. Les nouvelles valeurs de moyenne et d'écart type sont calculées en utilisant les équations 1 et 2. Les signatures de ce nouvel ensemble sont alors générées. Ces dernières peuvent alors remplacer les anciennes signatures. Ce processus de mis à jour étant effectué en arrière plan, il n'a pas d'impact sur les performances du traitement des nouvelles requêtes. En outre, étant donné que les signatures sont exprimées sous la forme de vecteurs de bits, le remplacement des anciennes par les nouvelles peut être également effectué sans pénaliser le traitement des nouvelles requêtes.

Fonction `2 Testing_Phase-comp_newrequest`

Data : Ensemble de requêtes de test R.
Result : Alerte pour les requêtes attaques **OU** invalides
begin
  compute attributes values for Uri and Query arg.
  if $\exists\, Uri$ **then**
    compute signature $NR\_signature$ for Uri
  $w_{min} \longleftarrow 1$
  while $\forall$ $signature\ S \in Uri\ signature\ file$ **do**
    $w \longleftarrow NR\_signature\ \&\&\ S$
    if $(w/no\_of\_attributes) < w_{min}$ **then**
      $w_{min} \longleftarrow w/no\_of\_attributes$
    if $w = 0$ **then** break
  $w_{max} \longleftarrow 0$; $w_{min} \longleftarrow 1$
  if $\exists\ query\_arg$ **then**
    compute signature NR_sig for Query arguments
  while $\forall\ query\ arguments$ **do**
    while $\forall\ signature\ S \in query\_arguments\ signature\ file$ **do**
      $w \longleftarrow NR\_signature\ \&\&\ S$
      if $(w/no\_of\_attributes) < w_{min}$ **then**
        $w_{min} \longleftarrow w/no\_of\_attributes$
      if $w = 0$ **then** break
    if $w_{max} < w_{min}$ **then** $w_{max} \longleftarrow w_{min}$
  if w_{min} *of Uri* $\neq 0$ *OR* w_{max} *of Query* $\neq 0$ **then**
    alert Attack
    if $false\ alarm$ **then**
      if $w_{min}\ of\ Uri > 0$ **then**
        recomp_signature(U)
      if $w_{max}\ of\ Qu\_arguments > 0$ **then**
        recomp_signature(Q)
  else
    $Valid$
end

5 Experimentations

De manière à évaluer la précision de notre SDI pour détecter de nouvelles attaques et le temps nécessaire à la mise à jour des signatures lors de la présence de fausses alarmes, différentes expérimentations ont été menées sur des logs de données réelles issus de la société Beeware et de l'Ecole des Mines d'Alès. Les résultats obtenus étant assez similaires, par manque de place, nous donnons ci-dessous les résultats pour un log composé de **2173831** requêtes valides et **5603** attaques.

5.1 Précision de la détection du SDI

Après avoir appris **1740978** requêtes de l'ensemble de données valides et calculé les signatures, nous obtenons **504497** arguments pour la partie Query et **1740978** Uri pour le jeu d'apprentissage. Le tableau 5.1 décrit la matrice de confusion obtenue pour un test avec **5603** requêtes invalides et **432853** requêtes valides. Ces résultats sont obtenus sans mettre à jour les signatures.

	Valide	**Attaque**
Prédit Valide	432821	*554*
Prédit Attaque	*32*	5044

TAB. 6 – *Matrice de confusion*

La matrice montre que nous avons un taux de vraie détection de requêtes valides de **99.998%** et que le taux de détection d'attaques est de **90.08%**.

5.2 Combien de requêtes doivent être apprises ?

Dans cette expérimentation, nous considérons **23367** requêtes pour l'apprentissage. Mais dans ce cas, nous apprenons une requête et ensuite nous la testons avec **3550** attaques et **7618** requêtes valides. A chaque fois, les signatures sont mises à jour lorsque la nouvelle requête est apprise. Le but de cette expérimentation est d'estimer le nombre de requêtes qu'il est nécessaire d'apprendre avant d'utiliser cette approche sur le serveur. La Figure 1 illustre quelques résultats. Nous pouvons constater que le nombre de fausses alarmes devient constant après un apprentissage de $\approx$ **7500** requêtes. Cette expérience montre qu'en fait il n'est pas obligatoire d'avoir un grand nombre de requêtes pour l'apprentissage. La procédure de mise à jour des signatures nécessite **6.82** secondes après avoir appris **23367** requêtes.

6 Conclusion

Dans cet article, nous avons proposé une nouvelle approche, utilisant les logs générés par les applications Web, pour détecter les tentatives d'intrusion sur un serveur Web. L'approche proposée modélise, à partir d'un ensemble de requêtes valides, les comportements "normaux" sur le serveur en fonction de différents attributs et génère des signatures statistiques utilisées pour détecter les tentatives d'intrusion. Elle ne nécessite pas de spécifier des valeurs de seuil pour les attributs. Elle peut également mettre à jour les signatures pour prendre en compte les évolutions du site. Cette technique a été développée pour détecter les nouvelles attaques plutôt que celles qui sont anciennes ou connues et qui peuvent être détectées par de nombreux SDI. Les expérimentations que nous avons menées ont montré que nous étions capables de reconnaître 99.98% des requêtes valides et plus de 90% de requêtes invalides.

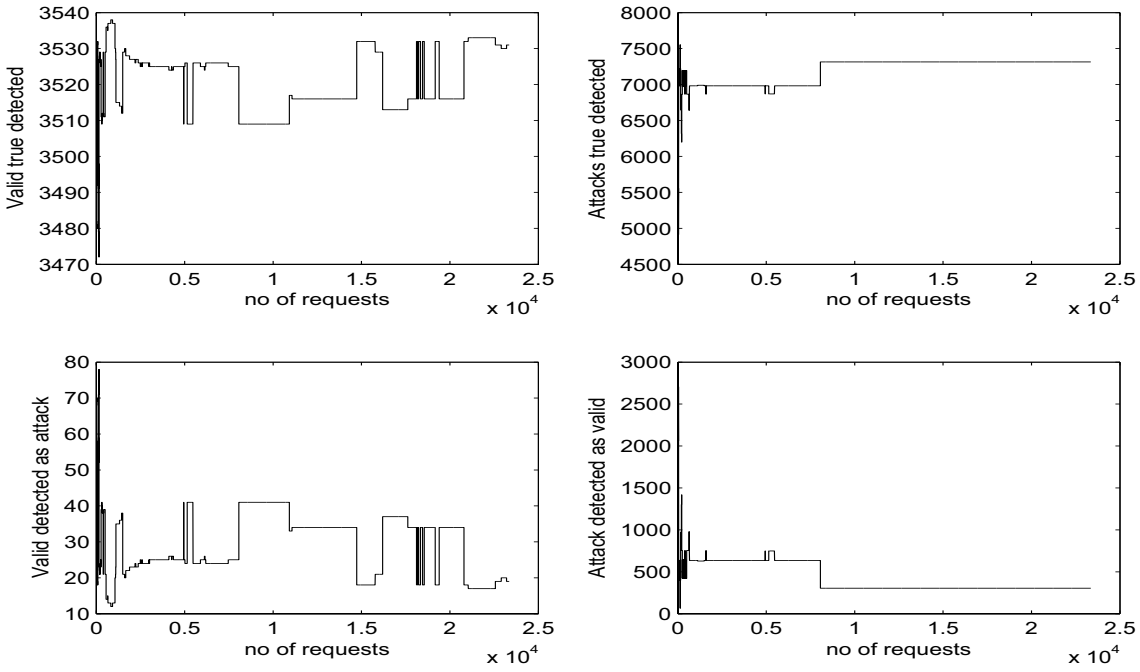


FIG. 1 – *Mise à jour des signatures après apprentissage des requêtes*

Remerciements

Ce travail, réalisé dans le cadre d'un projet de transfert de technologie financé par la région Languedoc Roussillon, résulte d'une collaboration avec l'entreprise BeeWare SA (http ://www.bee-ware.net) qui en exploitera industriellement les résultats.

Références

Adeva, J. J. G. et J. M. P. Atxa (2007). Intrusion detection in web applications using text mining. *Engineering Applications of Artificial Intelligence 20*(4), 555–566.

Apache, D. (2007). http ://httpd.apache.org/docs/.

Berners-Lee, T., L. Masinter, et M. McCahill (1994). Uniform resource locators (url), http ://www.ietf.org/rfc/rfc1738.txt. *http ://www.ietf.org/rfc/rfc1738.txt*.

Cohen, I., F. G. Cozman, N. Sebe, M. C. Cirelo, et T. S. Huang (2004). Semi-supervised learning of classifiers : Theory, algorithms and their application to human-computer interaction.

de Sá Silva, L., A. C. F. dos Santos, T. D. Mancilha, J. D. da Silva Simões, et A. Montes (2007). Detecting attack signatures in the real network with annida. *Elsevier Ltd*.

Denning, D. (1987). An intrusion-detection model. *IEEE Transactions on Software Engineering 13*(2), 222–232.

Esposito, M., C. Mazzariello, F. Oliviero, S. Romano, et C. Sansone. Evaluating pattern recognition techniques in intrusion detection systems.

Fogla, P. et W. Lee (2006). Evading network anomaly detection systems : formal reasoning and practical techniques. *Proceedings of the 13th ACM conference on Computer and communications security*, 59–68.

Giacinto, G., R. Perdisci, M. D. Ri, et F. Roli (2006). Intrusion detection in computer networks by a modular ensemble of one-class classifier.

Kruegel, C., G. Vigna, et W. Robertson (2005). A multi-model approach to the detection of web-based attacks. *Computer Networks : The International Journal of Computer and Telecommunications Networking 48*(5), 717–738.

Lee, W., S. J. Stolfo, P. K. Chan, E. Eskin, W. Fan, M. Miller, S. Hershkop, et J. Zhang (2001). Real time data mining-based intrusion detection. *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01. Proceedings 1*(1), 89–100.

Li, X.-B. (2005). A scalable decision tree system and its application in pattern recognition and intrusion detection. *Decision Support Systems 41*(1), 112–130.

Mangarae, A. et C. Morganti (2007). https ://www.securinfos.info/english/security-papers-hacking-whitepapers.php.

McHugh, J., A. Christie, et J. Allen (2000). Defending yourself : the role of intrusion detection systems. *IEEE Software*, 42–51.

Newsome, J., B. Karp, et D. Song (2005). Polygraph : Automatically generating signatures for polymorphic worms. *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, 226–241.

Paxson, V. (1999). Bro : a system for detecting network intruders in real-time. *Computer Networks 31*(23-24), 2435–2463.

Pereira, A., G. Franco, L. Silva, et W. M. Jr. (2007). A hierarchical characterization of user behavior. *Proceedings of the WebMedia & LA-Web 2004 Joint Conference 10th Brazilian Symposium on Multimedia and the Web 2nd Latin American Web Congress 00*, 2–9.

Pietraszek, T. et A. Tanner (2005). Data mining and machine learning-towards reducing false positives in intrusion detection. *Elsevier Ltd*.

Proctor, P. (2001). *Practical Intrusion Detection Handbook*. Upper Saddle River, NJ : Prentice-Hall.

Robertson, W., G. Vigna, C. Kruegel, et R. A. Kemmerer (2006). Using generalization and characterization techniques in the anomaly-based detection of web attacks. *Proceedings of Network and Distributed System Security Symposium Conference, Internet Society, 2006*.

Saniee, M., J. Habibi, Z. Barzegar, et M. Sergi (2007). A parallel genetic local search algorithm for intrusion detection in computer networks. *CSICC 2006*.

Wagner, D. et D. Dean (2001). Intrusion detection via static analysis. *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, 156.

Yeung, D.-Y. et Y. Ding (2003). Host-based intrusion detection using dynamic and static behavioral models. *Pattern Recognition 36*(1), 229–243.

Summary

Security of web servers have become a sensitive subject today. Prediction of normal and abnormal requests is hard due to generation of large number of false alarms in many anomaly based IDS. In this paper, we introduce a novel intrusion detection approach using incremental calculation of statistical signatures based on the modelling of normal requests and their distribution value without explicit intervention. Experiments conducted on real datasets have shown high accuracy up to 99.98% for predicting valid request as valid and false positive rate as low as 9.88%.