EGC 2004

Clermont-Ferrand
20 janvier 2004

Atelier n°6

Fouille de Données Complexes dans un processus d'extraction de connaissances

Pierre Gançarski (AFD-LSIIT, Strasbourg)
et
Brigitte Trousse (AxIS-INRIA, Sophia Antipolis)

Responsable Cours & Ateliers :
Mohand-Saïd HACID, LIRIS, LYON

Premier atelier sur la
"Fouille de données complexes dans un processus d'extraction des connaissances"

Clermont-Ferrand, France
20 Janvier 2004

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 " Fouille de données complexes " et s'inscrit dans le cadre de la conférence EGC04 qui se tient à Clermont Ferrand.

Cet atelier se veut être un lieu de rencontre que nous souhaitons annuel où chercheurs/industriels peuvent partager leurs expériences et expertises dans le domaine de la fouille de données complexes i.e. des données non structurées comme c'est le cas dans le web, les séquences vidéo, etc.

L'atelier se veut ouvert. On y trouvera 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). Nous souhaitons des discussions riches sur les liens entre différentes disciplines.

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.

Ainsi est-on amené à devoir manipuler des données souvent non structurées

- Issues de diverses provenances comme des capteurs ou sources physiques d'informations variées ;
- Représentant la même information à des dates différentes ;
- Regroupant différents types d'informations (images, textes) ou encore de différentes natures (logs, contenu de documents, ontologies, etc.).

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.

La liste de thèmes pouvant être abordés dans cet atelier :

- Pré traitement, structuration et organisation des données complexes :
  - Enrichissement des données (données inférées, connaissances, etc.)

- Sélection, nettoyage des données
- Codage, transformation des données
- Intégration de données
- Conception d'entrepôts de données
- Modélisation des données complexes, méta-données, XML …

• Processus et méthodes de fouille de données complexes :
- Evaluation des méthodes actuelles
- Proposition d'approches nouvelles (par exemple hybrides ou multi-stratégies)
- Sélection de sources des données et d'attributs
- Utilisation de relations spatiales ou temporelles entre les données
- Utilisation de connaissances du domaine pour optimiser l'extraction…

• Retours d'expériences d'extraction de connaissances à partir de données complexes (Web, sciences du vivant, etc.).

Déroulement de la journée

L'atelier est constitué d'une série d'exposés (présentations orales courtes ou longues).

Une présentation des deux thèmes de travail « Fouille de données complexes » est prévue à partir de 16h, suivie d'une discussion plus générale sur les actions à entreprendre au niveau du groupe de travail.

Responsables : Gançarski Pierre (AFD-LSIIT) et Trousse Brigitte (Axis-Inria) .

Les responsables tiennent à remercier vivement :
- Les auteurs pour leur contribution
- Les membres du comité de lecture pour leur travail scientifique effectué dans les délais très courts ainsi qu'aux trois relecteurs additionnels : Fabrice Rossi (AxIS INRIA), Lionel Savary (Prism-Université de Versailles) et Eric Lefevre (LGI2A – Université de l'Artois)
- Mohand-Saïd Hacid, responsable des ateliers à EGC 2004 et Jean-Marc Petit, président du comité d'organisation, pour leur aide et leur disponibilité pour la préparation de cet atelier
- Ainsi que Sophie Honnorat (Axis INRIA) pour son soutien dans l'organisation.

Comité de lecture

Le comité de lecture est composé d'un représentant par laboratoire membre du GT "Fouilles de Données Complexes" et d'experts du domaine.

- Aufaure Marie-Aude (Supélec)
- Bennani Younès (LIPN)
- Bouet Marinette (LIMOS)
- Bounekkar Ahmed (LASS)
- Boussaïd Omar (ERIC)
- Diday Edwin (CEREMADE)
- Elfaouzi Nour-Eddin (INRETS)
- Frélicot Carl (L3I)
- Gallinari Patrick (LIP6)
- Martin Lionel (LIFO)
- Masséglia Florent (INRIA)
- Morin Annie (IRISA)
- Lebart Ludovic (CNRS et ENST)
- Lechevallier Yves (INRIA)
- Napoli Amedeo (LORIA)
- Nugier Sylvaine (EDF)
- Saidi-Glandus Alexandre (LIRIS)
- Tommasi Marc (GRAPPA)
- Tourcel Bernard (LIFL)
- Wemmert Cédric (LSIIT)
- Zeitouni Karine (PRISM)
- Mohand-Saïd Hacid (LIRIS)
- Zighed Djamel (ERIC)

Programme

8h30 – 8h45	**Accueil des participants**
8h45 – 9h00	**Ouverture de l'atelier**
9h00 – 12h30	**Communications orales - Session 1**
9h00 - 9h25	Extension de l'algèbre relationnelle aux données symboliques *Tao Wang - Karine Zeitouni*
9h25 - 9h50	Sélection automatique d'index dans les entrepôts de données *Kamel Aouiche - Jérôme Darmon - Omar Boussaïd*
9h50 - 10h15	Extraction de connaissances provenant de données multisources pour la caractérisation d'arythmies cardiaques *Elisa Fromont - Marie-Odile Cordier - René Quiniou*
10h15 - 10h25	Classification multisource via la théorie des croyances *Anis Ben Aïssa - Nour-Eddin El Faouzi - Eric Lefevre*
10h30 - 11h00	*Pause*
11h00 - 11h25	Intégration des connaissances du domaine pour la fouille de données complexes *Walid Ben Admed - Mounib Mekhilef - Michel Bigand - Yves Pages*
11h25 - 11h50	Classification automatique à partir de données logs Web et de connaissances sur le site *Mireille Arnoux - Yves Lechevallier -Doru Tanasa - Brigitte Trousse -*
11h50 - 12h15	Recherche par le contenu dans une base d'images fixes : l'intérêt des règles d'association
12h15 - 12h25	Tableau de bits indexé pour la recherche de séquences fréquentes - Application à l'enquête MENAGE *Lionel Savary - Karine Zeitouni*
12h30 - 14h	***Déjeuner***

14h - 15h30	**Communications orales - Session 2**
14h00 - 14h25	Extraction de classes homogènes et création d'objets symboliques *Aïcha El Golli - Yves Lechevallier*
14h25 - 14h50	Application aux images hyperspectrales d'une nouvelle méthode de sélection d'attributs pour la classification d'objets complexes *Alexandre Blansché - Pierre Gançarski*
14h50 - 15h00	DISDAMIN : Algorithmes de data mining distribués *Valérie Fiolet - Bernard Toursel*
15h00 -15h10	Pertinence des métriques fractionnaires pour l'analyse des données de grande dimension (signature génomique) *Sylvain Lespinats - Patrick Deschavanne - Alain Giron - Bernard Fertil*
15h10 - 15h20	Visualisation avec les cartes topologiques catégorielles *Mustapha Lebbah - Fouad Badran- Sylvie Thiria*
15h20 - 15h30	Détection de motifs dans des images médicales par apprentissage supervisé *Arthur Tennehaus - Maxime Saporta - Alain Giron - Bernard Fertil*
15h30 - 15h40	SOM-ART : Incorporation des propriétés de plasticité et de stabilité dans une carte auto-organisatrice *Farida Zehraoui - Younès Bennani*
15h30 - 16h00	*Pause*
16h00 -17h30	**Rencontre entre les participants du groupe de travail FDC** ***Animateur : Djamel Zighed***
	Présentation des deux thèmes du groupe
	Structuration et organisation des données complexes (30 mn) Animateurs : Omar Boussaid et Florent Masséglia
	Classification d'objets issus de plusieurs sources de données (30 mn) Animateur : Pierre Gançarski
	Discussion sur les actions futures du groupe
17h30-18h30	**Cocktail**

Table des matières

Extension de l'algèbre relationnelle aux données symbolique

Tao Wan, Karine Zeitouni

Laboratoire PRISM
Université de Versailles
{Tao.Wan, Karine.Zeitouni}@prism.uvsq.fr

Résumé. De nombreuses applications nécessitent la gestion de données non atomiques, c'est à dire qui ne sont pas de simples valeurs numériques ou catégorielles. Les données symboliques constituent une piste intéressante pour la gestion de telles applications. Seulement, les travaux existants sont davantage orientés vers l'extension des méthodes statistiques et d'analyses à ce type de données que pour l'interrogation et la manipulation. Cet article propose une extension de l'algèbre relationnelle aux données symboliques. L'essentiel de cette extension est de redéfinir les restrictions et les jointures avec des prédicats flous. En effet, la comparaison exacte entre données symboliques complexes provoquerait souvent des réponses vides. Des requêtes complexes peuvent être formulées en utilisant des liens logiques dont la sémantique est** paramétrée pour s'adapter à des scénarios spécifiques.

1. Introduction

1.1. Motivation de l'extension de l'algèbre relationnelle aux données symboliques

De nombreuses applications nécessitent la gestion de données non atomiques, c'est à dire qui ne sont pas de simples valeurs numériques ou catégorielles. C'est le cas de l'analyse des risques du transport où l'on gère des mesures de distributions (de pollution par exemple), ou des intervalles (temporels) et enfin des ensembles ou des séquences de trajets pour la description des déplacements de la population [2]. Les données symboliques constituent une piste intéressante pour la gestion de telles applications. Par ailleurs, l'analyse exploratoire des bases de données ainsi produites est la clé de voûte de la compréhension du phénomène étudié.

L'OLAP constitue un outil d'analyse exploratoire efficace pour des données relationnelles. Nous nous proposons d'étudier l'extension des techniques OLAP vers les données symboliques. Notre recherche à terme vise la génération d'entrepôts de données symboliques, le prolongement des fonctions OLAP et des cubes de données vers ces nouveaux types et l'extraction de résumés symboliques. Mais auparavant, les opérateurs de base, tels que la sélection ou la jointure, doivent être définies. Il n'existe pas, à notre connaissance, de travaux sur l'interrogation et la manipulation des données symboliques. Par conséquent, le début de notre travail vise à étendre l'algèbre relationnelle aux données symboliques.

1.2. Introduction aux données symboliques

Nouveau paradigme de description de données, les données symboliques permettent de définir des structures de données complexes. Celles-ci représentent les variations internes telles que des distributions ou des intervalles et sont liées par des règles de dépendances ou des taxonomies [1]. Ainsi, elles sont plus adaptées que les données atomiques pour représenter des observations dont les attributs sont naturellement multi-valués ou définissant des distributions. On parle de données symboliques natives.

Un autre intérêt des données symboliques est de permettre d'extraire et de résumer des objets de grandes bases de données en concepts de plus haut niveau afin de mieux les appréhender et d'en extraire de nouvelles connaissances. Ces concepts sont alors décrits par des valeurs symboliques. Par comparaison au modèle relationnel, ces valeurs symboliques sont comparables aux agrégats, sauf qu'au lieu de retourner une valeur simple, ils résument de façon plus riche les données en multi-valeurs, en

histogrammes ou en intervalles. La figure 1 montre un exemple de comparaison entre agrégation classique des données relationnelles et génération des données symboliques à partir de la même table.

Id	Couleur	Matériel	Température
1	Rouge	Fer	30°
1	Bleu	Cuivre	30°
1	Rouge	Cuivre	30°
2	Bleu	Cuivre	45°
2	Bleu	Cuivre	25°
3	Vert	Bronze	39°
3	Bleu	Fer	39°

Table1 : Une table relationnelle

Select Id, MostFrequent(Couleur), MostFrequent(Matériel), Min(Température), From Table1
Group by Id;

Id	Couleur	Matériel	Température
1	Rouge	Cuivre	30°
2	Bleu	Cuivre	25°
3	Vert	Bronze	39°

Table 2 : Agrégation classique de la table 1 basée sur Id de la table 1

Id	Couleur	Matériel	Température
1	Rouge, Bleu	Fer (33%), Cuivre (67%)	30°
2	Bleu	Cuivre (100%)	[25°, 45°]
3	Vert, Bleu	Bronze (50%), Fer (50%)	39°

Table 3 : Génération d'une table symbolique basée sur Id de la table 1

Figure 1 : Comparaison entre agrégation classique des données relationnelles et génération des données symboliques

1.3. Spécificités

Les principaux types des données symboliques sont les intervalles, les multi-valeurs et les données multi-valuées pondérées dites distributions. Les multi-valeurs peuvent être traitées comme un cas particulier des distributions où les poids sont identiques. Au vu des applications réelles, le résumé par intervalle est moins informatif. C'est pourquoi, nous nous intéressons d'abord à l'extension de l'algèbre relationnelle [14] aux données symboliques de type distribution.

Le problème de l'interrogation des données symboliques de type de distribution est que leur comparaison par égalité est rarement satisfaite. En effet, elle consiste à comparer leur liste des valeurs et précisément l'égalité de la distribution de chaque valeur. Ainsi, des distributions peuvent être différentes tout en étant très similaires. Pour éviter ce problème, nous intégrons le concept de requêtes floues [3] où la satisfaction d'un critère peut avoir un certain degré de liberté. Cela veut dire que la mesure d'une requête est comprise entre 0 et 1. Une requête floue est une composition logique de prédicats flous.

Le reste du papier est organisé comme suit. La section II présente des travaux connexes sur l'extension de l'algèbre aux données possibilistes et floues. La section III donne une formalisation des concepts de bases préalables à la définition de l'algèbre symbolique, tels que les requêtes floues et les mesures de similarité. La contribution principale du papier, à savoir l'approche que nous proposons pour étendre l'algèbre relationnelle aux données symboliques, est présentée en section IV. Des perspectives de ces travaux sont dégagées dans la section V.

2. Travaux connexes

La représentation et la manipulation des distributions est un thème qui a été abordé dans des travaux sur les bases de données probabilistes et floues ainsi que dans le domaine des bases multimédias.

La théorie des possibilités [6] offre un cadre unifié pour représenter les valeurs précises et les valeurs imprécises telles que les intervalles usuels ou flous et les valeurs nulles. Voir [3] pour plus de détail. Ainsi, les données contenant des informations imprécises peuvent être modélisées par des bases de données possibilistes, où une valeur d'attribut mal connue est représentée par une distribution de possibilités. Au delà de la description, la manipulation de ce type de données a fait l'objet de travaux dans les années 80. Ceux là ont posé quelques bases, particulièrement pour les requêtes contenant des opérations simples, à savoir la sélection et la projection [7]. Toutefois, un langage de requêtes restreint à la sélection et à la projection n'est guère satisfaisant dans la mesure où il ne permet de manipuler qu'une relation et a donc un pourvoir d'expression limité. Dans [4, 5], un nouveau type de requêtes, appelée requêtes possibilistes, de la forme : "dans quelle mesure est-il possible que le tuple t appartienne à la réponse à la requête Q" a été étudié. Il s'agit donc d'une extension des questions booléenne (appartenance oui/non à la réponse), à une mesure de satisfaction de la requête. Cette extension des requêtes à la logique possibiliste [8, 9, 10] s'adapte bien aux types de données possibilistes. Néanmoins, la forme de données possibilistes selon la définition[1] dans [7] ressemble, mais ne correspond pas tout à fait aux types de données multi-valuées pondérées telles que nous les considérons. Donc nous ne pouvons pas emprunter directement l'approche de l'algèbre étendue définie dans [3, 4, 5].

Dans des bases de données multimédias, certains types d'attributs contiennent des données complexes, telles qu'un histogramme de couleurs, un vecteur de texture, etc. Le type de données histogramme est très semblable aux distributions de probabilités. Pour ce genre de bases de données, la plupart des contributions proviennent du domaine de Recherche d'information (Information Retrieval) [11, 12] sans préoccupation de la manipulation des données. Dans [13], un langage de requêtes floues est proposé, il accepte des données floues aux niveaux des attributs et des tuples. Cependant chaque cellule ne contient qu'une valeur pondérée.

Au vu de ces travaux, il n'existe pas actuellement de proposition d'algèbre de manipulation de données multi-valuées pondérées.

3. Concepts de base

Dans cette section, les concepts de base de notre algèbre étendue sont présentés.

Notations :

Soit R une table de données symboliques, soit $X = \{A_1, \ldots, A_n\}$ un sous-ensemble d'attributs symboliques. On note dom (A_i) le domaine de valeurs A_i. La valeur de l'attribut A d'un tuple t est notée : $t.A = \{v_1(u_1), \ldots, v_m(u_m)\}$ où $A^v = \{v_1, \ldots, v_m\}$ est la liste de valeurs de l'attribut A dans t, et $A^u = \{u_1, \ldots, u_m\}$ est la liste de scores ou de poids attachés à chaque valeur dans A^v.

Les prédicats sont combinés dans une formule selon le syntaxe $f ::= p \mid f \wedge f \mid f \vee f \mid \neg f \mid (f)$, où f est une formule et p un prédicat flou. L'évaluation de f sur un tuple t est un score, $s(f, t) \in [0, 1]$, qui signifie A quel degré t satisfait f. $s(f, t)$ dépendant des prédicats flous dans f est intentionnellement non spécifié, elle est calculé par la « fonction de score s_f » en respectant les sémantiques des opérateurs de logique floue. Les arguments de s_f sont des scores, $s(p_i, t)$, de tuple t avec du respect des prédicats apparaissant dans f. Donc, $s(f(p_1, \ldots, p_n), t) = s_f(s(p_1, t), \ldots, s(p_n, t))$. Un *prédicat flou* a soit la forme $A \approx v$, où A est un attribut, $v \in$ dom (A) est une valeur constante et $\approx$ un *opérateur flou*; soit la forme $A_1 \approx A_2$, où les deux attributs A_1

[1] Une distribution de possibilités est une application π d'un domaine donné dans l'intervalle [0,1] où $\pi(a)$ donne le degré de possibilité associé au fait que la valeur effective de la variable soit a. La condition de normalisation impose qu'au moins une valeur (a_0) soit complètement possible, i.e., $\pi(a_0) = 1$

et A_2 sont dans le même domaine. L'évaluation de p sur t rend un score, $s(p, t) \in [0, 1]$, traduisant le degré de similarité entre $t.A$ et la valeur v. Les fonctions de score correspondent aux *t-normes* et *t-conormes* flous [15], les opérateurs *And* ($\wedge$) et *Or* ($\vee$) satisfont des propriétés intéressantes comme l'associativité et la commutativité [15]. On distingue deux interprétations pour ces opérateurs logiques : FS (fuzzy standard) et FA (fuzzy algébrique) comme illustré dans le tableau suivant :

	FS	FA
$s(f_1 \wedge f_2, t)$	$\text{Min}(s(f_1, t), s(f_2, t))$	$s(f_1, t) \,.\, s(f_2, t)$
$s(f_1 \vee f_2, t)$	$\text{Max}(s(f_1, t), s(f_2, t))$	$s(f_1, t) + s(f_2, t) - s(f_1, t) \,.\, s(f_2, t)$
$s(\neg f, t)$	$1 - s(f, t)$	$1 - s(f, t)$

Dans la plupart des applications l'interprétation choisie est FS. Dans ce papier, nous utilisons également FS pour calculer les opérateurs logiques.

Similarités des données symboliques

Comme chaque cellule d'une table de données symboliques peut être une liste de valeurs pondérées, une mesure de similarité entre deux cellules du même domaine ou entre deux tuples n'est plus une comparaison simple avec un résultat booléen mais une évaluation complexe avec un score en résultat. De plus, la combinaison de prédicats avec la théorie du flou ajoute un certain degré de complexité. Dans cette partie, les mesures de similarités au niveau des cellules et au niveau des tuples sont définies respectivement.

(A) Similarité au niveau cellule

La similarité au niveau cellule permet de définir des critères de type (attribut comparateur valeur | attribut) ou une combinaison logique de ceux-là. Ils sont à la base d'opérateurs algébriques comme la restriction et la jointure.

Etant donné un tuple t, la valeur symbolique de t sur l'attribut A une liste de valeurs avec des pondérations, $t.A = \{v_1(u_1), \ldots, v_m(u_m)\}$. Si un prédicat $q : A \approx w$, où w est une valeur symbolique constante. On distingue deux cas de figures. Le premier est lorsque les données ne sont pas pondérées. Dans ce cas, la mesure de prédicat q sur t peut être définie par la superposition des deux valeurs symboliques $t.A$ et w [1], $s(q, t) = |t \cap w| / t \cup w|$.

Ex1 : $t.A$ = {rose, rouge}, w ={rose, vert, bleu, marron, noir}, s (q,t) = 1/6

Dans le cas où l'on considère la pondération des valeurs pour un prédicat q de la forme q: $A_1 \approx w$ ou q: $A_1 \approx A_2$, la mesure de similarité est calculée par la distance de χ^2 [16]:

$s(q, t) = 1 - 1/2(\sum_i |u_i(A_1) - u_i(A_2)|^2/u_i(A_1) + \sum_i |u_i(A_1) - u_i(A_2)|^2/u_i(A_2))$.

Ex2 : $t.A_1$ = {(0.4) rouge, (0.5) bleu, (0.1) jaune} et $t.A_2$ = {(0.3) rouge, (0.4) bleu, (0.3) jaune}, $s(q, t)$ = 1 – 1/2 (0.01/0.3 + 0.01/0.4 + 0.04/0.3 + 0.01/0.4 + 0.01/0.5 + 0.04/0.1) = 0.69.

(B) Similarité au niveau tuple

Selon la définition de projection relationnelle, les tuples en double doivent être éliminés. Ici, nous définissons une ε-projection en n'en gardant qu'un seul exemplaire des tuples trop similaires (ceux dont la similarité dépasse un certain seuil ε). Il en est de même pour l'union, l'intersection et la différence.

Pour ce faire, on définit une mesure de similarité entre deux tuples de données symboliques t1 et t2 par la fonction de dissimilarité [17] $\boldsymbol{d}_p(\boldsymbol{t}_1,\boldsymbol{t}_2) = \sqrt[p]{\sum_{k=1}^{n}[\boldsymbol{c}_k \boldsymbol{m}(\boldsymbol{t}_1\boldsymbol{A}_k, \boldsymbol{t}_2\boldsymbol{A}_k)]^p}$ (1), où $k \in [1, n]$, c_k est le poids sur chaque attribut et la fonction $m(w1, w2)$ est calculée par la distance de χ^2.

4. Extension de l'algèbre des données relationnelles aux données symboliques

Cette section détaille les opérateurs algébriques définies sur les données symboliques de type distribution. Il s'agit de l'extension de la sélection, de la projection et de la jointure. C'est bien une

extension à ce nouveau type de données, dans la mesure où les types atomiques numérique ou catégorielles sont intégrés.

4.1. Sélection symbolique

La sélection symbolique est un opérateur sur une table symbolique produisant une nouvelle table symbolique de même schéma, mais comportant les seuls tuples qui vérifient la condition précisée en argument. Ici, comme dans la section III, nous considérons deux types de conditions: celles ne comparant que les valeurs (type 1), et celles tenant compte de la pondération (type 2). Pour le type 1, le prédicat est calculé par la superposition des valeurs présentée à la section III. Quant au type 2, le prédicat est une évaluation par la *distance de* X^2. Par ailleurs, les conjonctions et les disjonctions de prédicats atomiques sont évaluées par l'interprétation FS.

Nous montrons ci-dessous un exemple de sélection symbolique avec des prédicats flous. Pour exprimer facilement notre exemple, nous avons utilisé un langage de type SQL.

Exemple : faire une sélection de la table symbolique 1, avec la couleur rouge ou bleu, la composition du matériel à 100% de bronze ou près de 50% de cuivre et la température entre 30° et 80° et ce avec un degré de satisfaction (score) de plus de 50%.

*Select * from* table 1
Where (Couleur = {rouge} or Couleur = {bleu}) and (Matériel = {Cuivre(0.5)}
or Matériel = {Bronze(1)}) and (30° < température < 80°)
With s_f ***>= 50%***

Id	Couleur	Matériel	Température
1	Rouge, Bleu	Fer (0.2), Cuivre (0.8)	30°
2	Bleu	Cuivre (1)	25°
3	Vert, Bleu	Bronze (0.5), Fer (0.5)	39°
4	Rouge	Bronze (1)	80°

σ Couleur = {rouge} or Couleur = {bleu}) and (Matériel = {Cuivre(0.5)} or Matériel = {Bronze(1)}) and (30° < température < 80°) with ***sf >= 50%***

Id	Couleur	Matériel	Température
1	Rouge, Bleu	Fer(0.2), Cuivre(0.8)	30°
3	Vert, Bleu	Bronze(0.5), Fer(0.5)	39°

4.2. Projection symbolique

La projection symbolique d'une table symbolique consiste à composer une nouvelle table symbolique en enlevant à la table initiale tous les attributs non mentionnés en opérandes et en éliminant les tuples en double qui sont conservés une seul fois.

Dans le modèle relationnel, la projection ne garde qu'un exemplaire des tuples dont les valeurs sont exactement identiques. Cependant, une donnée symbolique peut être une liste de valeurs avec des distributions, deux données symboliques sont rarement identiques mais peuvent être très similaires.
Ex : t_1 = ({rouge, bleu}, {fer (0.25), Cuivre (0.75)}, 30°), t_2 = ({rouge, bleu}, {fer (0.24), Cuivre (0.76)}, 30°) sont presque identiques.

Afin d'éviter la redondance de données, nous considérons que les tuples dont la similarité dépasse un certain seuil sont presque identiques et un exemplaire suffit pour les représenter. Il est à noter que cet opération d'élimination des données similaire est seulement appliquée lorsque parmi les attributs projetés, certains sont de type multi-valué pondéré, autrement, on utilise toujours la projection relationnelle.

Le problème est que contrairement à l'égalité, la similarité n'est pas transitive. En effet, un tuple t_1 similaire à un deuxième tuple t_2, lequel est similaire à t_3 n'entraîne pas forcément que t_1 est similaire à t_3. De fait, il n'est pas facile de séparer les tuples réellement distincts. Ce problème se retrouve dans les méthodes de clustering en fouille de données.

Cela soulève néanmoins plusieurs questions. Comment déterminer les ensembles de tuples similaires ? Comment déterminer, entre plusieurs tuples de données symboliques similaires, lequel éliminer ? Faut-il conserver le plus significatif, mais alors, comment le déterminer ? Faut-il au lieu de garder un tuple calculer une moyenne ou un centroïde des tuples symboliques ?

Ces problèmes sont les mêmes que ceux rencontrés lors des opérations ensemblistes, par exemple : union, intersection, différence, etc.

Parmi les solutions qu'on peut considérer, la première pourrait être de parcourir la table symbolique et de considérer dans une même classe, l'ensemble des tuples similaires au tuple courant qui sera alors un représentant étant donné qu'il correspondra au centre de la classe. Une autre solution serait d'appliquer un algorithme de clustering comme il en existe sur les données symboliques. Le résultat dans ces deux cas est non déterministe et dépend de l'ordre du parcours des tuples. A ce stade de l'étude, nous n'avons pas tranché sur ces questions.

Exemple : faire une projection de la table symbolique 2 en choisissant l'attribut Matériel avec degré de similarité 90%.

Id	Couleur	Matériel	Température
1	Rouge, Bleu	Fer(0.2), Cuivre(0.8)	30°
2	Bleu	Cuivre(1)	25°
3	Rouge, Bleu	Fer (0.3), Cuivre(0.7)	39°
4	Rouge	Bronze(1)	80°

$\Pi_{\text{Matériel}}$

Matériel
Fer(0.2), Cuivre(0.8)
Cuivre(1)
Fer (0.3), Cuivre(0.7)
Bronze(1)

Après la fonction de calcul de similarité (1) introduite dans la section III, les deux tuples {Fer(0.2), Cuivre(0.8)} et {Fer(0.2), Cuivre(0.8)} sont presque identiques. La question est donc de savoir lequel de ces deux tuples éliminer ?

4.3. Jointure symbolique

Une jointure en relationnel compose deux à deux des tuples de deux tables sous la condition qu'il répondent à un critère sur leurs attributs respectifs. La définition de l'opération de jointure ici est une extension du critère à la similarité. Dans le cas présent, cette similarité est définie de la même manière que pour la sélection impliquant avec la seule différence qu'elle compare les attributs entre eux. La requête de jointure peut également retourner le score de similarité entre les tuples joints.

5. Conclusion et perspectives

Dans cet article, nous avons proposé une extension de l'algèbre relationnelle aux données symboliques de type multi-valué pondéré. Plus précisément le concept de requête floue est intégré et des extensions sont proposées pour les opérateurs le type sélection, projection et jointure. Cet article présente le début d'un travail de recherche motivé par le fait qu'il n'existe pas d'équivalent sur ce type de données. Les besoins des applications pour ce type de représentation sont réel comme dans notre projet [2]. Les

perspectives de cette recherche est d'étendre les techniques OLAP à ce type de données afin de permettre des agrégations et une analyse flexible.

Référence

[1] H.-H. Bock, E. Diday Analysis of Symbolic Data – Exploratory Methods for Extracting Statistical Information for Complex Data. In *series: Studies in Classification, Data Analysis, and Knowledge Organisation*, Vol. 15, Springer-Verlag, Berlin, (2000), ISBN 3-540-66619-2

[2] European project HEARTS - HEALTH EFFECTS AND RISKS OF TRANSPORT SYSTEMS http://www.euro.who.int/hearts

[3] P. Bosc, H, Prade, An introduction to fuzzy and possibility theory-based approaches to the treatment of uncertainty and imprecision in data base management system, In *Uncertainty Management in Information Systems – From Needs to Solutions*, (A. Motro and P. Smets Eds), Kluwer Academic Publishers, 1997, p. 285-324.

[4] L. Duval, O. Pivert, A propos de requêtes possibilistes adressées à des bases de données possibilistes. In Proc *18ème Journées Bases de Données Avancées 2002*.

[5] P. Bosc, L. Duval, O. Pivert, About Selections and Joins in Possibilistic Queries Addressed to Possibilistic Databases. In Proc DEXA 2002, LNCS 2453, pp. 597-606, 2002.

[6] L.A. Zadeh, fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, vol. 1, 1978, p. 3-28.

[7] H. Prade, C. Testemale, Generalizing database relational algebra for the treatment of incomplete/uncertain information and vague queries. Information Sciences, vol. 34, 1984, p. 115-143.

[8] D. Dubois,, J. Lang, H. Prade, Automated reasoning using possibilistic logic: semantics, belief revision and variable certainty weights. In Proc *Perprints of the 5th Workshop on Uncertainty in Artificial Intelligence*. Windsor, Ontario, Aug. 18-20, 1989, 81-87. Revised version in IEEE Trans. On Data and Knowledge Engeineering, 1993, to appear.

[9] D. Dubois,, J. Lang, H. Prade, Handling uncertainty, context,vague predicates, and partial inconsistency in possibilistic logic. Preprints of the Fuzzy Logic in Artificial Intelligence Workshop held in conjunction with IJCAI'91, Sydney, Australia, Aug. 25, 13-23.

[10] D. Dubois,, J. Lang, H. Prade, Possibilistic logic. In Proc *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol . 3(D.M. Gabbay, ed.), Oxford University Press, to appear.

[11] R. J. Smith and S.F. Chang. Tools and Techniques for Color Image Retrieval, In Proc *ISET/SPIE*, volume 2670, 1994. Storage & Retrieval for Image and Video Databases IV.

[12] M. Ortega,Y. Rui, K. Chakrabarti, K. Porkaew, S, Mehrotra and T. Huang; Supporting Ranked Boolean Similarity Queries in {MARS. In Knowledge and Data Engineering, Volume 10, 1998, 905-925.

[13] P. Ciaccia, D. Montesi, W. Penzo and A. Trombetta Fuzzy Query Languages for Multimedia Data. In *Design and Management of Multimedia Information Systems: Opptunities and Challenges*, M.R, Syed editor, Idea Group Publishing, Hershey, PA, USA, 2001

[14] G. Gardarin, Bases de données: objet & relationnel. In Eyrolles, troisième tirage 2001.

[15] G. J. Klir and B. Yuan, Fuzzy Sets and Fuzzy Logic. Prentice Hall PTR, 1995.

[16] D. Csiszár, Information-type measures of difference of probability distributions and indirect observation, *Studia Scient. Math. Hung*., vol. , pp. 299-318, 1967.

[17] D. Malerb, F.Esposito and M. Monopoli Comparing dissimilarity measures for probabilistic symbolic objects. In *Proc Data Mining III* 2002.

.

Mot-clés : données symboliques, algèbre symbolique, prédicats flous, requêtes complexes, lien logique

Extraction de classes homogènes et données symboliques

Aïcha El Golli[*,**], Yves Lechevallier[*]

[*]INRIA-Rocquencourt
Domaine De Voluceau,
BP 105 Bâtiment 18
78153 Le Chesnay Cedex, France
aicha.elgolli@inria.fr
yves.lechevallier@inria.fr
[**]Université de Paris IX Dauphine,
Lise Ceremade, Place du Maréchal De Lattre de Tassigny,
75775 Paris Cedex 16, France

Résumé. L'objectif de ce travail est de proposer une méthode divisive de classification qui construit un arbre de décision binaire. Cette méthode construit les classes par divisions successives de l'ensemble des individus en choisissant la meilleure coupure sur une variable et tout en minimisant le critère d'inertie intraclasse. On obtient une classification facilement interprétable, l'arbre hiérarchique étant ici un arbre de décision. Ceci a motivé son application dans le processus d'extraction d'objets symboliques (Diday et al. (1991)) à partir des bases de données relationnelles. L'association de cette méthode au processus d'extraction permet d'enrichir les descriptions obtenues. La performance de cette méthode ainsi que son utilité sont évaluées par quelques exemples.

1 Introduction

Ayant un ensemble d'individus G de Ω, le but de la généralisation en analyse de données est de construire une bonne représentation de G. Une approche proposée dans le cadre de l'analyse de données symbolique (données ayant une structure complexe telles que les intervalles, les distributions, ...) (Diday et al. (1991)), permet d'extraire à partir d'une base de données un ensemble de descriptions par généralisation. Cette généralisation constitue une première et importante étape de l'analyse de données symbolique car elle permet d'associer aux classes d'individus une description de nature symbolique. Cette généralisation permet de résumer un ensemble de tuples, représentant un groupe, par une "bonne" description symbolique. Souvent cette approche présente une sur-généralisation dans le sens que certaines descriptions incluent des individus potentiels, individus peu probables d'appartenir aux groupes de départ. Une idée de décomposition, permettant d'éliminer ces individus potentiels tout en enrichissant les descriptions, est apparue. Cette décomposition nécessite une méthode de partitionnement telle qu'une méthode divisive de classification sur chacun des groupes. La méthode de décomposition proposée est issue de la méthode de classification décrite dans la thèse de Marie Chavent (Chavent (1997)), (Chavent (1998)). La construction des différents

segments se réalise par divisions successives de l'ensemble d'individus, un nœud est toujours divisé en deux nœuds fils en minimisant le critère d'inertie intraclasse. Ces deux nœuds fils sont obtenus en choisissant la meilleure coupure sur une variable. La classification obtenue est facilement interprétable grâce à l'arbre de décision binaire obtenu, et les descriptions symboliques finales sont facilement obtenues grâce aux différentes coupures de l'arbre. Dans ce papier nous allons d'abord commencer par introduire la méthode de généralisation utilisée dans le cadre de l'analyse de données symbolique et la méthode divisive de classification proposée. Nous proposons ensuite l'association de cette dernière au processus de généralisation pour finir par exposer les performances et l'utilité de cette décomposition dans l'enrichissement des descriptions.

2 Construction d'objets symboliques par généralisation

La méthode de généralisation proposée dans le cadre de l'analyse de données symbolique, permet la construction d'objets symboliques booléens ou modaux à partir d'une requête SQL et par groupement. Cette généralisation part donc d'une base de données relationnelle et résume les tuples décrivant un groupe par une description symbolique qu'on appelle *assertions* (Diday et al. (1991)) et chacune d'elles n'est autre qu'une conjonction d'événements élémentaires. Un événement élémentaire est sous la forme: $[Y_i\ R\ d]$ avec Y_i une variable symbolique, R une relation ($=$, $\in$,...) et d une description (une partie du domaine de la variable Y_i).

Définition: Opérateurs de généralisation (Stephan et al. (2000))

Ayant une population Ω décomposée en k ensembles de groupes (G_1,G_2,...,G_k). L'ensemble $E = 1,...,k$ est ensemble des labels associés aux groupes. Chaque variable symbolique $Y_i : E \longrightarrow B_i$, doit être définie à partir de la variable monovaluée $\widetilde{Y}_i$ définie sur la population. Les deux principaux opérateurs de généralisation utilisés sont:

1. Pour une variable quantitative $\widetilde{Y}_i$, on définit une variable intervalle Y_i: $Y_i(k) = [a_k,b_k]$ tel que $a_k = min\widetilde{Y}_i(w)$ et $b_k = max\widetilde{Y}_i(w)$ avec $w \in \Omega$ un individu du groupe G_k. Dans ce cas, B_i est l'ensemble de tous les intervalles inclus dans le domaine de $\widetilde{Y}_i$.
2. Pour une variable qualitative $\widetilde{Y}_i$, la variable symbolique correspondante peut être :
   - Une variable multivaluée Y_i, c'est à dire une correspondance (set-valued function) où $Y_i(k)$ représente l'ensemble des valeurs observées sur les individus du groupe G_k et B_i est l'ensemble des parties du domaine de $\widetilde{Y}_i$.
   - Une variable modale Y_i où $Y_i(k) = (\eta_1^k, \ldots, \eta_l^k)$ est le l-tuple qui représente les poids associés aux catégories $1, \ldots, l$ de la variable $\widetilde{Y}_i$ par une mesure discrète Y_i associée à G_k

La généralisation des groupes G_i, à partir d'une table relationnelle par des assertions A_i, induit une sur-généralisation. Cette sur-généralisation est caractèrisée par l'existence d'un grand nombre d'observations potentielles, observations qui ne correspondent à aucune description des individus de G_i .

L'objectif de ce travail est donc de spécifier l'assertion de départ de manière à réduire efficacement son volume tout en recouvrant le plus possible les individus du groupe.

En effet, nous mesurons la qualité d'une assertion par un critère de densité, qui est le nombre d'individus recouverts par l'assertion par unité de volume de la description.

Définition: Le critère de densité d'une assertion (Stephan (1998))

Soit un groupe G_i, une partie de la population Ω, et son assertion correspondante A_i de description d_i. On définit la densité de A_i par rapport à G_i comme:

$$Dens(A_i) = \frac{Card(G_i)}{Vol(d_i)}$$

avec $Card(G_i)$, le nombre d'individus de G_i.

$d_i = (d_{i^1}, d_{i^2}, ..., d_{i^p})$ étant la description de G_i, le volume de d_i est défini comme:

Définition:

$$vol(d_i) = \prod_{j=1}^{p} \mu(d_{ij}) \quad où \quad \mu(d_{ij}) = \begin{cases} card(d_{ij}) & \text{si} \quad Y_j \quad qualitative \\ \overline{d_{ij}} - \underline{d_{ij}} & \text{si} \quad Y_j \quad quantitative \end{cases}$$

Dans le cas d'existence de règles, le volume est remplacé par le potentiel de description (Decarvalho (1996)).

Donc plus la densité au sein de l'assertion est élevée, plus la qualité de la généralisation augmente. La figure ci-dessous, Figure1 constituée de 75 points dans le plan et représentant un groupe d'individus, est un cas de sur-généralisation. La généralisation a induit la présence de zones ne possédant aucune observation, ce qui diminue la qualité de cette dernière. Une décomposition donc sera très utile afin de réduire efficacement le volume de l'assertion.

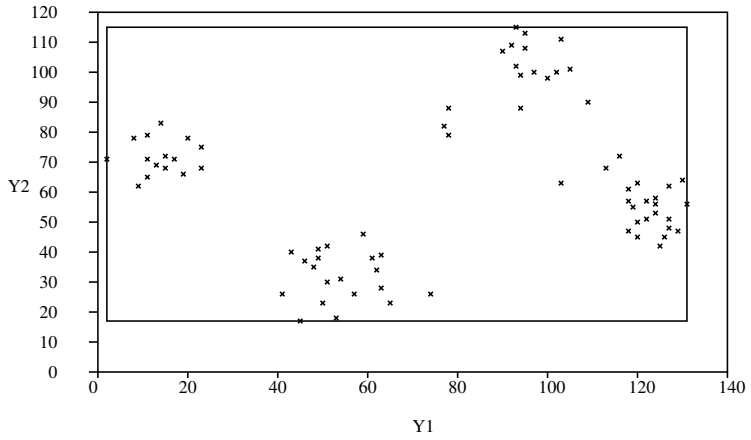


FIG. 1 – *Une sur-généralisation nécessitant une décomposition*

L'amélioration de la qualité de la généralisation est cruciale dans la suite des traitements en analyse de données symboliques. Car les opérateurs de généralisation utilisés traitent les descriptions variable par variable sans prise en compte des associations. Ces

opérateurs sont souvent sensibles aux observations aberrantes et à l'hétérogénéité de certaines descriptions. Lorsqu'il existe des associations entre les variables, on observe un grand nombre d'éléments qui ne correspondent à aucune description individuelle. Pour cela, nous proposons une méthode de décomposition de l'assertion par partitionnement des individus. Ce partitionnement vise à réduire les associations entre les variables et possède l'avantage d'associer à chaque classe d'individus, sa généralisation au moyen d'une assertion. Dans ce qui suit nous allons vous présenter la méthode de partitionnement choisie et basée sur une méthode divisive de classification.

3 Une méthode divisive de classification

Les méthodes divisives de classification sont des méthodes de classification hiérarchiques. Elles partent d'un ensemble d'individus Ω et procèdent par divisions successives des classes jusqu'à l'obtention de la partition la plus fine. En revanche, les méthodes ascendantes de classification partent des singletons et procèdent par agrégations successives. Les complexités de ces deux familles de méthodes de classification hiérarchiques sont différentes. En effet, lors de la première étape d'une méthode ascendante, il faut évaluer toutes les agrégations possibles de deux individus parmi les n individus, soit $n(n-1)/2$ possibilités, tandis que dans un algorithme descendant on doit réaliser toutes les divisions de n individus en 2 sous ensembles non vides, soit $(2^{n-1}-1)$ possibilités. Plusieurs stratégies ont été proposées dans le cadre des méthodes divisives, afin de réduire cette complexité (Macnaughton (1964)), (Gowda et al. (1978)). Nous proposons d'utiliser la méthode divisive développée par Marie Chavent dans le cadre de la décomposition (Chavent (1997)), (Chavent et al. (1999)), (Chavent (2000)) sur des données quantitatives ou qualitatives où la recherche de la solution optimale ne se fait qu'à partir des partitions admissibles qui doivent respecter les structures des variables.

3.1 Présentation générale de la méthode proposée

Dans le cadre des données symboliques, l'ensemble des objets à classer sont des objets symboliques qui sont décrits par des variables intervalles, mulivaluées ou modales. Le choix du critère d'évaluation des bipartitions construites à chaque étape est un critère d'homogénéité H égal à la double somme des carrés des distances entre individus appartenant à la même classe. Ce critère nécessite le calcul du tableau de dissimilarités. Une bipartition admissible est une partition induite par une question binaire qui dépend du type de la variable. En effet, elle est de la forme "$Y_i \leq c$" si Y_i est une variable quantitative, avec c une valeur de coupure et de la forme "$Y_i \in \{m_k,...,m_j\}$" si Y_i est une variable qualitative, avec $\{m_k,...,m_j\}$ est une partie du domaine de Y_i. L'algorithme divisif proposé donc dans (Chavent (1997)) est le suivant:

Initialisation:

$P_1 = \Omega$; $k \leftarrow 1$;

Tant Que $k < K-1$ **alors**:

1. pour chaque classe $C \in P_k$, choisir parmi les bipartitions (C_1,C_2) de C induites

par les *questions binaires*, la partition qui minimise:

$$H(C_1) + H(C_2) = \frac{p_j p_{j'}}{2\mu(C_1)} \sum_{w_j \in C_1} \sum_{w_{j'} \in C_1} d^2_{jj'} + \frac{p_j p_{j'}}{2\mu(C_2)} \sum_{w_j \in C_2} \sum_{w_{j'} \in C_2} d^2_{jj'}$$

2. puis choisir la classe $C \in P_k$ qui maximise:

$$W(P_k) - W(P_{k+1}) = H(C) - H(C_1) - H(C_2)$$

3. $P_{k+1} = P_k \cup \{C_1, C_2\} - \{C\}$
4. $k \leftarrow k + 1$;

Fin Tant Que

K étant le nombre de classes choisi et donc correspondant au critère d'arrêt, $\mu(C_1)$ et $\mu(C_2)$ étant les poids respectifs de C_1 et C_2.

Remarque: L'arbre de décision obtenu est une hiérarchie sur les $(K+1)$ classes qui est indicée par $W(P_k) - W(P_{k+1})$. Ainsi une classe divisée avant une autre est représentée par un palier plus haut dans l'arbre hiérarchique.

Quand le nombre d'individus est grand (dépassant 1000), le calcul du tableau de dissimilarités entre les individus deux à deux augmente rapidement la complexité. Comme dans notre cas les données sont quantitatives ou qualitatives, la notion de centre de gravité peut être définie. Ainsi la complexité de notre algorithme est liée au nombre des bipartitions:

- Si la variable Y_i est quantitative, on évalue au maximum $(n-1)$ bipartitions, n étant le nombre d'individus de la classe en question. En effet, si il y a n observations pour la variable Y_i, il y a $(n-1)$ points de coupure sur cette variable.
- Si la variable Y_i est qualitative ordinale, on évalue au maximum $(m-1)$ bipartitions, m étant le nombre de modalités de Y_i.
- Dans le cas d'une variable qualitative nominale on se heurte à un problème de complexité et le nombre de dichotomies du domaine d'observation est alors égal à $(2^{m-1} - 1)$.

Le critère d'inertie a été choisi comme critère d'homogénéité. L'inertie totale d'une classe ainsi que l'inertie intraclasse d'une partition se calculent par rapport à leurs centres de gravité respectifs:

$$I_g(C) = T = \sum_{x_i \in C} p_i d^2_M(x_i, g_C) = H(C) \tag{1}$$

avec g_C est le centre de gravité de la classe C:

$$g_C = \frac{1}{\mu(C)} \sum_{i \in C} p_i x_i \qquad avec\ \mu(C) = \sum_{i \in C} p_i$$

3.1.1 Le critère d'inertie comme mesure de qualité d'une classe

Soit n le nombre d'individus dans Ω. Chaque individu est décrit sur p variables, $Y_1, Y_2, ..., Y_p$, par un vecteur x_i, doté d'un poids p_i, avec $i = 1,...,n$. Généralement les

poids p_i des individus sont égaux à 1 ou à $1/n$. L'inertie I de la classe C_k est basée sur la distance d_M entre les individus de la classe et son centre de gravité. La distance d_M choisie est:

* la distance Euclidienne si les variables sont continues et qualitatives ordinales;
* la distance de χ^2 sur le tableau disjonctif complet dans le cas des variables qualitatives nominales.

(M est une matrice symétrique définie positive).

- cas quantitatif ou qualitatif ordinal

$$\forall x_i \in R^p, d_M^2(x_i,g_k) =^t (x_i - g_k)M(x_i - g_k) \tag{2}$$

Avec g_k est le centre de gravité de la classe C_k. Dans le cas où $M = Id$, $d^2(x_i,g_k) = \sum_{j=1}^{p}(x_i^j - g_k^j)^2$

- cas qualitatif nominal: Dans ce cas les individus sont décrits par un ensemble de variables qualitatives nominales. Une étape préliminaire consiste d'abord à recoder les valeurs observées grâce à un tableau à codage disjonctif complet formé de valeurs dans l'espace $B^m = \{0,1\}^m$

$$\forall x_i \in B^m, \chi^2(x_i,g_k) = \frac{n * p}{n_j} \sum_{j=1}^{m} (\frac{x_i^j}{p} - g_k^j)^2 \tag{3}$$

avec n est le nombre total d'individus, p le nombre total de variables, m le nombre total de modalités et x_i^j le représentant de l'individu i pour la modalité j dans l'espace $B = \{0,1\}$,

$n_j = \sum_{i=1}^{n} x_i^j$ $\qquad$ $g_k^j = \frac{n_j(k)}{p * n(k)}$ $\qquad$ $n_j(k) = \sum_{i \in C_k} x_i^j$

$n(k)$ le nombre d'individus dans la classe C_k

Étant donné un espace métrique muni d'une métrique Euclidienne, alors minimiser le critère d'homogénéité associé à la partition en deux classes (C_1,C_2) de C, revient d'après le théorème de Huygens à maximiser l'inertie interclasse:

$$B = \mu(C_1)d^2(g_C,g_{C_1}) + \mu(C_2)d^2(g_C,g_{C_2})$$

Or d'après le critère de Ward (Ward (1963)) on a:

$$B = \frac{\mu(C_1) * \mu(C_2)}{\mu(C_1) + \mu(C_2)} \; d^2(g_{C_1},g_{C_2})$$

Le critère d'évaluation de la bipartition dans notre cas donc se résume à la simple distance entre les centres de gravité des deux sous-classes C_1 et C_2 formant la bipartition de C. De plus, lors de l'évaluation de l'ensemble des partitions admissibles, le

calcul des centre de gravités g_{C_1} et g_{C_2} de deux coupures consécutives, c_i et c_{i+1}, se déduit facilement grâce aux formules:

$$g_{C_1}(c_{i+1}) = \frac{\mu(C_1)g_{C_1}(c_i) - \mu(A)g_A}{\mu(C_1) - \mu(A)}$$

$$g_{C_2}(c_{i+1}) = \frac{\mu(C_2)g_{C_2}(c_i) + \mu(A)g_A}{\mu(C_2) + \mu(A)}$$

A étant l'ensemble des individus qui passent de C_1 vers C_2 lors du changement de question binaire.

3.1.2 Le traitement des valeurs manquantes

Dans la pratique il y a souvent des données manquantes. Plusieurs techniques ont été dévelopées pour résoudre ce problème. Certaines de ces techniques ont choisi d'éliminer les individus ayant des valeurs manquantes d'autres, ont plutôt choisi d'estimer la distance entre deux vecteurs de données à valeurs manquantes. Comme nous avons choisi le critère d'inertie cela nécessite au départ le calcul des centres de gravité de toutes les bipartitions induites pour chaque étape. Le calcul d'un centre de gravité se réalise sur l'ensemble des valeurs observées.

L'individu ayant une valeur manquante sur la variable traitée, sera ensuite affecté à la classe de la bipartition induite dont le centre de gravité est le plus proche. La technique choisie ici pour le calcul de la distance entre le vecteur x_i et le vecteur centre de gravité g_k, contenants des valeurs manquantes, est la suivante (Jain et al. (1988)): On définie d'abord la distance d_j entre 2 vecteurs sur une variable j:

$$d_j = \begin{cases} 0 & \text{si } x_i^j \textit{ ou } g_k^j \textit{ est manquant} \\ x_i^j - g_k^j & \textit{sinon} \end{cases}$$

ensuite la distance entre x_i et g_k est:

$$d(i,g_k) = \frac{p}{p - d_0}\sum_{j=1}^{p} d_j^2$$

avec d_0 est le nombre des valeurs manquantes dans x_i ou g_k ou les deux. Quand il n'y a pas de valeurs manquantes, notez que $d(i,g_k)$ n'est autre que la distance Euclidienne.

3.2 Conclusion

Cette méthode divisive a l'avantage de réduire considérablement la complexité connue des algorithmes divisives. C'est une méthode simple et qui permet d'avoir un arbre de décision et une simple interprétation des classes homogènes obtenues. Dans notre cas l'utilisation du critère de Ward (Ward (1963)), a considérablement réduit les calculs.

4 Exemple et utilité de la décomposition dans l'opérateur de généralisation symbolique

On reprend l'exemple de la figure1 (Ruspini (1974)). Groupe G_1 de 75 individus repartis sur deux variables quantitatives Y_1 et Y_2. Sur ce groupe l'opérateur de généralisation produit l'assertion suivante: $A_1 = [Y_1 \in [2,131]] \wedge [Y_2 \in [17,115]]$.

L'application de la méthode de décompostion, basée sur la méthode divisive proposée précédemment, sur ce groupe permet de diviser notre groupe en 4 sous-groupes. Dans la première itération, la classe à diviser est la classe composée de l'ensemble des individus du groupe et parmis les p(n-1) bipartitions possibles (p: nombre de variables, n: nombre d'individus), soit 148 bipartitions (C1, C2), on choisit celle qui a la plus petite inertie intraclasses. Ceci va être traduit par la question binaire "$Y_1 \leq 75.5$". Dans un deuxième temps, on doit choisir de diviser l'une des classes C1 ou C2. On a choisi C1 et sa bipartition (C11, C12) correspondante à la question binaire "$Y_2 \leq 54$", car elle induit la plus grande diminution d'inertie. Ensuite la classe C2 est choisie parmi l'ensemble des classes (C11, C12, C2) pour être divisée en (C21, C22). La question binaire correspondante est "$Y_2 \leq 75.5$". Selon notre critère d'arrêt choisi, on s'arrête à ce niveau. L'arbre de décision final induit par notre méthode divisive est donc le suivant:

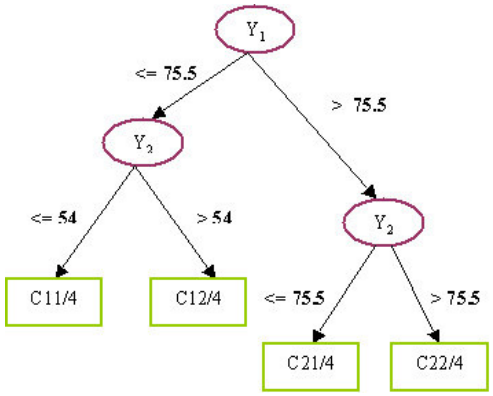


FIG. 2 – *Arbre de décision produit par la méthode divisive*

Cet arbre de décision est une hiérarchie sur 4 classes. Une classe divisée avant une autre est représentée plus haut dans l'arbre de décision afin d'avoir l'ordre de découpage.

Une fois cette décomposition réalisée, l'application de notre opérateur de généralisation sur le groupe G_1 renvoie la description suivante:

$$[Y1 \in [41,74] \wedge Y2 \in [17,46]] \vee [Y1 \in [2,23] \wedge Y2 \in [62,83]]$$

$$\vee [Y1 \in [103,131] \wedge Y2 \in [42,72]] \vee [Y1 \in [77,109] \wedge Y2 \in [79,115]]$$

Cette nouvelle description de G_1 est une disjonction d'assertions qui sera bien prise en

compte dans la suite des traitements. Le résultat de cette généralisation associée à la décomposition peut être un objet symbolique de type ***Hordes*** (Diday et al. (1991)).

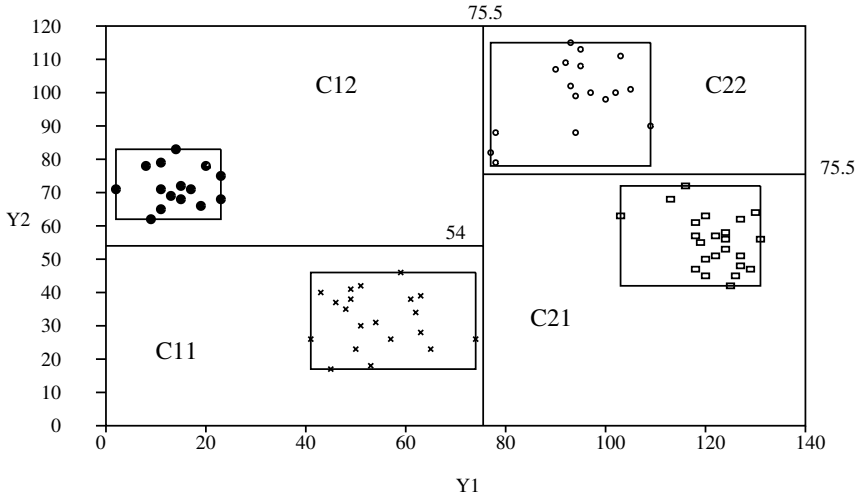


FIG. 3 – *La décomposition en 4 sous-classes*

5 Conclusion

L'intégration d'une décomposition dans le processus de généralisation améliore dans la plupart des cas la qualité des assertions obtenues car elle permet d'homogénéiser les groupes. Cette amélioration n'entraîne pas une perte d'information, comme dans la plupart des cas. Au contraire elle constitue des connaissances supplémentaires sur les groupes extraits des bases de données relationnelles. Ces connaissances sont prises en considération et constitues un grand apport pour les traitements en analyse de données symbolique. Le résultat de la décomposition constitue une méta donnée pour la base de données symbolique.

Références

CHAVENT M. (1997), Analyse de données symboliques une méthode divisive de classification , Thèse de doctorat, Université Paris Dauphine, 1997.

CHAVENT M. (1998), a monothetic clustering method , pattern recognition letters, vol. 19, 1998, pp 989-996.

CHAVENT M., STEPHAN V. (1999), From Generalization to clustering in the relational Database context, Proceedings of the conference on Knwoledge Extraction and Symbolic Data Analysis : KESDA'98, Eurostat's collection, 1999, pp 105-117.

CHAVENT M. (2000), CriterionBased Divisive Clustering for Symbolic objects , Ana lysis of symbolic data : exploratory methods for extracting statistical information from com plex data, Springer, 2000, pp 299-311.

GOWDA C.K., KRISHNA G. (1978), Disaggregative clustering using the concept of mutual nearest neighborhood., IEEE Transactions on Systems, Man, and Cybernetics 8., 1978.

DECARVALHO F. (1996), Histogrammes et indices de proximités en analyse de don nées symboliques , Actes de l'école d'été sur l'analyse de données symboliques. LISE CEREMADE, Université de Paris IXDauphine, Paris, , 1996, pp 101-127.

DIDAY E., KODRATOFF Y. (1991), des objets de l'analyse de données à ceux de l'analyse des connaissances , Induction symbolique et numerique à partir des données, Éditions Cépaduès, 1991, pp 9-76.

JAIN A. K., DUBES R. C. (1964), Algorithms for clustering data, Prentice Hall, 1988.

MACNAUGHTONSMITH, Dissimilarity analysis (1964), A new technique of hierarchical subdivision. , Nature 202, 1964.

RUSPINI E. (1970), Numerical methods for fuzzy clustering, Information Science, vol. 2, 1970, pp 319-350.

STEPHAN V. (1998), Construction d'objets symboliques par synthèse des résultats de re quêtes SQL , Thèse de doctorat, Université Paris Dauphine, 1998.

STEPHAN V., HÉBRAIL G., LECHEVALLIER Y. (2000), Generation of symbolic objects from relational databases , Analysis of symbolic data : exploratory methods for extracting statistical information from complex data, Springer, 2000, pp 78-105.

WARD J. (1963), Hierchical grouping to optimize an objective function, JASA, Vol.58, 1963.

Summary

The aim of this work is to propose a divisive clustering method that builds a binary decision tree. The division of the clusters is performed according to the within-cluster inertia criterion. The obtained clustering is easily interpretative, the hierarchical tree here is a decision tree. This motivated its application to the extraction process of symbolic objects (Diday (1991)) from data bases. Its association to the extraction process enriches the descriptions obtained. The performance of this method and its usefulness are evaluated with some examples.

Extraction de connaissances provenant de données multisources pour la caractérisation d'arythmies cardiaques

Elisa Fromont*,Marie-Odile Cordier*
René Quiniou *

*IRISA campus de Beaulieu, 35042 Rennes
efromont@irisa.fr
cordier@irisa.fr
quiniou@irisa.fr

Résumé. Cet article expose la problématique liée à l'extraction de connaissances provenant de données multisources et propose une application en cardiologie. Le but est ici d'apprendre des règles pertinentes caractérisant des arythmies cardiaques à partir de données reflétant des comportements cardiaques, telles que les différentes voies d'un électrocardiogramme ou la mesure de la pression artérielle. Les règles sont obtenues grâce à un apprentissage par programmation logique inductive soit globalement à partir de toutes les données disponibles, soit indépendamment à partir des données sur chacune des voies puis par fusion des règles obtenues.

1 Problématique

L'extraction de connaissances à partir de données provenant de sources hétérogènes pose deux principaux problèmes : l'un lié à la nature des sources et en particulier à la confiance et aux connaissances que l'on possède sur celles-ci et l'autre au choix des techniques à utiliser pour extraire des connaissances.

Pour le premier problème, deux cas de figure sont envisagés :

1. Si l'on étudie des données de même type provenant de sources différentes, comme par exemple les affirmations de plusieurs agriculteurs consultés pour prédire le temps, le problème est de quantifier le degré de pertinence ou même la fiabilité de chacune des sources (Cholvy 03). En effet, si l'une des sources s'avère beaucoup plus fiable ou porteuse d'informations que les autres, on peut se demander l'intérêt d'introduire de nouvelles données qui pourront entacher d'incertitudes les données de la source plus fiable ou plus pertinente. Ce problème est généralement résolu par des algorithmes de type "vote" (Dubois et al 2001) dans lesquels chaque source se voit associer une pondération liée à l'estimation de sa pertinence ou de sa fiabilité. Sur des données contradictoires ou redondantes, les données de la source qui se voit attribuer le meilleur score seront choisies.

2. Si les données offrent des points de vue différents sur une même situation, le problème est de savoir si l'un des points de vue apporte plus d'informations que les autres ou s'il est préférable de profiter de la complémentarité de ceux-ci. Par exemple, si l'on veut prédire le temps qu'il fera demain et que l'on dispose d'un baromètre, d'un thermomètre et d'un capteur d'humidité, on peut penser que toutes ces informations peuvent servir et sont fiables mais que l'information provenant du baromètre apporte à elle seule une réponse à la question. On peut aussi penser qu'utiliser la température seule offre trop peu d'indications pour répondre à la question, par contre, associer la température au baromètre peut permettre de préciser et d'accentuer la pertinence du diagnostic. Dans le cas d'un capteur peu fiable, on peut également extraire des informations moins précises (des variations de température au lieu d'une mesure à un temps donné) mais plus fiables, puis utiliser ces informations pour renforcer celles apportées par la ou les sources estimées fiables.

Dans les deux cas, il est toujours possible de réduire l'incertitude sur les données des sources estimées moins fiables ou moins pertinentes en utilisant des connaissances sur le domaine (dans le cas de l'exemple, on sait qu'un temps pluvieux s'accompagne d'une montée de pression et d'une hausse de température).

Le deuxième problème est de choisir la méthode d'extraction des connaissances à partir des différentes sources. Trois choix sont possibles.

1. Fusionner les informations a priori, c'est-à-dire directement au niveau des capteurs (Thoraval et al 1997, Hernandez et al 1999) et extraire des connaissances à partir des données transformées. Par exemple si les données proviennent des capteurs d'humidité, de température et de pression, on cherchera à se ramener à un capteur "virtuel" dont les données seront fonction des trois mesures fournies. La fusion à ce niveau peut entraîner une perte d'information et une structuration indésirable des données. Cette méthode est équivalente à l'extraction de connaissances provenant d'une seule source de donnée (le capteur virtuel), nous ne l'étudirons donc pas ici.
2. Fusionner les informations apportées par chacune des sources au moment de l'extraction des connaissances. Les exemples constituant la base de connaissance sont alors décrits, dans le cadre de notre exemple météorologique, à la fois en terme de température mais aussi en terme de pression et d'humidité. Les données ne sont pas fusionnées pour former de nouvelles données, elles sont agrégées. Ceci permet d'éviter l'étape de fusion proprement dite, mais le résultat fourni par l'outil d'apprentissage sera bien fonction des différentes informations relatives à chaque source. Ce choix augmente considérablement la combinatoire du problème et la complexité des calculs effectués. Cette combinatoire peut être néanmoins nettement diminuée lorsque les sources sont corrélées (une partie de l'information est redondante) pour peu que l'on dispose de connaissances suffisantes pour réduire les données initiales. Notons également que ce choix ne permet pas de réutiliser des informations extraites d'une source en cas de dysfonctionnement d'une ou plusieurs autres.
3. Extraire des connaissances indépendamment à partir des différentes sources puis

combiner les résultats obtenus pour répondre au problème posé. Cela peut se faire par combinaison de classifieurs (Gancarski et al 2002), par vote (Dubois et al 2001), en utilisant des règles de "bonne fusion" issues des connaissance sur le domaine (Bloch et al 2001) etc. Cette méthode pose le problème du choix des classifieurs ou du biais de langage. Ceux-ci peuvent en effet se montrer aussi dissemblables et ardus à combiner que peuvent l'être les données initiales. Cette méthode semble cependant plus adaptée si les données fournies par les sources sont indépendantes. Il permet en outre de disposer de l'information extraite des autres sources en cas de dysfonctionnement de l'une d'elles.

Nous présentons tout d'abord le problème d'extraction de connaissances à partir de données multisources provenant de capteurs utilisés en médecine pour le diagnostic d'arythmies cardiaques. Puis, nous donnons quelques résultats obtenus dans ce contexte. Nous discutons avant de conclure, quelques questions soulevées par ce travail.

2 Application à la rythmologie

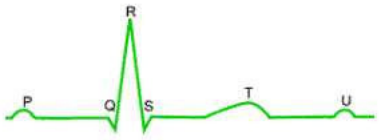


FIG. 1 – *Exemple de battement cardiaque normal*

2.1 Brève introduction à la cardiologie

Les troubles du rythme cardiaque (ou arythmies) sont dus à des problèmes de conduction électrique à l'intérieur du cœur. L'électrocardiogramme (ECG) est le reflet de l'activité électrique cardiaque. Il est mesuré en plaçant des électrodes sur le thorax près du cœur (6 voies de V1 à V6) ou sur les bras et les jambes (dérivations bipolaires D1 à D3 et unipolaires AV). Chaque électrode mesure un signal différent. Un électrocardiogramme "normal" est une succession de battements normaux comme celui indiqué sur la Figure 1. Deux ondes particulières caractérisent cet ECG : l'onde P et le complexe QRS (toutes deux conséquences d'une propagation électrique dans des parties spécifiques du cœur). Les ondes T et U, beaucoup moins utilisées, ne sont pas prises en compte dans cette étude. Lorsque l'ECG est anormal (distance entre les ondes ou forme des ondes anormale, absence d'onde etc.) on peut diagnostiquer une arythmie. L'activité cardiaque se reflète également sur la courbe de pression artérielle. En effet, lorsque les ventricules se contractent pour expulser le sang, la pression augmente fortement : c'est la *systole ventriculaire* (elle se produit peu après l'observation d'un complexe QRS sur l'ECG). La phase de repos (avant l'onde P) est appelée *diastole*.

FIG. 2 – *Exemple de signaux étudiés*

2.2 Le problème de rythmologie

Les médecins sont habitués à diagnostiquer des troubles du rythme cardiaque à partir des différentes voies de l'ECG. Pourtant, ils ont parfois à leur disposition des données telles que la mesure de la pression artérielle, les phonocardiogrammes, des mesures de ventilation, d'accélération etc. Ces informations supplémentaires pourraient être utilisées en Unité de Soin Intensif pour Coronariens (USIC) pour réduire le nombre de fausses alarmes provenant des appareils programmés pour diagnostiquer des troubles du rythme. D'un point de vue pratique, seules certaines arythmies très graves (considérées comme des "alarmes rouges") sont diagnostiquées automatiquement, et ce de manière trop sensible dans un souci de sécurité. Le but du travail amorcé dans le projet Calicot (Carrault et al 2003) est d'améliorer le diagnostic des troubles du rythme cardiaque. En particulier, nous voudrions étendre le diagnostic à d'autres arythmies, non létales si elles sont détectées suffisamment précocement (les alarmes dites "oranges"), en utilisant les données provenant de plusieurs sources homogènes (plusieurs voies d'un même ECG par exemple) ou hétérogènes (pression artérielle et voies d'ECG par exemple).

2.3 Données

La base de données utilisée est la base MIMIC (Multi-parameter Intelligent Monitoring for Intensive Care) (Moody et al 1997). Elle contient des données enregistrées sur 72 patients en USIC au Beth Israël Hospital Arrhythmia Laboratory. Les enregistrements ont une durée variable pouvant excéder 40 heures mais sont divisés en fichiers de 10 minutes chacun. Certaines des données utilisées sont très corrélées voire redondantes, puisqu'elles concernent différentes voies d'un même électrocardiogramme (souvent V1 et V5 cf Figure 2). D'autres sont indépendantes, comme les informations sur la pression artérielle systolique ou voie hémodynamique (ABP : "Arterial Blood Pressure").

Les données brutes extraites de la base MIMIC sont transformées en descriptions

symboliques de signaux en utilisant des outils de traitement de signal. Ces descriptions symboliques sont stockées dans des bases de connaissances logiques sous forme de faits Prolog (cf Figures 3 et 4). La Figure 3 donne un exemple de 4 faits Prolog apparaissant dans un exemple de Doublet ventriculaire. Le premier fait est un complexe *QRS* nommé *r7_doublet_3_II* apparaissant au temps *5026* de forme *normal*. L'onde précédente est nommée *p7_doublet_3_II*, la distance entre l'onde de même type (ici *QRS*) précédente et l'onde courante est de *734*, la distance entre l'onde de même type à la position (n-2) et le *QRS* courant est de *1454*, la distance entre l'onde *P* précédente et le *QRS* (intervalle *PR*) est de *168* et la distance entre l'onde *P* à la position (n-2) et le *QRS* courant (intervalle *PR2*) est de *902*. La Figure 4 apportent des données similaire pour le pression. Nous disposons pour chaque exemple de l'information sur l'amplitude des ondes étiquetées (ondes *P* et *QRS* pour l'ECG, la diastole et la systole pour la voie hémodynamique), la distance entre deux ondes, les ondes précédentes et la forme des ondes. Ce processus de traitement des données a été expérimenté lors du projet Calicot (Quiniou et al 2001) et a porté ses fruits en ce qui concerne l'extraction de connaissances à partir de données provenant d'une seule voie d'ECG. Les données concernent 7 arythmies cardiaques particulières: la tachycardie ventriculaire (considérée comme une alarme rouge en USIC), le doublet ventriculaire, le bigéminisme, l'extra-systole ventriculaire, la tachycardie supra-ventriculaire, la fibrillation auriculaire (considérées comme des alarmes orange en USIC) et le rythme sinusal ou rythme normal.

```
  .......
qrs(r7_doublet_3_II,5026,normal,p7_doublet_3_II,734,1454,168,902).
qrs(r8_doublet_3_II,5638, abnormal,r7_doublet_3_II,674,1408,842,1576).
qrs(r9_doublet_3_II,6448, abnormal,r8_doublet_3_II,796,1470,1638,2372).
p(p8_doublet_3_II,7146, normal,r9_doublet_3_II, 2256,2990,618,1414).
 ........
```

FIG. 3 – *Exemple de données ECG pour un doublet ventriculaire*

```
  .......
diastole(pd4_rs_3_ABP,3406,-882,ps3_rs_3_ABP,632,612,766,1516,4).
systole(ps4_rs_3_ABP,3558,-279,pd4_rs_3_ABP,603,152,764,1514,-29).
  ......
```

FIG. 4 – *Exemple de données pression pour un rythme sinusal*

2.4 Extraction de connaissances

La méthode choisie pour extraire les connaissances est la programmation logique inductive. Plus précisément, nous avons utilisé un logiciel nommé ICL (De Raedt et al 1995). Cette méthode nous permet d'obtenir des règles discriminantes en Prolog caractérisant un ensemble fini d'arythmies cardiaques. Le choix de la logique du premier ordre rend les règles facilement interprétables. Les données utilisées pour l'apprentis-

sage dépendent du choix de la technique de fusion. Deux choix parmi ceux présentés en section 1 sont possibles.

- Le premier choix consiste à agréger les informations provenant des différentes sources pour former une base d'apprentissage. Pour chaque enregistrement disponible, les données de pression et les données concernant les deux voies d'ECG sont agrégées. Le résultat de l'apprentissage sera donc un ensemble de règles comportant des événements provenant des différentes voies. Cette méthode pose le problème du choix du biais de langage pour l'apprentissage (dans le cas d'ICL, il s'agit d'un programme écrit en DLAB (De Raedt et al 1997)) puisque celui-ci devra mêler à la fois des connaissances sur l'électrocardiogramme et sur la voie hémodynamique. En outre, l'agrégation des données augmente la taille du langage et de la base de connaissance et donc, la complexité de l'espace de recherche des hypothèses.
- La deuxième solution est d'apprendre séparément les règles sur chaque source puis fusionner les règles obtenues. Cette méthode pose le problème de l'écriture de règles de "bonne fusion" permettant de synchroniser certains éléments des différentes règles. Dans l'application qui nous intéresse, ces règles peuvent être des contraintes temporelles entre évènements (la diastole se place entre le QRS et l'onde P, la systole se place avant l'onde P mais après la diastole, le délai entre QRS et diastole est compris entre 200 et 300 ms, etc.). Il est cependant plus facile d'obtenir les règles de diagnostic dans ce cas de figure puisque les biais d'apprentissage sont a priori moins complexes et l'apprentissage est plus rapide compte tenu du nombre limité de données.

3 Premiers résultats

3.1 Apprentissage simultané

Pour le moment, aucune expérience n'a encore été réalisée à ce sujet. La difficulté provient du biais d'apprentissage. En effet, de même qu'il est difficile d'écrire des règles de "bonne fusion" dans le cas d'apprentissages indépendants, ici, il est difficile de trouver un biais de langage qui inclut les différentes sources de manière cohérente.

3.2 Apprentissages indépendants

Des apprentissages indépendants concernant les différentes voies de l'ECG ont été réalisés sur une base de donnée différente, sans données multisources, dans le cadre du projet Calicot (Quiniou et al 2001). Les apprentissages effectués sur la base MIMIC (et en particulier le biais d'apprentissage utilisé pour les voies ECG) sont donc fortement inspirés de ces premières expériences. La Figure 5 présente un exemple de règles apprises pour les arythmies *tv* et *tsv* sur la voie V2 de l'électrocardiogramme. La première des règles exprime le fait qu'une tachycardie ventriculaire peut être diagnostiquée si 3 QRS anormaux consécutifs sont observés (sans onde P intercalée). La deuxième règle exprime le fait qu'une tachycardie supra-ventriculaire peut être discriminée (par rapport aux six

autres arythmies étudiées) si deux intervalles RR très courts (*rr1 : short*) apparaissent entre deux ondes P normales et deux QRS normaux .

On remarque que ces règles sont très dépendantes du choix de l'ensemble des classes d'apprentissage. En effet, dans les premières expériences menées dans le cadre du projet Calicot, nous avons montré que l'arythmie ventriculaire "bloc de branche gauche" peut être caractérisée exactement de la même manière qu'une tachycardie ventriculaire. Si l'on ajoutait des exemples décrivant une telle arythmie dans l'ensemble de données provenant de la base MIMIC, la règle produite ne serait plus discriminante, il faudrait procéder à un nouvel apprentissage.

```
class(tv) :-
   qrs(R0, abnormal, _), qrs(R1, abnormal, R0),
   qrs(R2, abnormal, R1).

class(tsv) :-
   qrs(R0,normal, _), p_wav(P1, normal, R0),
   qrs(R1, normal, P1), rr1(R0,R1, short), p_wav(P2, normal, R1),
   qrs(R2, normal, P2), rr1(R1,R2, short).
```

FIG. 5 – *Exemple de règles apprises sur une voie V2 de l'ECG*

L'onde P est nettement moins visible sur la voie V5 du fait de l'emplacement de l'électrode fournissant ce signal. Nous avons donc décidé de n'apprendre les règles sur les données provenant de cette voie, qu'en fonction des complexes QRS. Un exemple d'apprentissage effectué pour les arythmies *tv* et *tsv* est donné Figure 6. Nous remarquons que la règle apprise pour la *tv* est semblable à celle apprise sur la voie V2. La règle apprise pour la *tsv* est cependant plus compacte que celle apprise sur la voie V2.

```
class(tv) :-
   qrs(R0, abnormal, _, _),
   qrs(R1, abnormal, _, R0),
   qrs(R2, abnormal, R1, R1).

class(tsv) :-
   qrs(R0,normal, _, _),
   qrs(R1, normal, _, R0),
   rr1(R0, R1, short).
```

FIG. 6 – *Exemple de règles apprises sur une une voie V5 de l'ECG*

Les apprentissages concernant la voie hémodynamique seule ont produit, jusqu'à présent, des règles complexes et non totalement discriminantes sur la majorité des apprentissages. Un exemple d'apprentissage pour les arythmies *tv* et *tsv* est donné Figure 7. Il existe au moins deux façons (deux règles) pour caractériser une tachycardie

```
class(tv) :-
diastole(Dias0,short,_), systole(Sys0,short,Dias0),
diastole(Dias1,short,Sys0), systole(Sys1,normal,Dias1),
ds1(Dias1,Sys1,normal),
diastole(Dias2,normal,Sys1), systole(Sys2,normal,Dias2),
diastole(Dias3,normal,Sys2), systole(Sys3,normal,Dias3),
amp_ss(Sys2,Sys3,pos,normal).

class(tv) :-
diastole(Dias0,normal,_), systole(Sys0,short,Dias0),
diastole(Dias1,short,Sys0), systole(Sys1,normal,Dias1),
amp_dd(Dias0,Dias1,pos,long),
diastole(Dias2,normal,Sys1), systole(Sys2,normal,Dias2),
dd1(Dias1,Dias2,normal), ds1(Dias2,Sys2,normal),
diastole(Dias3,normal,Sys2), systole(Sys3,normal,Dias3),
amp_ss(Sys2,Sys3,pos,normal).

class(tsv) :-
diastole(Dias0,normal,_), systole(Sys0,normal,Dias0),
diastole(Dias1,normal,Sys0), systole(Sys1,normal,Dias1),
amp_dd(Dias0,Dias1,neg,normal),
ss1(Sys0,Sys1,short),
ds1(Dias1,Sys1,long).
```

FIG. 7 – *Exemple de règles apprises sur une voie hémodynamique*

ventriculaire : soit par une succession de quatre diastoles et quatre systoles avec une différence d'amplitude petite (*short*) entre les deux premiers enchaînement diastole-systole, soit par une systole de moindre amplitude (qui se traduit par des différences d'amplitude diastole-systole et systole-diastole petite et un intervalle diastole-diastole long (*amp_dd : long*)). La tachycardie supra-ventriculaire est caractérisée par un intervalle systole-systole (*ss1*) très court alors que l'intervalle diastole-systole (*ds1*) est long.

Des travaux sont en cours pour améliorer ces résultats en jouant notamment sur le biais de langage utilisé ainsi que sur le choix des arguments utilisés pour caractériser l'arythmie.

L'apprentissage indépendant produit des règles exploitables indépendamment ou conjointement suivant les contextes : en cas de signaux clairs, une seule voie peut s'avérer suffisante ; en cas de signaux bruités, plusieurs règles peuvent être nécessaires à la reconnaissance, certaines peuvent même s'avérer inexploitables si le bruit est trop important ou si les électrodes sont défaillantes. Par contre, si l'on veut exploiter toutes les informations simultanément, il faut définir des relations entre les événements se déroulant sur la voie hémodynamique et les événements se déroulant sur l'électrocardiogramme afin de fusionner les règles apprises. Dans le cas qui nous intéresse, il peut être possible de fusionner les règles par un nouvel apprentissage. En effet, les règles obtenues peuvent être fusionnées naïvement dans un premier temps (en faisant par exemple une conjonction des prédicats appartenant au corps des règles) puis en apprenant, à partir des exemples liés à la classe, une règle plus spécifique fusionnant les prédicats entre eux. Cette méthode a l'avantage de diminuer l'espace de recherche : en effet, le treillis de recherche est borné par les clauses comprenant l'une la conjonction et l'autre la disjonction des prédicats contenus dans le corps des règles "indépendantes". Le vocabulaire utilisé est lui aussi limité à celui utilisé dans les deux règles que l'on cherche à fusionner.

Discussion et conclusion

Nous avons dans un premier temps exprimé la problématique liée à l'extraction de connaissances à partir de données multisources. Puis, nous avons exposé le cas pratique de l'apprentissage de règles caractérisant des arythmies cardiaques à partir de données provenant de plusieurs capteurs utilisés en médecine. Les premiers résultats obtenus dans le cadre de l'apprentissage indépendants de règles ont été exposés, ainsi que des éléments de recherche concernant la fusion de ces règles indépendantes.

Les règles apprises sur la voie de pression seule sont peu corrélées avec les connaissances médicales usuelles, et donc difficile à valider directement par un expert. En effet, les médecins sont peu habitués à repérer des arythmies sur la voie hémodynamique, ils se servent plutôt de ces données pour apprécier la sévérité d'une arythmie ou pour la localiser. Une idée possible serait alors de regrouper certaines classes en fonction de la localisation de l'arythmie dans le cœur (sinusal, ventriculaire, supra-ventriculaire) pour obtenir des résultats moins précis mais plus fiables. En outre, on pourrait apprendre par PLI non pas la caractérisation de l'arythmie cardiaque (celle-ci pourrait être donnée par l'ECG), mais un indice de gravité de cette arythmie ce qui compléterait

les informations fournies par l'ECG.

On se rend compte que quelque soit le type de fusion choisi, la fusion de données hétérogènes et indépendantes demande beaucoup de connaissance au préalable sur les relations possibles entre les données. On peut donc se demander comment utiliser ce type de méthode de façon générique sur des données a priori peu usitées et peu connues du corps médical.

Cette étude préliminaire servira de base à un travail approfondi dont le but est de mettre en évidence l'intérêt de la fusion de données pour la caractérisation des arythmies cardiaques.

Références

I. Bloch et A. Hunter et Alain Appriou et A. Ayoun et Salem Benferhat et P. Besnard et L. Cholvy et R. Cooke et F. Cuppens et D. Dubois et H. Fargier et M. Grabisch et R. Kruse et J. Lang et S. Moral et H. Prade et A. Saffiotti et P. Smets et C. Sossai. (2001), Fusion: General concepts and characteristics, International Journal of Intelligent Systems, 16, pp 1107-1134.

G. Carrault et M.-O. Cordier et R. Quiniou et F.Wang (2003), Temporal Abstraction and Inductive Logic Programming for arrhythmia recognition from ECG, Artificial Intelligence in Medicine,28, pp 231-263.

L. Cholvy (2003), Information Evaluation in fusion: a case study, ECSQARU-03 Workshop: Uncertainty, Incompleteness, Imprecision and Conflict in Multiple Data Sources .

L. De Raedt et W. Van Laer (1995), Inductive constraint logic, Lecture Notes in Computer Science,997, pp 80-94.

L. De Raedt and L. Dehaspe, Clausal discovery, Machine Learning, 26, pp 99-146.

D. Dubois et M. Grabisch et H. Prade et P. Smets (2001), Using the transferable belief model and a qualitative possibility theory approach on an illustrative example: the assessment of the value of a candidate, International Journal of Intelligent Systems,16,pp 1183-1192.

A. I. Hern et G. Carrault et F. Mora et L. Thoraval et G. Passariello et J. M. Schleich (1999), Multisensor fusion for atrial and ventricular activity detection in coronary care monitoring, IEEE transactions on biomedical engineering,46(10),1186-1190.

G. B. Moody et Roger G. Mark (1997), A Database to Support Development and Evaluation of Intelligent Intensive Care Monitoring, Harvard-MIT Division of Health Sciences and Technology, Cambridge, MA, USA, Cardiology Division, Beth Israel Hospital, Boston.

R. Quiniou et M.-O. Cordier et G. Carrault et F. Wang (2001), Application of ILP to cardiac arrhythmia characterization for chronicle recognition, Proceedings of ILP, pp 193-215.

L. Thoraval et G. Carrault and J. Schleich et R. Summers et M. Van de Velde et J.Diaz (1997), Data fusion of electrophysiological and haemodynamic signals for ventricular rhythm tracking, IEEE Engineering in Medicine and Biology Magazine,16(6), pp 48-55.

C. Wemmert AND P. Gancarski (2002), A Multi-View Voting Method to Combine Unsupervised Classifications, IASTED Artificial Intelligence and Applications.

Summary

This article describes the main problems encountered while extracting knowledge from multi-sensor data. It also suggests an application to cardiology. The aim of this application is to learn efficient rules describing cardiac arrhythmias from cardiac comportemental data such as an electrocardiogram or the measure of the arterial blood pressure. Rules are obtain thanks to an Inductive Logic Programming learning algorithm in two different ways. Firstly from the whole set of available data. Secondly from the fusion of the data obtain from each sensor. Advantages and drawbacks of the two methods are then analyzed.

Classification multisource via la théorie des croyances

Anis BEN AÏSSA*, Nour-Eddin EL FAOUZI*, Eric LEFEVRE**

*Laboratoire d'Ingénierie Circulation Transports
INRETS - ENTPE / LICIT
25, Avenue François Mitterrand Case 24
69675 Bron Cedex
Anis.Ben-aissa@inrets.fr nour-eddin.elfaouzi@inrets.fr
**Laboratoire LGI2A
Université d'Artois
Faculté des Sciences Appliquées
62400 Béthune
eric.lefevre@iut-geii.univ-artois.fr

Résumé. La classification en présence de plusieurs sources de données hétérogènes est abordée comme un problème de fusion de classifieurs. Dans ce travail, la méthodologie de la fusion, opérée par la théorie des croyances, est présentée et plusieurs solutions techniques sont abordées. Les résultats de ces différentes propositions sont illustrés dans le cadre de l'estimation des temps de parcours sur un réseau routier. Ces schémas montrent la propension de la fusion de classifieurs à améliorer la qualité, en terme de taux de bien classés.

1. Introduction

Les méthodes de classification et plus particulièrement les méthodes de discrimination tentent de construire des règles de décision permettant d'affecter un élément au groupe dont il est proche, au sens d'une certaine distance, connaissant uniquement le vecteur de ses caractéristiques. Ces règles de décision sont élaborées de telle sorte qu'elles produisent un minimum d'erreurs lorsqu'elles seront appliquées dans une approche prédictive. Plusieurs approches ont été proposées pour répondre à ces impératifs. A titre d'exemple, on peut citer les méthodes discriminantes (Mardia et al., 1979 ; McLachlan, 1992), méthodes des plus proches voisins (Fix et Hodges, 1951), méthodes neuronales (Ripley, 1996), arbres de classification (Breiman et al., 1984) et plus récemment les méthodes basées sur la théorie de l'apprentissage statistique (Vapnik 2000). Le lecteur intéressé pourra trouver une revue et comparaison de ces méthodes dans Boutou et al. 1994.

Ces différentes propositions reposent sur l'exploitation de l'information sur les éléments à classer sous forme d'un tableau de données. Avec l'amélioration des systèmes de recueil et de collecte d'informations et l'apparition de nouvelles sources de données, les informations sont le plus souvent multiformes et multisources interdisant l'application directe de telles propositions. On est alors face à un problème de classification dans une configuration multisource. L'exploitation des spécificités de chacune des sources et la prise en compte de leurs imperfections sont à l'origine des difficultés opérationnelles et nécessitent le développement d'un cadre méthodologique spécifique.

Un exemple type de ce problème de classification est celui de l'estimation des temps de parcours sur un axe urbain. Traditionnellement, la notion du temps de parcours est utilisée pour répondre aux préoccupations primaires de la gestion du trafic, qui consistent à apporter la réponse la plus satisfaisante aux besoins de déplacements. Le temps de parcours est alors utilisé comme indicateur permettant de qualifier la *qualité de service* du réseau de transport concerné. Le temps de parcours utilisé dans ce contexte est principalement de type instantané.

La réponse à ce besoin primaire de gestion de flux de trafic fait apparaître, sous la demande forte des usagers, un besoin secondaire qui concerne l'information routière. Dans ce cas, le temps de parcours constitue une information, associé à la notion d'impédance (ou coûts) liée aux itinéraires du réseau ou/et à un indicateur de congestion.

Face à ces diverses utilisations, la question d'une estimation avec une précision acceptable du temps de parcours se pose. Ce problème est particulièrement ardu en milieu urbain, où l'on doit faire face à un certain nombre de difficultés théoriques, techniques et méthodologiques. Ainsi, pour connaître l'état du trafic sur un axe urbain, les capteurs classiques utilisés pour mesurer les conditions de circulation s'avèrent inefficaces dans certaines circonstances. Avec l'apparition de nouveaux moyens de mesure (caméras, moyens de localisation de type GPS ou téléphones cellulaires …), on recourt donc de plus en plus à d'autres sources de données visant à compléter l'information fournie par les moyens de mesure classiques et, en conséquence, à améliorer la qualité de l'estimation du temps de parcours. Le problème de l'estimation du temps de parcours devient alors un problème typique de fusion de données.

La collecte des données par les véhicules traceurs en complément des données de capteurs au sol, permettrait de fournir une meilleure image (globale et complète) de l'état du trafic et, par la même, de faire face aux imperfections des mesures (données manquantes, aberrantes…). Cette collecte, de plus en plus accessible, notamment avec le développement récent de l'usage du téléphone cellulaire et plus généralement celui des réseaux UMTS (*Universal Mobile Telecommunications System*), laisse entrevoir la mise en place de systèmes de gestion de trafic économiquement bon marché, utilisant conjointement des données collectées par des véhicules tests et des données issues de capteurs au sol traditionnels.

Les propriétés de complémentarité et de redondance de ces deux sources de données peuvent donc être mises à profit afin d'élaborer une solution multisources pour le problème d'estimation du temps de parcours en milieu urbain. De cette approche multisources sera exigée d'exploiter au mieux les avantages de chacune des sources d'information, tout en essayant de pallier leurs limitations individuelles respectives.

Le présent article portera sur le problème de classification multisource pour l'estimation du temps de parcours. Il s'agit essentiellement de la fusion des classifieurs basées sur chacune des sources en utilisant la théorie des croyances.

2. Fusion de classifieurs

Dans cette section, après une présentation succincte du problème générique que l'on cherche à résoudre et de la théorie des croyances, nous présenterons trois schémas de fusion de classifieurs basés sur cette théorie et l'apprentissage statistique.

2.1. Problème générique

Le problème générique de la classification est la détermination de la classe d'appartenance d'un élément sur la base de la connaissance seule de ses caractéristiques donnée par un vecteur x ∈ R^n, $x = \{x_1, \ldots, x_n\}$. Ce vecteur est assimilé à un échantillon issu d'une population $\wp$ constituée de r classes $c_1, \ldots, c_r$. On appelle donc classifieur tout outil de reconnaissance qui reçoit un vecteur de mesures x en entrée et donne des attributs de la classe de cette mesure x. Plus précisément, soient un vecteur $x \in \square^{n\blacksquare}$ et un ensemble de classes $\Lambda = \{c_1, \ldots, c_r\}$ un classifieur δ est fonction définie par :

$$\delta : R^n \to [0.1]^r$$
$$x \mapsto \delta(x) = (\delta_1(x), \ldots, \delta_r(x))$$

Les composantes $\delta_h ; h = 1, \ldots, r$ peuvent être vu comme des estimations des probabilités a *posteriori* des classes, pour x donné. Autrement dit, $\delta_h(x) = P(y = h|x)$ avec et y désigne le label des classes et à valeurs dans Λ.

La décision de δ peut être durcie de sorte qu'un indice net de classes dans Λ soit assignée à x. Ceci est typiquement fait par la règle du maximum d'adhésion (règle bayésienne) :

$$\begin{cases} \delta_k(x) = 1 \text{ si } \quad k = \arg\min_{i=1,\ldots,r} \{\delta_i(x)\} \\ \delta_h(x) = 0 \quad \text{si} \quad h \neq k \end{cases}$$

On notera par la suite $\delta(x) = k$ lorsque $\delta_k(x) = 1$.

Dans le cas où plusieurs classifieurs sont disponibles, l'agrégation de ces classifieurs permet le plus souvent d'améliorer la qualité de la classification en terme de taux de bien classés. Cette propension à l'amélioration est d'autant plus importante que les classifieurs soient complémentaires. Le plus souvent les différents classifieurs reposent soit sur des différentes méthodologies (Kuncheva et al., 2001), soit sont issus de sources différentes (El Faouzi, 2000a).

Les schémas d'agrégation, quant à eux, vont de la simple moyenne arithmétique (Alexandre et al., 2000) jusqu'aux approches basées sur les théories de l'incertain (approche bayésienne et réseaux du même nom) et de l'imprécis (approches crédibiliste et possibiliste), en passant par des schémas fondés sur les réseaux de neurones, l'approche des k plus proches voisins (Xu et al., 1992) et des techniques de rééchantillonnages (Breiman, 1998 ; El Faouzi, 1997).

Le travail rapporté dans le présent article s'inscrit dans le cadre de la fusion de classifieurs issus de sources différentes et l'approche d'agrégation utilisée repose sur la théorie des crédibilités.

2.2. Théorie des croyances

Cette théorie, basée sur les travaux de Dempster (1967, 1968), a été formalisée par Shafer (1976). Cette théorie, dont le principe est de pouvoir associer à un événement une probabilité inférieure et une probabilité supérieure, constitue une généralisation de l'approche bayésienne.

L'idée de base de cette théorie est qu'à partir de l'ensemble des classes Λ, appelé cadre de discernement, on définit sur l'ensemble des parties de Λ une fonction de masse m ayant ses valeurs dans l'intervalle $[0,1]$ vérifiant $m(\varnothing) = 0 \quad et \quad \sum_{A \subseteq \Lambda} m(A) = 1.$

La quantité $m(A)$ s'interprète comme la part de croyance placée strictement sur A. Cette quantité se différencie d'une probabilité par le fait que la totalité de la croyance est répartie non seulement sur les classes singletons mais aussi sur les classes composites. La plupart du temps la construction de cette fonction dépend de l'application envisagée (Dromigny-Badin, 1998). Toutefois, quelques approches ont vu le jour afin de généraliser l'affectation des croyances (Denoeux, 2000 ; Appriou 1991).

Une fois le jeu de masse défini pour chacune des sources en présence, la décision finale est élaborée d'une part en combinant les masses par une règle propre à cette théorie, appelée règle de combinaison orthogonale de Dempster et d'autre part en adoptant une des multiples règles de décision qu'offre la théorie des croyances.

Le principe de la règle de combinaison est relativement simple. Pour le besoin de l'exposé, considérons deux sources S_1 et S_2 auxquelles sont associées deux fonctions de masses m_1 et m_2. Si u et v désignent les sorties des deux classifieurs δ^1 et δ^2 issus des sources S_1 et S_2 respectivement. La masse associée à leur jonction $w = u \cap v$ est proportionnelle à $m_1(u) \times m_2(v)$. Comme plusieurs paires d'ensembles u et v peuvent avoir w pour intersection, la masse associée à w doit alors intégrer les produits des masses de ces ensembles. Cependant, si u et v sont disjoints, la masse associée à leur intersection

est portée sur l'ensemble vide. Ceci violerait l'hypothèse imposée d'une masse nulle de l'ensemble vide. Cette difficulté peut être contournée par une simple normalisation.

$$m_1 \oplus m_2(w) = \begin{cases} 0 \text{ si } w = \varnothing \\ \dfrac{1}{1-\lambda_c} \sum_{u \cap v = w} m_1(u) \times m_2(v) \text{ si } w \neq \varnothing \end{cases}$$

où $\lambda_c = \sum_{u \cap v = \varnothing} m_1(u) \times m_2(v)$ représente le conflit entre les sources, lequel est pris en compte dans la combinaison sous forme d'un facteur de normalisation. Cette règle de combinaison a été critiquée dans plusieurs travaux dont (Yager, 1987). Ainsi, d'autres opérateurs alternatifs à cette somme orthogonale ont été étudiés dans la littérature : opérateurs non normalisés (Smets, 1990), opérateurs disjonctifs (Dubois et al. 1988), généralisation d'opérateurs (Lefevre, 2001),…

La théorie des croyances propose plusieurs critères de décisions. Nous reprenons certaines de ces règles ici. Pour plus d'informations le lecteur pourra se reporter à (Denoeux, 1997).

1. **Règle 1 :** La décision combinée est la classe ayant la valeur de crédibilité maximale :

$$\theta_k = \arg\min_{\theta_i, i=1,..,r} \left[Cr(\theta_i) = \sum_{c_j \subseteq \{\theta_i\}} m(c_j) \right]$$

La crédibilité *Cr* constitue le degré de croyance minimal que l'on attribue à une classe.

2. **Règle 2 :** La décision combinée est la classe ayant la valeur de plausibilité maximale :

$$\theta_k = \arg\min_{\theta_i, i=1,..,r} \left[Pl(\theta_i) = \sum_{c_j \cap \{\theta_i\} \neq \varnothing} m(c_j) \right]$$

$$= \arg\min_{\theta_i, i=1,..,r} \sum_{\{\theta_i\} \subseteq c_j} m(c_j)$$

La plausibilité *Pl* constitue le degré de croyance maximal que l'on attribue à une classe.

3. **Règle 3 :** La décision combinée est la classe ayant la valeur de probabilité pignistique maximale :

$$\theta_k = \arg\min_{\theta_i, i=1,\ldots,r} \left[Pg(\theta_i) = \sum_{c_j \subseteq \{\theta_i\}} \frac{m(c_j)}{|c_j|} \right]$$

Ce critère consiste à choisir la classe simple la plus probable en équirépartissant la masse placée sur chaque classe composée sur les classes simples qui la composent.

C'est cette dernière règle qui sera utilisée par la suite car elle garantie que la classe retenue après fusion soit une classe singletons, i.e. élément de Λ.

2.3. Schémas de fusion de classifieurs

Les schémas de fusion de classifieurs que l'on va introduire reposent essentiellement sur la prise en compte des erreurs des classifieurs individuels. Les erreurs de chaque classifieur δ^h sont usuellement consignés dans la matrice de confusion donnée par :

$$M_h = \begin{pmatrix} n_{11}^{(h)} & \cdots & n_{1n}^{(h)} \\ \vdots & \ddots & \vdots \\ n_{r1}^{(h)} & \cdots & n_{rr}^{(h)} \end{pmatrix} \quad h = 1,\ldots,\ell$$

La ligne i correspond à la classe c_i de l'apprentissage et la colonne j correspond à la classe reconstituée par la classifieur δ^h, i.e. $\delta(x) = j$. Cette matrice est obtenue par apprentissage sur un échantillon test et peut être considérée comme la connaissance *a priori* sur les performances du classifieur δ^h. Les éléments diagonaux sont les pourcentages des concordances entre les classes reconstituées par le classifieur et les classes de références. En d'autres termes, ceci représente le nombre de fois où la classe de référence et la classe reconstituée par le classifieur coïncident. Les éléments hors diagonale donnent quant à eux les pourcentages de discordances (i.e. de confusion).

A partir de cette matrice, on définit le taux de reconnaissance $\left(\tau_{rec}^{h}\right)$ et le taux de confusion $\left(\tau_{conf}^{h}\right)$ par :

$$\tau_{rec}^{h} = \frac{\sum_{i=1,..r} n_{ii}^{(h)}}{\sum_{i,j=1,..,r} n_{ij}^{(h)}} \qquad \text{et} \qquad \tau_{conf}^{h} = 1 - \tau_{rec}^{h}$$

Nous supposons que l'on dispose d'une matrice de confusion par classifieur et nous présentons dans les sections suivantes trois approches utilisées pour l'élaboration des masses à partir de ces matrices de confusion.

La première approche, proposée initialement par Xu et al. (Xu et al.,1992) constitue à la fois une méthode de référence en fusion de classifieurs et le point de départ de ce travail. Nous avons cherché à en améliorer les performances par une exploitation optimale de la matrice de confusion.

2.3.1 Premier schéma :

Dans ce premier schéma, le taux de reconnaissance et de confusion sont utilisés pour définir les masses de croyances. On définit ce jeu de masse de la façon suivante :

$$(\text{SF.1}) \quad \begin{cases} m_k(c_i) = \tau_{rec}^{h} \\ m_k(\overline{c}_i) = \tau_{conf}^{h} \end{cases} \quad \forall i = 1,\ldots,r$$

Pour cette première approche, le support des masses de croyance est formé des classes et de leurs complémentaires. Par ailleurs, ce schéma impose qu'un même jeu de masses soit associé à une classe abstraction faite de la classe retenue par le classifieur, i.e. $\delta^h(x) = s_h$, $s_h \in \Lambda$.

2.3.2 Deuxième schéma

Cette approche constitue une première amélioration de l'approche précédente et consiste à mieux exploiter les informations contenues dans la matrice de confusion en conditionnant la définition des masses par la sortie de chaque classifieur. En d'autres termes, si

$\delta^h(x) = j$, on définit un taux de reconnaissance sachant que le classifieur δ^h à retenu la classe j par :

$$\tau^h_{rec|j} = \frac{n^{(h)}_{jj}}{\sum_{i,j=1,..r} n^{(h)}_{ij}} \qquad \text{et} \qquad \tau^h_{conf|j} = 1 - \tau^h_{rec|j}$$

et le jeu de masse est obtenu en substituant au taux de reconnaissance et au taux de confusion leurs versions conditionnelles :

$$(\text{SF.2}) \quad \begin{cases} m_k(c_i) = \tau^h_{rec|i} \\ m_k(\overline{c_i}) = \tau^h_{conf|i} \end{cases} \quad \forall i = 1,\ldots,r$$

2.3.3 Troisième schéma :

Cette troisième approche consiste à un conditionnement dans lequel les éléments de la colonne $\delta^h(x) = j$ vont être utilisés pour élaborer un jeu de masse associé aux r classes contenues dans Λ.

$$(\text{SF.3}) \quad \begin{cases} m_k(c_i) = n^{(k)}_{ij} \Big/ \sum_{i=1,..,r} n^{(k)}_{ij} \\ m_k(\Lambda) = \sum_{i=1}^{r} n^{(k)}_{ij} - n^{(k)}_{jj} \end{cases} \quad \forall i, j = 1,\ldots,r$$

Cette fonction de croyance n'étant pas normalisée, un coefficient de normalisation est appliqué afin de respecter les propriétés des jeux de masses. Nous obtenons alors la fonction de croyance suivante :

$$m_k(A) = m_k(A) \Big/ \sum_{B \subseteq \Lambda} m_k(B) \quad \forall A \subseteq \Lambda$$

Les deux derniers schémas constituent deux propositions originales développés dans le cadre de ce travail que nous avons cherché à comparer avec la méthode de Xu et al.,1992.

Une fois les jeux de masses associés à chaque classifieurs élaborés, la fusion est opérée en faisant appel à la combinaison orthogonale de Dempster. La décision finale est prise en utilisant la règle du maximum de probabilité pignistique.

3. Mise en oeuvre

3.1. Données

Les données utilisées dans cette étude ont été recueillies à l'issue de la compagne de mesures réalisée sur un axe urbain de Toulouse. La compagne de mesure s'est déroulée pendant huit jours, du 16 au 25 novembre 1993, à raison de trois heures par jour, réparties en deux périodes : 14h30 – 16h et 16h30 – 18h. Le choix de ces périodes est dicté par le besoin de rendre compte de conditions de trafic variées. L'axe urbain sur lequel a été réalisée cette expérimentation mesure deux kilomètres environ et comporte 4 tronçons par sens de circulation. Seul l'itinéraire correspondant au sens sud–nord a été retenu dans le cadre de cette étude.

L'enquête de mesure a consisté en un recueil de données issues de plusieurs sources d'information. Nous en avons considéré trois qui ont été disponibles simultanément :

- **Source « capteurs » :** cette source est composée d'un réseau de capteurs à boucles électromagnétiques (une douzaine), qui délivrent des données de trafic, principalement le débit et le taux d'occupation, toute les 12,5ms , soit environ de 115200 observations .
- **Source « traceurs » :** Quatre véhicules disposant de capteurs embarqués (dits traceurs), qui fournissent des données cinématiques et donc leurs temps de parcours sur chaque tronçon. Ces véhicules avaient comme consigne de parcourir l'axe étudié pendant les deux périodes temporelles suscitées (74 800 réalisations).
- Un recueil exhaustif des temps de parcours sur cet axe vient compléter les sources d'information dont on dispose. Ce recueil est réalisé par la reconnaissance des plaques minéralogiques des véhicules (110 400 réalisations)

Les deux premières sources servent à bâtir deux classifieurs que nous souhaitons fusionner, alors que la troisième fournit le temps de parcours de référence que l'on cherche à reconstituer.

Une particularité du problème d'estimation de temps de parcours est son double dimensionnement à la fois spatial et temporel. Une première étape consiste à recaler temporellement et spatialement les différentes données disponibles à savoir les temps de parcours.

Dans le cadre de cette étude, on dispose d'un recueil d'informations sur le temps de parcours sur un axe routier soit par mesure directe (cas des véhicules traceurs) soit par estimation (cas des capteurs à boucles). Une première tâche a consisté à une correspondance temporelle des différentes données en les agrégeant sur une période temporelle de six minutes.

A l'issue de cette étape, on obtient 240 observations pour les capteurs, 156 observations pour les traceurs et 230 pour les temps de parcours de référence.

La conversion des données trafic en temps de parcours est rendue nécessaire par l'un des principes de fusion qui veut que, pour pouvoir être combinées et comparées, les différentes informations doivent être manipulées dans un espace de représentation commun. Ainsi, la conversion des données issues de capteurs à boucles électromagnétiques est réalisée par la méthode proposée par Bonvalet et Robin-Prévallée, dite BRP (Bonvalet et Robin-Prévallée, 1987).

Cette méthode se fonde sur l'existence d'une relation linéaire entre le temps de parcours et le taux d'occupation sur chaque tronçon qui compose l'axe à étudier. Le temps de parcours associé à l'itinéraire est alors défini par la somme des temps de parcours sur les tronçons.

Nous disposons ainsi des temps de parcours élaborés soit à partir de données trafic soit à partir de données traceurs. A l'issue de cette phase de recalage, les temps de parcours résultants sont de dimensions spatiales et temporelles équivalentes.

Afin de transformer le problème de l'estimation en un problème de classification, un découpage préalable des temps de parcours en classes a été effectué. Dans cette application, nous avons choisi six classes de temps de parcours sur l'axe routier considéré. Elles sont définies à partir des temps de parcours de référence (enquête minéralogique).

Notons la complémentarité entre la source « capteurs » et la source « traceurs ». En effet, les données fournies par des capteurs au sol sont des mesures quasi exhaustives, c'est-à-dire couvrant l'ensemble des véhicules ayant empruntés le tronçon, avec un échantillonnage et une résolution temporelle excellente. Cependant, ces mesures sont très imprécises (principalement le taux d'occupation qui est très bruité…) avec un échantillonnage spatial qui dépend de la densité des capteurs. En effet, ces mesures ne représentent l'état du trafic qu'à l'endroit où le capteur est placé et non sur l'ensemble du tronçon.

À l'inverse, les données fournies par des véhicules munis de capteurs embarqués (véhicules tests), sont quant à elles, des mesures très précises et de qualité inégale avec une excellente couverture spatiale. Elles expriment l'état du trafic sur l'ensemble du tronçon. Cependant, elles sont non exhaustives, car ne couvrant qu'une partie des véhicules ayant emprunté le réseau pendant une période temporelle donnée.

3.2. Evaluation des résultats

Pour quantifier l'apport de la fusion, nous avons calculé trois taux de bien classés (exprimés en pourcentage) correspondant d'une part aux deux sources d'informations prises individuellement : informations issues du réseau de capteurs et par les véhicules traceurs et d'autre part au taux bien classés obtenu après fusion.

Ces différents taux ont été calculés à partir d'un échantillon de validation (échantillon n'ayant pas fait partie des données d'apprentissage) obtenu par tirage aléatoire.

Classifieurs individuels et fusionné	Schémas de fusion via la théorie des croyances Règle de max de probabilité pignistique		
	SF.1	SF.2	SF.3
Capteurs	33,91 %		
Traceurs	30,07 %		
Fusion	**33,91 %**	**33,91 %**	**42,17 %**

TABLE I. *Performances des différents classifieurs et de leurs fusions.*

On voit alors que l'application des trois schémas de fusion précédents au problème de l'estimation de la classe des temps de parcours montre des résultats variables d'une stratégie à l'autre.

On remarque que la fusion, basée sur les deux premiers schémas, ne fait guère mieux que le meilleur classifieur. La raison essentielle est que les performances de chacun des classifieurs ne sont pas correctement modélisées. Bien que le conditionnement introduit dans le second schéma tend à mieux appréhender les performances de ces classifieurs, il n'est cependant pas suffisant pour apporter une amélioration notable dans le processus de fusion. Le premier schéma qui est considéré comme une méthode de référence pour la fusion de classifieurs s'est donc révélée inefficace dans le cadre de notre application.

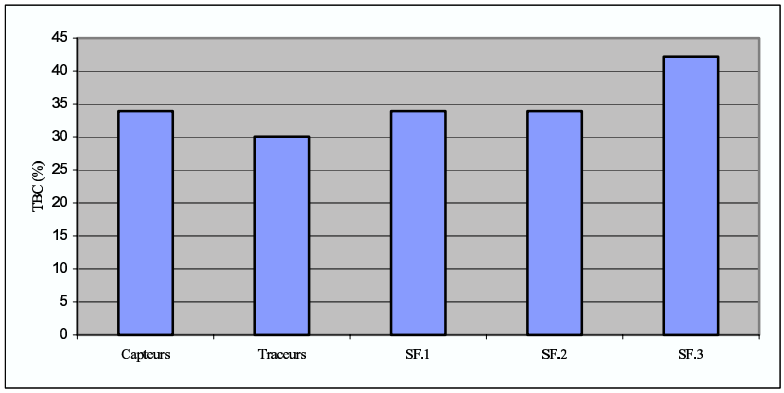


FIGURE I. *Performances de la fusion en terme de taux de bien classés.*

Le troisième schéma qui intègre toute la connaissance contenue dans la matrice de confusion tend à détecter correctement les vraies classes de temps de parcours. La fusion dans ce cas surclasse le meilleur classifieur avec une amélioration de l'ordre de 24,4 %.

4. Conclusion

L'objectif du travail effectué était de proposer un cadre méthodologique ainsi que des solutions au problème d'estimation de temps de parcours en présence de données issues de sources hétérogènes. Deux sources ont été considérées ici : des capteurs classiques de trafic constitués d'une boucle électromagnétique, qui permettent de mesurer le débit et le taux d'occupation et d'en déduire une estimation de temps de parcours moyen, et un échantillon réduit de véhicules traceurs, qui recueillent les temps de parcours qu'ils ont réalisé.

Abordant le problème de l'estimation du temps de parcours comme un problème typique de fusion de classifieurs et utilisant la théorie des croyances pour sa finesse de modélisation de la connaissance, nous avons utilisé trois méthodes différentes de fusion de classifieurs : la méthode de Xu (qui a constituée le point de départ de notre travail) et deux autres approches permettant de tirer meilleur profit de la matrice de confusion. Parmi ces méthodes, la troisième approche s'est avérée la plus efficace d'entre elles. Il est évident que l'on ne peut pas généraliser ces résultats car ils ont été obtenus dans le cadre d'une application spécifique. Une étude sur d'autres bases de données est envisagée.

Références

[Alexandre et al., 2000] Alexandre L, Campilho A., Kamel M. – «Combining independant and unbiased classifiers using weighted average». In : Proc. 15th Conf. On Pattern Recognition, vol 2. IEEE Press, 2000.

[Appriou, 1991] Appriou A. – «Probabilités et incertitude en fusion de données multi-senseurs», Revue Sceintifique et Technique de la Défense, 11, pp 27-40, 1991.

[Ben Aïssa, 2003] Ben Aïssa A – «Fusion de classifieurs pour l'estimation des temps de parcours : Théorie de l'évidence», Rapport de stage LICIT, N°0303, 2003, sous la direction de N.E. El Faouzi.

[Bonvalet, 1987] Bonvalet F., Robin-Prévallée Y. – Mise au point d'un indicateur permanent des conditions de circulation en Ile-de-France., Revue *T.E.C.*, N° 84-85, Septembre/Décembre, 1987.

[Breiman et al., 1984] Breiman L., Friedman J.-H., Olshen R., and Stone C. J., – « *Classification and regression trees*», The Wadsworth statistics/probability series, 1984.

[Breiman, 1998] Breiman L. – « Arcing classifiers», *Annals of Statistics*, 26, pp. 801-824, 1998.

[Dempster, 1967] Dempster A. P. – Upper and Lower Probabilities Induced by Multivalued Mapping. *Annals of Mathematical Statistics,* Vol. 38, pp. 325-339, 1967.

[Dempster, 1968] Dempster A.P. – «A Generalization of Bayesian Inference», *Journal of the Royal Statistical Society*, 30B, pp. 205-247, 1968.

[Denoeux, 1997] Denoeux T. – «Analysis of evidence-theoretic decision rules for pattern classification», Pattern Recognition, 30(7), pp. 1095-1107, 1997.

[Denoeux, 2000] Denoeux T. – «A neural network classifier based on Dempster-Shafer theory», IEEE Transactions on Systems, Man and Cybernetics, Part A, 30(2), 2000

[Dromigny-Badin, 1998] Drogmigny-Badin A. – «*Fusion d'image par la théorie de l'évidence en vue d'applications médicales et industrielles*», Thèse de doctorat : INSA Lyon, 1998.

[Dubois et Prade *et al.*, 1988] Dubois D. et Prade H. – «Representation and combination of uncertainty with belief functions and possibility measures», Computer Intelligent, 4, pp. 224-264, 1988.

[El Faouzi, 1997] El Faouzi N.E. – «Fusion linéaire d'estimateurs multiples : Méthodes de pondération et de rééchantillonnage. », *Rapport LICIT n°9701*, 1997.

[El Faouzi, 2000a] El Faouzi N.-E.,– «Fusion de données pour l'estimation des temps de parcours via la théorie de l'évidence», Recherche Transport Sécurité n° 68, 2000.

[El Faouzi, 2000b] El Faouzi N.-E. – «Fusion de données : Concepts et méthodes», Rapport de Synthèse n°2002, INRETS-LICIT, 2000.

[Fix and Hodges, 1951] Fix E., Hodges J. – «Discriminatory analysis, non parametric discrimination : consistency properties», *Technical report, Randolph Field, Texas : USAF School of aviation Medicine*, 1951.

[Kuncheva *et al.*, 2001] Kunchava L. I., Bezdek C. J., Duin R.P.W. – «Decision templates for multiple classifier fusion : On experimental comparison», *Pattern Recognition*, 34, pp.299-314, 2001.

[Lefèvre, 2001] Lefèvre E. – «*Fusion adaptée d'informations conflictuelles dans le cadre de la théorie de l'évidence*», Thèse de doctorat, INSA Rouen, 2001.

[Lu, 1996] Lu Y. – «Knowledge integration in a multiple classifier system», *Appl. Intell.*, 6, pp.75-86, 1996.

[Mac Lachlan, 1992] Mac Lachlan G. J. – «*Discriminant analysis and statistical pattern recognition*», Wiley, New York , 1992

[Mardia, 1979] Mardia K. V., J. T. Kent, J. M. Bibby – «*Multivariate Analysis*. », Academic Press Inc., San Diego, 1979.

[Ripley, 1996] Ripley B. D. – « *Pattern recognition and neural networks*», Cambridge University Press, Cambridge, New York, 1996.

[Rogova, 1994] Rogova G. – «Combining the results of several neural networks classifiers » , *Neural Networks*, 7(5), 1994.

[Shafer 1976] Shafer G. – «*A Mathematical Theory of Evidence », Princeton University Press*, Princeton, New Jersey, Etats-Unis, 1976.

[Smets, 1990] Smets P. – «The combination of evidence in the transferable belief model», *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 12(5), pp. 447-458, 1990.

[Xu et *al.,* 1992] Xu L., Krzyzak A., Suen C. Y. – «Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognation», *IEEE Transactions on Systems, Man and Cybernetics*, 22(3), 1992.

[Yager, 1987] Yager R.R. - «On the Dempster-Shafer framework and new combination rules. », *Information Sciences*, 41, pp. 93-138, 1987.

[Vapnik, 2000] Vapnik V. – « *The nature of statistical learning theory* », Springer, New York, 2nd Edition, 2000.

Intégration des connaissances du domaine pour la fouille de données complexes

Walid Ben Ahmed*,*, Mounib Mekhilef*, Michel Bigand**, Yves Page*****

** LGI – Laboratoire de Génie Industriel, Ecole Centrale Paris, Grande voie des Vignes 92295 Châtenay-Malabry cedex, France, http://www.pl.ecp.fr,{ walid, mekhilef@lgi.ecp.fr}*
*** Équipe de Recherche en Génie Industriel, Ecole Centrale de Lille, BP 48, 59651 Villeneuve d'Ascq cedex, France, http://www.ec-lille.fr, Michel.Bigand@ec-lille.fr*
****LAB (PSA-Renault), Laboratoire d'Accidentologie, de Biomécanique et d'études du comportement humain, 132, rue des Suisses-92000 Nanterre.*

RESUME. L'extraction de connaissances à partir de bases de données (ECB) est appliquée au domaine des accidents de la route. La problématique générale est l'intégration de données de différents types (données structurées, reconstructions d'accident, entretiens et photos) et issues de différentes sources d'expertise (médecine, psychologie, mécanique, ingénierie de la route, etc.). Face à cette complexité, l'intégration des connaissances expertes du domaine d'accidentologie s'avère fondamentale durant les différentes phases du processus d'ECB et surtout pendant la phase de préparation des données ainsi que celle d'interprétation des résultats. Dans cet article, nous présentons l'utilisation d'un Modèle multi-vue de Représentation de Connaissance en Accidentologie (MRCA) et nous montrons son apport dans la première et dernière phase d'un processus d'ECB.

ABSTRACT. The knowledge Discovery in Databases (KDD) is applied to road accident domain. The main issue animating our research is the integration of data having different types (structured data, accident reconstruction, interviews and pictures) and stemming from different domains (medicine, psychology, mechanics, road engineering etc.). Hence, to tackle this complexity, there is a pressing need for the accidentological knowledge integration in the KDD process and especially in the data preparation step and in the result interpretation step. This paper is aiming to present the use of a Knowledge Representation Model in Accidentology (KRMA) and to show how it allows to improve the first and the last step in a KDD process.

MOTS-CLÉS : Extraction de connaissance de bases de données, pré-traitement de données, intégration de données, incorporation des connaissances du domaine.
KEYWORDS : Knowledge Discovery in Database, data pre-processing, data integration, background knowledge incorporation

FDC-EGC, Clermont Ferrand 2004

1. Introduction

La compréhension du déroulement des accidents de la route permet de développer des solutions tendant à réduire le nombre ainsi que la gravité de ces accidents ; c'est l'un des objectifs de l'accidentologie. Le LAB[1] développe des études détaillées sur la scène des accidents : une équipe de spécialistes est envoyée immédiatement sur les lieux de l'accident, qui recherche tous les indices pouvant aider à expliquer ce qui s'est passé. Cette équipe est composée d'un psychologue, d'un expert du véhicule et d'un expert de la route.

Ces informations sont ensuite stockées dans des bases de données appelées EDA (pour Études Détaillées des Accidents), constituées de données sur le conducteur, l'environnement, le véhicule, mais aussi des photographies et des reconstructions cinématiques et comportementales réalisées par des experts à l'aide de logiciels spécialisés

l'exploitation de ces EDA est coûteuse en terme de temps et d'expertise étant donnée :

- Le nombre important d'accidents déjà stockés (>1000 accidents),
- Le nombre important de paramètres caractérisant chaque accident (900),
- La variété des types des données (données structurées, entretiens, reconstructions et photos),
- La complexité des interactions entre les composants du triptyque Conducteur-Véhicule-Environnement (CVE).

L'objectif est d'obtenir des représentations plus globales sous la forme de scénario-type d'accident (STA). Des spécialistes construisent des STA en s'appuyant sur leur expertise pour la reconnaissance de similitudes entre cas d'accidents, ces similitudes étant identifiées à partir d'une examen méthodique de chaque cas d'accident (Fleury et al., 2001) (Van Elslande et al., 1997). Cette démarche présente certaines difficultés :

- L'examen au cas par cas est coûteux en terme de temps,
- En partant des mêmes cas d'accidents, il est possible que des experts différents aboutissent à des représentations différentes et non partageables,
- Variabilité au cours du temps des constructions d'un même expert partant des même données,

Pour contourner ces problèmes et extraire les connaissances exploitable pour le développement des solutions de prévention, des méthodes et outils issus de l'Extraction des Connaissances de Base de données (ECB) peuvent être développés et appliqués. Dans ce domaine, certains travaux ont porté sur la modélisation des connaissances expertes pour développer des systèmes à base de connaissances pour l'aide au diagnostic et à l'analyse des accidents de la circulation routière (Dieng et al., 1997) (Dieng, 1997) (Alpay et al., 1996) (Belanger, 2000) (Mercantini et al., 1998) (Ferrandez et al., 1996) (Boury-Brisset et al. 2000). D'autres ont porté sur la construction d'une base de connaissances exploitable par les différents intervenants dans l'analyse des accidents en utilisant des techniques de capitalisation de connaissances (Boury-Brisset et al. 2000).

[1] Laboratoire d'Accidentologie, de Biomécanique et d'études de comportement humain

Dans un processus de fouille de données complexes telles que les données d'accidents de la route, le choix et l'application d'une technique de data mining parmi d'autres (classification, classement, règles d'association etc.) est une tâche difficile. Mais, cette étape n'est pas discutée dans ce papier. Nous nous intéressons aux deux étapes les plus complexes dans un processus d'ECB qui sont la préparation de données et l'interprétation des résultats.

Dans cet article, nous utilisons un Modèle de Représentation de Connaissances en accidentologie (MRCA) développé dans (Ben Ahmed et al., 2003). Ce modèle avait pour vocation d'intégrer des connaissances du domaine dans le processus d'ECB.

Dans la première partie de cet article, nous présentons la problématique de préparation de données, d'interprétation des résultats ainsi que celle d'intégration des connaissances du domaine. Dans la deuxième partie nous présentons brièvement le Modèle de Représentation de Connaissances en Accidentologie (MRCA). Dans la dernière partie, nous présentons deux cas d'étude pour montrer l'apport du MRCA aux niveaux de la première et la dernière phase d'un processus d'ECB.

2. Etat de l'art

2.1. *Préparation des données*

La première phase dans un processus d'ECB est la préparation des données (Fayyad et al., 1996). En effet, les données initiales peuvent être incomplètes, bruitées/aberrantes, et incohérentes (Han et al., 2000) (Pyle et al., 1999) (Famili et al., 1997). Un processus de pré-traitement et de préparation de données contient les étapes suivantes:

- ***Le nettoyage de données :*** Cette tâche consiste en la détection et la suppression des erreurs, du bruit et de l'incohérence de données pour améliorer leur qualité (Rahm et al., 2000),
- ***L'intégration de données :*** Cette tâche combine des données de sources multiples, détecte et résout des conflits de valeurs (Calvanese et al., 2001),
- ***La transformation de données*** : Cette tâche consiste en l'extraction/construction de nouvelles variables (features) pour fournir une nouvelle représentation des données adéquate à l'application, au domaine et à l'objectif de l'étude (Jagadish et al., 1997). Plusieurs méthodes telles que l'agrégation, la généralisation, la normalisation permettent l'extraction de ces variables (Han et al., 2000),
- ***La réduction de données*** : Le coût de mesure et l'exactitude des résultats d'une technique de data mining sont les deux raisons principales pour maintenir le nombre de données le plus faible possible (Jagadish et al., 1997). Mais cette réduction peut causer une perte d'information. Il s'agit donc de trouver un compromis. Une des stratégies de réduction de données est l'agrégation par les cubes de données, la compression de données, la discrétisation et la génération de hiérarchie (Han et al., 2000).

2.2. *Interprétation des résultats*

Bien que fondamentale pour réussir un processus d'ECB, l'interprétation et l'évaluation des connaissances découvertes sont les problématiques les moins étudiées dans la littérature. La majorité des travaux pour l'intégration des connaissances expertes a porté sur les techniques de visualisation de ces connaissances. Néanmoins, des travaux sur la visualisation et le data mining interactif ont été développés (Keim , 1996). Sinon, le travail d'interprétation et d'évaluation est généralement manuel et se base sur le travail de l'expert ce qui rend cette tâche difficile et coûteuse. L'interprétation des résultats de classification d'accident destinée à identifier une typologie discriminante est aujourd'hui réalisée au LAB à la fois sur la base de critères statistiques usuels et sur la base d'une expertise en accidentologie qui n'est pas complètement explicite et formalisée. Ceci peut rendre coûteux la comparaison de plusieurs résultats issus de l'application de différents algorithmes de classification.

2.3. *Intégration des connaissances expertes*

L'utilisation et l'intégration des connaissances expertes du domaine sont très importantes en data mining (Palmeri et al., 2000) pour découvrir de nouvelles connaissances cachées dans les données interprétables et utilisables.

Cette problématique est fréquemment étudiée en ingénierie de connaissances (Studer et al., 1998) pour le développement des PSM (Problem Solving Method) et la construction des Systèmes à Base de Connaissance (SBC) (KBS : Knowledge Based System) (Schreiber et al., 1994) (Angele et al. 1996) (Puerta et al., 1992) (Eriksson et al., 1995). Dans la méthode commonKADS (Schreiber, 1994) (Wielinga, 1994), par exemple, le but du Model Expert est d'intégrer, à travers les trois types de connaissance (domaine, inférence et tâche) les connaissances nécessaires pour résoudre un problème donné. Dans l'approche MIKE (Model-based and Incremental Knowledge Engineering) (Angele et al 1996) aussi, le rôle de l'étape d'élicitation, la première étape du processus d'acquisition, est l'intégration des connaissances du domaine. La connaissance résultante exprimée en langage naturel est stockée dans des protocoles de connaissance. Dans l'approche PROTEGEII (Puerta, 1992) (Eriksson, 1995) aussi, le rôle des ontologies du domaine est de définir une conceptualisation partagée des connaissances pour développer des Systèmes à Base de Connaissances(SBC)

Pour la construction de l'un des modèles de connaissances tels que le modèle expert dans CommonKADS ou le modèle cognitif dans KOD (Vogel, 1989), les méthodes que nous avons présentées dans le paragraphe précédent utilisent une approche Bottom-Up. Autrement dit, elles se basent essentiellement sur une étape d'élicitation suivie d'une étape de conceptualisation et de construction du modèle concerné. Le projet ACACIA (Dieng, 1997 ; Dieng, 1996) utilisant la méthode CommonKADS pour développer un modèle expert du domaine en est un exemple. L'inconvénient dans cette approche réside dans le risque d'incomplétude des représentations offertes par le modèle final. En effet, comme la construction du modèle est basée sur l'élicitation, si l'un des experts ayant un point de vue

particulier ne participe pas dans la phase d'élicitation, ce point de vue risque de ne pas apparaître dans le modèle final.

Pour contourner ce problème, nous avons développé dans (Ben Ahmed et al., 2003) une approche Top-Down qui consiste à commencer par modéliser le phénomène étudié lui-même (accident de la route dans notre cas) pour identifier les points de vue nécessaires à l'étude. La deuxième étape consiste à instrumenter ces points de vue par élicitation des connaissances. Cette approche appliquées au domaine d'accidentologie a permis le développement du MRCA que nous présentons dans le paragraphe suivant.

3. Présentation du MRCA

Le MRCA développé dans (Ben Ahmed et al., 2003) utilise l'approche systémique de modélisation des systèmes complexes pour identifier les différents points de vue d'analyse de l'accident de la route.

Il identifie quatre points de vue :

- Point de vue ***ontologique*** ou ***structurel*** (ce qu'est le system) : il représente les composants du système (conducteur, infrastructure, trafic, conditions ambiantes et véhicule), les différentes interactions entre ces composants ainsi que leurs taxonomies ;
- Point de vue ***fonctionnel*** (ce que fait le system) : il représente le processus global du fonctionnement du système complexe Conducteur-Véhicule-Environnement (CVE) qui combine plusieurs procédures (régulation, saisie d'information, mémorisation, décision etc.) (Van Elslande et al., 1997) ;
- Point de vue ***transformationnel et comportemental*** (comment évolue le system) : il décrit l'évolution du système CVE. Ce modèle intègre le modèle séquentiel et causal de représentation de l'accident développé par l'INRETS (Brenac et al., 1997). Il représente l'accident à travers quatre étapes séquentielles (conduite normale, rupture[1], urgence[2] et choc) ;
- Point de vue ***motivationnel*** ou ***téléologique*** (quels sont l'objectif et la motivation du système) : il permet l'analyse de l'accident à travers l'objectif du conducteur lors de la conduite en général et lors d'un accident en particulier.

Ces quatre points de vue ont permis l'identification de plusieurs types de connaissances utilisées en accidentologie (Ben Ahmed et al., 2003). Pour formaliser ces connaissances, le MRCA utilise une approche « orientée attribut » (attribute-oriented). Concrètement, il s'agit de faire des classifications des 900 variables caractérisant l'accident selon les différents types de connaissances identifiées.

---

[1] Phase de rupture : elle est caractérisée par l'intervention d'un élément nouveau qui va faire basculer une situation de conduite en une situation d'urgence, créant une sollicitation ou une exigence qui va au delà des capacités du conducteur

[2] Phase d'urgence : la situation pendant laquelle le conducteur doit entreprendre une manœuvre pour éviter le choc

Un premier classement des variables a été effectué par deux équipes d'experts selon les quatre points de vue de la systémique décrits ci-dessus. Chaque équipe est composée d'un psychologue, expert infrastructure et expert véhicule.

Un deuxième classement expert des variables a été effectué selon le modèle structurel en affectant à chaque variable le composant concerné (conducteur, véhicule, environnement, piéton, interaction conducteur/véhicule, interaction conducteur/environnement etc.).

Un troisième classement expert a été effectué selon le modèle fonctionnel en affectant à chaque variable l'étape fonctionnelle concernée (perception, diagnostic, pronostic, décision et action).

Un quatrième classement expert a été effectué selon le modèle transformationnel en affectant à chaque variable l'étape séquentielle correspondante selon le modèle de l'INRETS (Brenac et al., 1997) (conduite normale, phase de rupture, phase d'urgence et phase de choc.).

Un cinquième classement expert a été effectué selon le niveau de granularité de chaque variable. Cette notion est très importante pour trouver des similarités entre les accidents. En effet, plus le niveau de détail des variables sélectionnées pour l'étude est fin, plus la chance de trouver des similarités est faible. Dans le cas extrême (niveau très fin), on peut trouver un accident par groupe. Inversement, si on reste à un niveau de détail très macroscopique, on peut finir par mettre tous les accidents dans un seul groupe.

Un sixième classement a été effectué selon le niveau d'importance de la variable par rapport à l'accidentologie primaire[3] d'une façon générale. La notion d'importance est bien sûr relative à l'objectif de chaque étude, mais nous avons défini cette importance par rapport à la problématique générale d'accidentologie primaire dans laquelle s'intègre notre étude.

Un septième et dernier classement des variables a été effectué selon leur niveau de fiabilité. En effet, il y a des variables issues de mesures, elles sont donc fiables. D'autres variables sont issues des déclarations du conducteur, elles le sont donc moins.

Les différents types de classements effectués sont présentés dans la figure 1.

Chacun de ces classements représente un type de connaissance experte en accidentologie. Il s'agit, pour certaines (fiabilité, pertinence, granularité etc.), de connaissances implicites, non formalisées et qui sont généralement utilisées par les accidentologistes lors des études effectuées dans le domaine.

Comme le domaine d'accidentologie est un domaine qui évolue (nouveaux systèmes de sécurité, nouvelles fonctions de confort, de communication etc.), certaines nouvelles variables sont ajoutées pour mettre à jour les BD EDA. Ces nouvelles variables seront classifiées selon les points de vue du MRCA.

---

[3] en sécurité routière on peut identifier trois approches : la sécurité primaire dont l'objectif est de prévenir une situation accidentelle (la prévention du risque au sens large, le freinage d'urgence, les aides informatives, le contrôle de trajectoire etc.), la sécurité secondaire dont l'objectif est de réduire la gravité des accidents (ceinture, air bag etc.) et la sécurité tertiaire dont l'objectif est d'améliorer la rapidité et la qualité des secours et des soins.

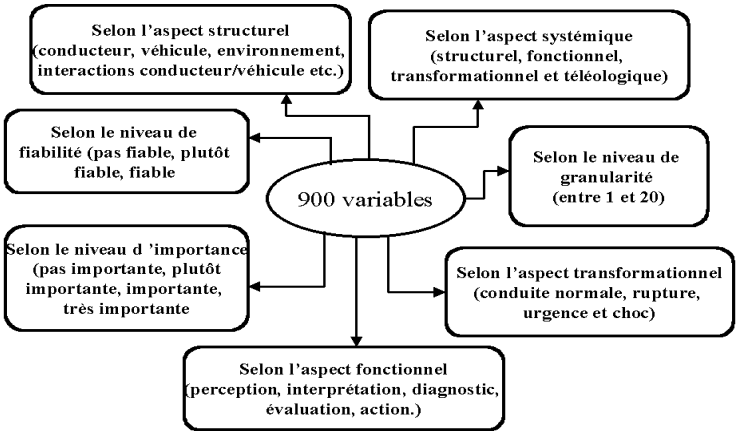


Fig.1. *Différentes classifications expertes des variables caractérisant les individus (accident)*

Dans la phase d'interprétation des résultats d'une classification permettant d'établir une typologie d'accidents, les classes de la typologie sont interprétées par des modalités de variables. A chacune de ces modalités est attaché un critère appelé Valeur-Test développé par (LeBart et al., 1977), le pourcentage des individus dans une classe décrits par une modalité donnée (MOD/CLASS) ainsi que le pourcentage d'une classe dans une modalité (CLASS/MOD).

En se basant sur ces trois paramètres, nous avons développé un outil permettant l'organisation automatique des modalités caractérisant chaque classe en :

- Modalité discriminante et caractérisante : C'est une modalité ayant une valeur-test supérieure à 2 (donc sur-représentée dans la classe par rapport à l'échantillon global) ou inférieure à –2 (donc sous-représentée dans la classe par rapport à l'échantillon global), mais elle est caractérisante car elle a un pourcentage MOD/CLASS supérieur à un seuil fixé par l'accidentologiste (le pourcentage généralement utilisé est 30%).
- Modalité discriminante mais non caractérisante : C'est une modalité ayant une valeur-test supérieure à 2 (sur-représentée) ou inférieure à –2 (sous-représentée), mais elle n'est pas caractérisante car son coefficient MOD/CLASS est très faible (< 30%).
- Variables non discriminantes, mais dont la modalité est caractérisante : Ce sont des variables ayant des valeurs-test entre –2 et 2 et un coefficient MOD/CLASS important. Le pourcentage de la modalité dans la classe est proche de celui dans l'échantillon global. Elles permettent de décrire le groupe, mais elles ne sont pas discriminantes pour ce groupe.

4. Application du MRCA dans un processus d'ECB

Pour évaluer l'apport du MRCA et donc de l'intégration des connaissances expertes en accidentologie, nous avons comparé des études qui ont été menées sans et avec l'utilisation du modèle. Les deux études ont pour vocation d'évaluer l'apport du MRCA dans la phase de sélection des variables et dans la phase d'interprétation des résultats issus d'une classification automatique des accidents de la route.

4.1. *Premier cas d'étude : sélection des variables*

Il s'agit d'une étude interne au sein du LAB (Riviere, 2003) dont l'objectif final est de développer un modèle général d'évaluation de l'efficacité a priori de prestations de sécurité. Comme étape intermédiaire, l'auteur a élaboré des scénarios-type d'accidents (STA) qui servent d'outil de synthèse dans la mesure où ils permettent le passage de la réflexion cas par cas à une vue générale et synthétique. L'échantillon d'accident à traiter contient 714 accidents décrits par les 900 variables. La construction d'un STA consiste en un regroupement de cas d'accidents présentant des similitudes de déroulement, puis en l'élaboration d'un déroulement prototypique (Fleury et Brenac, 2001). Ces STA peuvent être élaborés par des experts selon des critères de similitude empiriques ou en appliquant des méthodes de classification automatique. C'est la deuxième approche qui a été appliquée dans cette étude.

Dans (Riviere, 2003) l'objectif de la classification était d'identifier des prestations de sécurité primaire. L'auteur a choisi d'élaborer des situations accidentelles autour de *l'approche détection de situation d'accident*. Ainsi, la classification concerne les trois premières phases de l'accident : la situation avant le déplacement, la situation de conduite et la situation de rupture. La sélection des variables actives dans la classification s'est effectuée selon le critère général suivant : quels sont les éléments caractéristiques de la situation d'avant le déplacement, du déplacement, de la situation de conduite et de la situation de rupture qui peuvent expliquer le départ de la situation accidentelle et l'illustration de la situation d'urgence. Ces éléments sont relatifs à l'individu (notamment relatifs à ses défaillances fonctionnelles s'il y a lieu), au véhicule, à l'environnement général et à l'infrastructure. 25 variables ont été choisies par l'auteur et le nettoyage des données a été fait manuellement en utilisant l'expertise. Les variables n'ont pas fait l'objet d'une projection systématique selon les différents composants du modèle systémique. Cette absence de projection a pu générer des oublis dans l'identification de variables actives dans la classification, même si toutes les autres variables ont été traitées en illustratif.

En utilisant le MRCA, nous sommes partis des mêmes données nettoyées et nous avons complété les variables sélectionnées par d'autres variables d'une façon garantissant la couverture des différentes phases séquentielles, des différents composants du système CVE (conducteur-Véhicule-Environnement) ainsi que les différentes étapes fonctionnelles (perception, diagnostic, pronostic, décision et action).

Pour éviter le risque d'avoir un grand nombre de variables et donc d'influencer la qualité de la classification, nous avons également identifié des variables actives et des variables illustratives (ou supplémentaires). Les variables actives sont celles qui sont déjà utilisées pour effectuer la classification. Les variables illustratives, comme leur nom l'indique, sont utilisées pour aider à l'interprétation et à la caractérisation des groupes.

L'apport par rapport à l'étude de base réside dans le fait que :

- Le MRCA nous a permis de détecter les phases séquentielles, les composants structurels ainsi les étapes fonctionnelles non-représentées ou sous-représentées ;
- Nous n'avons pas refait appel aux experts pour la sélection des nouvelles variables étant donné que les variables ont déjà été classées selon les différents points de vue experts dans le MRCA ;
- L'interprétation des résultats est moins coûteuse en terme de temps car leur projection sur les différents points de vue experts est automatisée ;
- Lors de l'interprétation des résultats, le MRCA offre à l'accidentologiste une représentation multi-points de vue experts grâce à la possibilité de décrire automatiquement les groupes selon les différents axes de la systémique.

4.2. *Deuxième cas d'étude : interprétation des résultats*

Dans ce deuxième cas d'étude, nous avons fait élaboré de scénarios-types d'accidents en utilisant plusieurs algorithmes de classification automatique [Nasri, 2003]. Les objectifs de cette étude peuvent se résumer en deux volets : le premier est de comparer les différents algorithmes de classification. Vu que toutes nos données sont catégorielles, une ACM (Analyse en Composantes Multiples) a été faite. L'application des algorithmes de classification a été faite sur les cinq premiers facteurs principaux. Le deuxième objectif de l'étude est de mener une réflexion sur un outil d'aide à l'interprétation des résultats issus des classifications.

La phase de sélection des variables a été effectuée en utilisant le MRCA. Les avantages d'utilisation de ce modèle étaient d'accélérer cette phase, mais surtout de permettre à une personne qui n'a pas de connaissance en accidentologie d'effectuer cette tâche qui est très délicate. Les variables sélectionnées ont été revues et validées par un expert du LAB.

Les principaux écueils sont :

- La grande difficulté de comparer des résultats issus de l'application d'un même algorithme sur les mêmes données en changeant à chaque fois les variables actives et supplémentaires ;
- La grande difficulté de comparer des résultats issus de l'application de plusieurs algorithmes de classification automatiques sur les mêmes données et les mêmes variables. Les algorithmes qui ont été utilisés sont kmeans (Diday, 1971), la classification hiérarchique ascendante, l'algorithme PAM (Partitioning Around a Medoids) (Kaufman et Rousseeuw 1990) et la classification floue. Il était donc

difficile de comparer en terme de qualité les résultats issus des différents algorithmes.

Ces impossibilités étaient dues au fait qu'à la sortie des algorithmes de classification on se trouve avec des classes qui dans le meilleur des cas sont décrites par des modalités affectées de certains critères permettant de juger leur pertinence statistique. Exemples de ces critères : pourcentage des individus dans une classe et qui sont décrits par une modalité donnée (MOD/CLASS), le pourcentage des individus dans l'échantillon global et qui sont décrits par une modalité donnée CLASS/MOD etc. (voir figure 2). Mais, étant donné que toutes les variables utilisées sont catégorielles avec plusieurs modalités (4 en moyenne), très vite les critères statistiques deviennent insuffisants et leur exploitation devient lourde pour trouver des descriptions interprétables des classes.

Pour contourner ces difficultés, plusieurs projections des résultats ont été effectuées sur le MRCA. Concrètement, nous avons développé un programme sous MATLAB permettant la lecture des tableaux issus de la classification et leur projection selon un ou plusieurs points de vue du MRCA. Les sorties sont sous format HTML en utilisant des codes couleur pour faciliter les analyses (les modalités discriminantes et caractérisantes sont en vert, les modalités discriminantes non-caractérisantes sont en jaune et les modalités non-discriminantes et caractérisantes sont en rouge). Le gain en terme de temps est très important sachant qu'on a pu avoir de meilleurs résultats en quelques secondes au lieu de trois semaines pour un travail manuel. Le gain est surtout au niveau de la qualité car les différentes projections selon les différents points de vue étaient difficiles à réaliser sans le MRCA.

CLASSE 1 / 18

V.TEST	PROBA	---- POURCENTAGES ---- CLA/MOD	MOD/CLA	GLOBAL	MODALITES CARACTERISTIQUES	DES VARIABLES
				3.35	CLASSE 1 / 18	
10.89	0.000	51.16	91.67	6.00	typacc=Perturbation	typacc
4.78	0.000	11.72	62.50	17.85	Perte_contrôle_tr_8#	critini
4.16	0.000	15.15	41.67	9.21	Indisponib_infor_30#	mecdef
3.71	0.000	7.30	70.83	32.50	prob_contrôle_ve_21#	sitacc
3.15	0.001	13.73	29.17	7.11	evini=Prob_infra	evini
3.14	0.001	7.33	58.33	26.64	Hors_chaussee	LocChoc
2.97	0.002	14.63	25.00	5.72	gene_exterieur/d_14#	evini
2.83	0.002	6.70	58.33	29.15	Vehicule_seul	vehiprior2
2.79	0.003	6.30	62.50	33.19	surf=Mouille	surf
2.21	0.014	6.80	41.67	20.50	Obp=Obstacle	Obp
2.09	0.019	6.18	45.83	24.83	Section_courante_17#	manident
1.84	0.033	10.53	16.67	5.30	Problème_conduite	evini

Figure 2. *Sortie SPAD d'une classification des accidents*

5. Conclusion

En fouille de données complexes, les tâches les plus délicates sont la première et la dernière phase du processus d'ECB à savoir la préparation des données et l'interprétation des résultats. Les méthodes statistiques peuvent fournir un outil d'aide à la sélection des variables

et à l'interprétation des résultats, mais ne peuvent pas remplacer les experts. D'un autre coté, les approches expertes atteignent leurs limites (coût élevé, non-reproductibilité, divergence entre experts, etc.) dès qu'il s'agit de grandes bases de données, d'un grand nombre de variables et de besoin d'études qui changent régulièrement nécessitant l'application de l'ECB plusieurs fois.

Pour pallier ces limites, nous avons développé une approche qui combine les deux approches, experte et automatique. Grâce au Modèle de Représentation des Connaissances en Accidentologie (MRCA) (Ben Ahmed et al., 2003) nous avons pu identifier et formaliser certaines connaissances implicites mais fondamentales pour réussir un processus d'ECB sur des données complexes telles que celles de l'accident de la route. Ces connaissances portent sur les aspects structurel, séquentiel, fonctionnel de l'accident ainsi que la pertinence des données, leur niveau de granularité et leur fiabilité.

Le MRCA permet l'intégration de connaissances expertes en accidentologie dans la phase de sélection des variables ainsi que dans la phase d'interprétation des résultats. En effet, dans la phase de sélection de variables, le MRCA permet à l'accidentologiste de faire une sélection selon un ou plusieurs points de vue, selon l'objectif de l'étude. Dans la phase d'interprétation des résultats, le MRCA permet, grâce à la systémique, la structuration de la lecture des résultats issus de la classification des accidents selon les différents points de vue experts dans le domaine de l'accidentologie.

Deux cas d'étude ont été menés pour évaluer l'apport de l'utilisation du MRCA. Le premier a porté sur l'élaboration d'une sélection de variables pour faire une classification automatique des accidents et ce sans et avec utilisation du MRCA. Ce cas a montré qu'avec l'utilisation du MRCA la préparation des données est moins coûteuse en terme de temps d'expert et l'accidentologiste dispose de plusieurs points de vue structurant la lecture des résultats. Dans le deuxième cas d'étude, le MRCA a été utilisé dans la phase de préparation des données et on a essayé d'effectuer une interprétation des résultats sans et avec projection des ces résultats sur le MRCA. Ce deuxième cas a montré aussi que l'interprétation avec utilisation du MRCA est plus rapide et fournit à l'accidentologiste plusieurs points de vue d'interprétation.

6. Bibliographie

Alpay, L. , Amerge, C., Corby, O., Dieng, R., Giboin, A., Labidi, S., Lapalut, S., Ferrandez, F., Fleury, D., Girard, Y., Jourdan, J.-L., Lechner, D. Michel, J. E., Van Elslande, P., Acquisition et modélisation des connaissances dans le cadre d'une coopération entre plusieurs experts : Application à un système d'aide à l'analyse de l'accident de la route. *Rapport final du contrat DRAST* n. 93.0033, 1996.

Angele J., Fensel D., Studer R., Domain and task modeling in MIKE*, in A. sutcliffe et al. (eds), domain Knowledge for interactive System Desing,* Chapman & Hall, 1996.

Belanger N., Elaboration d'un système d'aide au diagnostique des accidents de la route par une approche cognitive, *thèse de doctorat,* Université de droit, d'économie et des sciences d'Aix-Marseille III, 2000.

Ben Ahmed W., Mekhilef M., Bigand M., Page Y., Development Of Knowledge Based System To Facilitate Design Of On-Board Car Safety Systems, *Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, Illinois USA*, *ASME 2003* September 2-6, 2003.

Boury-Brisset A.-C. and Tourigny N., Knowledge capitalisation through case bases and knowledge engineering for road safety analysis, *Knowledge-based systems,* vol. 13 ,2000, p. 297-305.

Brenac T., L'analyse séquentielle de l'accident de la route : comment la mettre en pratique dans les diagnostics de sécurité routière, Outil et méthode, Rapport de recherche n° 3, 1997, INRETS.

Calvanese D., De Giacomo G., Lenzerini M., Nardi D., and Rosati R.. Data integration in data warehousing. *Int. J. of Cooperative Information Systems*, 2001.

Diday E., La méthode des nuées dynamiques, *Revue statist. Appl.* Vol. 19, N° 2, pp. 19-34, 1971.

Dieng R., Comparison of Conceptual Graphs for Modelling Knowledge of Multiple Experts : Application to Traffic Accident Analysis, *Projet Acacia, INRIA* Rapport de recherche n°3161, 1997.

Dieng R., Giboin A., Amerge C., Corby O., Despres S., Alpay L., Labidi S., Lapalut S., Building of a Corporate Memory for Traffic Accident Analysis. *Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'96)*, Banff, Canada, p. 35.1-35.20, , 1996.

Eriksson H., Shahar Y., Tu S.W., Puerta A.R., Musen M.A., Task modeling with reusable problem-solving methods, *Artificial Intelligence*, vol. 79, 1995, p. 293-326.

Famili A., Shen W. M., Weber R., Simoudis E., Data Preprocessing and Intelligent Data Analysis, *Inetelligent Data Analysis*, vol. 1, p 3-23, 1997.

Fayyad U., Piatetsky-Shapiro G., and Smyth P., The KDD Process for Extracting Useful Knowledge from Volumes of Data, *Communications Of The ACM*, vol. 39, no. 11, 1996.

Ferrandez F., Fleury D., Girard Y., Alpay L., Amerge C., Corby O., Acquisition et modélisation des connaissances expertes en situation de coopération : application à un système d'aide à l'analyse des accidents de la route. Rapport final de Recherche, PREDIT, 1996.

Fleury D. and Brenac T., Accident prototypical scenarios, a tool for road safety research and diagnosis studies, *Accident Analysis & Prevention*, vol.33 , p. 267-276, 2001.

Han J., Kamber M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, ISBN 1-55860-489-8, 2000.

Jagadish H.V. et al., Special Issue on Data Reduction Techniques, *Bulletin of the Technical Committee on Data Engineering*, 20(4), December 1997.

Kaufman L. and Rousseeuw P., Finding Groups in Data : An Introduction to Cluster Analysis*, John Wiley and Sons*,1990.

Keim D. A. & Kriegel A. P., Visualization techniques for mining large databases: acomparision, *IEEE transaction on knowledge and data Engineering*, Vol. 8, No. 6, 1996.

Lebart L., Morineau A. et Piron M.*, Statistique exploratoire multidimentionnelle,* 2ème éd., Dunod, Paris, 1997.

Mercantini J. M., Chouraqui E. et Bélanger N., Etude d'un système d'aide au diagnostic des accidents de la circulation routière, *In Actes de la Conférence Ingénierie des Connaissances, IC'98*, 1998.

Nasri S., Application du data mining en accidentologie*, Rapport de projet de fin d'étude,* LAB, Renault/PSA, 2003.

Palmeri T. J., Blalock C., The role of background knowledge in speeded perceptual categorization, *Cognition* 77, 2000, p. B45-B57.

Puerta A.R., Egar J.W., Tu S.W., Musen M.A., A multiple-method Knowledge acquisition shell for the automatic generation of Knowledge acquisition tools, *Knowledge acquisition*, vol. 4, 1992, p. 171-196.

Pyle D., *Data Preparation For Data Mining* , Morgan Kaufmann Publishers, ISBN 1-55860-529-0, 1999.

Rahm E. and Do H. H., Data Cleaning: Problems and Current Approaches. *IEEE Data Engineering Bulletin*, vol. 23, n). 3, 2000.

Riviere C., Configurations accidentelles, identification de prestations sécuritaires et évaluation a priori de l'efficacité de ces prestations, *Rapport LAB*, Laboratoire d'accidentologie et biomécanique (LAB) PSA/Renault, 2003.

Schreiber A. Th., Wielinga B., de Hoog R., Akkermans H., Van de Velde W., CommonKADS : A comprehensive methodologiy for KBS development, *IEEE Expert*, 1994, 28-37.

Studer R., Benjamins V. R., and Fensel D., Knowledge Engineering: Principals and methods, *Data & Knowledge Engineering*, vol. 25, 1998, p. 161-197.

Van Elslande P., Alberton L., Scénarios-types de production de l'erreur humaine dans l'accident de la route, problématique et analyse qualitative, Rapport de recherche n°218, 1997, INRETS.

Vogel C., KOD: la mise en œuvre, Editions Masson, 1989.

Wielinga B.J., Expertise Model Definition Document, Esprit Project P5248 KADS-II, KADS-II/M2/UvA/026/5.0, 06/1994.

Classification automatique à partir de logs Web et de connaissances sur le site

Mireille Arnoux*,***, Yves Lechevallier**,
Doru Tanasa***, Brigitte Trousse***, Rosanna Verde**,****

*Département d' Informatique, Université de Bretagne Occidentale,
20 Avenue Le Gorgeu, 29285 Brest Cedex, France
Mireille.Arnoux@univ-brest.fr
**INRIA-Institut National de Recherche en Informatique et en Automatique,
Domaine de Voluceau-Rocquencourt B.P.105, 78153 Le Chesnay Cedex, France
Yves.Lechevallier@inria.fr
***INRIA-Institut National de Recherche en Informatique et en Automatique,
B.P.93, 2004 Route des Lucioles, 06902 Sophia Antipolis Cedex, France
Doru.Tanasa, Brigitte.Trousse@inria.fr
****Dip. Strategie Aziendali e Metodologie Quantitative
SUN - Seconda Università di Napoli,
Piazza Umberto I, 81043 Capua, Italie
rosanna.verde@unina2.it

Résumé. Dans ce travail nous proposons une analyse des données du Web Usage Mining. Outre les fichiers log Web l'analyse prend en compte des connaissances sur le site Web et des informations sur les utilisateurs du site. Un processus WUM contient trois étapes : pré-traitement, fouille de données et analyse des résultats. Tout d'abord nous donnons une courte description du processus WUM et des données Web et puis, nous détaillons l'étape de pré-traitement et l'entrepôt de données utilisé.
Pour l'étape de fouille de données, nous proposons deux méthodologies hybrides basées sur l'Analyse en Composantes Principales (ACP), respectivement l'Analyse des Correspondances Multiples (ACM) suivis d'une classification dynamique. Ces deux méthodologies ont été appliquées sur les log Web des sites Web d'INRIA et nous présentons les résultats obtenus en section 4.

1 Introduction et motivations

Le développement spéctaculaire du Web a entraîné au cours de ces dernières années une explosion des données liées à son activité. Pour analyser ce nouveau type de données, sont apparues de nouvelles méthodes d'analyse regroupées sous le terme Web Mining dont les trois axes de développement actuels sont les suivants :

- **Web Content Mining** : analyse textuelle avancée, intégrant les particularités du Web telles que les liens hypertextes et la structure sémantique des pages,
- **Web Structure Mining** : analyse de la structure de liens hypertextes de pages Web en vue d'une catégorisation des pages et sites Web et/ou une classification de sites Web,

– **Web Usage Mining** : analyse des comportements de navigation.

1.1 Web Usage Mining

La création de sites de grande taille nécessite de prendre en compte la navigabilité du site du point de vue de l'utilisateur. L'étude des parcours des utilisateurs (le WUM) à partir des logs issus de fichiers serveurs ou de traces propriétaires peut aider le responsable du site Web à repenser la structure et l'ergonomie du site, à repérer les problèmes rencontrés par les utilisateurs et, enfin à améliorer la navigabilité. Une analyse WUM peut permettre aussi au responsable du site Web d'améliorer les temps de réponse du serveur Web ou de faire de recommandations à l'utilisateur (insertion des liens dynamiques dans les pages).
Un processus WUM est communement divisé en trois étapes principales : pré-traitement des données, fouille de données et analyse des résultats (cf. fig. 1).

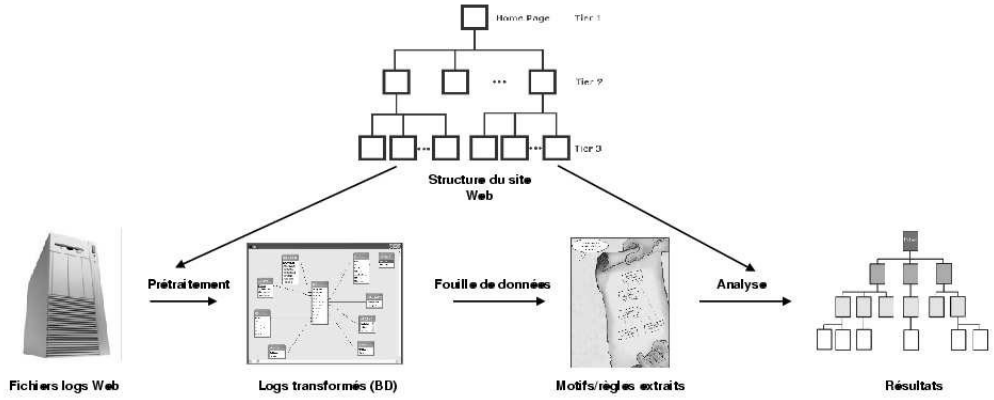


FIG. 1 – *Le schèma d'un processus WUM*

1.2 Caractéristiques des données d'usage Web

Les données utilisées en Web Usage Mining proviennent aujourd'hui principalement des fichiers **log HTTP**. Suivant le protocole client-seveur http, le poste client qui souhaite accéder à une ressource va émettre une requête adressée au serveur et contenant l'adresse d'allocation de la ressource :

```
GET http://www.inria.fr/accueil.html
```

A l'autre bout, le serveur Web interprète la requête HTTP, accède à la ressource demandée et la retourne au client. Comme dans la plupart des programmes informatiques, l'ensemble des opérations effectuées par le serveur sont enregistrées dans des fichiers log qui permettent de disposer d'une trace détaillée de l'activité du serveur. Nous utilisons le format de logs HTTP le plus répandu, l'ECLF (Extended Common Log Format) [Luotonen, 1995].

Pour analyser véritablement les usages du site Web nous avons également utilisé, outre les fichiers log, des connaissances sur le site Web (hiérarchie des rubriques, graphe des liens hypertexte, etc.) et des informations sur les utilisateurs du site (leurs profils) qui ont constitué des sources supplémentaires d'information. On se trouve donc, confronté au problème de l'organisation de ces données volumineux et diverses. Une structure d'accueil de type base de données est nécessaire et plus précisément elle doit s'orienter vers un entrepôt de données. Rappelons que l'entrepôt est développé à des fins décisionnelles ; il permet, par exemple, de faire des analyses de l'usage pour améliorer un site (structure), optimiser son fonctionnement (cache) et délivrer des recommandations à l'utilisateur. L'entrepôt est alimenté par le mécanisme d'extraction, de transformation et de chargement des données issues des différentes sources. Pour le cas des fichier log du serveur web, les étapes et les difficultés rencontrées sont détaillées dans les paragraphes suivants.

2 Nécessité de structurer les données d'usage Web

La première étape d'un processus WUM, c'est-à-dire le prétraitement des données, est fondamentale pour permettre une véritable analyse de l'usage. Les objectifs de cette étape sont d'identifier et de structurer les navigations des utilisateurs. Avant d'enregistrer les données dans un entrepôt de données elles sont nettoyées et transformées dans une première phase de pré-traitement des fichiers log Web.

2.1 Pré-traitement des fichiers log Web

Nettoyage des données : Le nettoyage des données pour les fichiers log Web consiste à supprimer les requêtes inutiles de fichiers « log ». Selon l'objectif poursuivi, ces requêtes peuvent concerner les images et les fichiers multimédia. L'identification de robots Web et la suppression des requêtes provenant de ces robots sont les autres tâches de cette étape. Pour plus de détails concernant l'étape de pré-traitement de fichier log HTTP, le lecteur intéressé pourra se repporter à [Tanasa et Trousse, 2003].

Transformation des données : L'unité d'analyse étant la séquence de pages visualisées et non la simple requête, il est préalablement nécessaire de regrouper les requêtes contenues dans les fichiers log [1] par utilisateur pour reconstituer ensuite ses sessions et ses navigations.

Identification des utilisateurs
Bien que cette tâche paraisse à première vue assez aisée, l'analyste est confronté à un certain nombre de problèmes techniques. Pour regrouper les requêtes, il est nécessaire de savoir quels utilisateurs les ont émises. Si l'utilisateur a accepté de s'enregistrer et s'identifie avec un login, alors le repérage est immédiat, mais cela ne concerne qu'une très faible minorité des visites. Une autre méthode répandue mais nécessitant l'acceptation de l'utilisateur, consiste à écrire dans la mémoire du navigateur, c'est-à-dire sur

1. Dans le cas de plusieurs serveurs Web il faut commencer par fusioner ces fichiers.

le poste client, un fichier d'identification nommé *"cookie"* qui sera réutilisé dans chacune des requêtes et permettra au serveur d'en identifier la provenance. En fait on ne dispose que de l'adresse IP qui est identique pour tous les utilisateurs partageant un même *router* ou pour ceux accédant à l'Internet via le même serveur proxy. Dans ce cas, il est difficile de parler d'identification d'utilisateur.

Identification des sessions utilisateur
Toutes les requêtes faites par un utilisateur pendant la période analysée constituent sa session Web. Pour constituer une session utilisateur nous allons grouper les requêtes par le triplet `(Login, Host, User Agent)`. Si le login n'est pas disponible on prend en compte seulement les deux dernières champs. Ces requêtes ordonnées de manière chronologique constitue la session de l'utilisateur.

Identification des navigations
Comme dernière étape de transformation de données, nous avons divisé les sessions en navigations. Une nouvelle navigation commence dès que une interval de plus de 30 minutes est observé entre deux requêtes consécutives.

2.2 Entrepôt des données de l'usage du web

Après les étapes décrites ci-dessus, les données de l'usage du web peuvent peupler la base de données. Il s'agit d'un entrepôt de données durable où tous les points de vue sont conservés par opposition à des visions plus éphémères ou plus spécialisées de la littérature que nous évoquerons ensuite. Le modèle que nous proposons est un schéma en étoile (cf. fig. 2).

Les faits : Les faits sont issus des fichiers « log » filtrés, traités et enrichis de données calculées comme par exemple *DureeTotale* ou *NbRequests*.

L'étude étant centrée sur l'utilisateur, les indicateurs sont essentiellement la durée de consultation et le volume consulté. Nous travaillons actuellement sur un suivi de l'utilisateur beaucoup plus précis que les classiques « logs » du serveur ; d'autres indicateurs sont alors disponibles relatifs à sa satisfaction à la lecture d'une page, qu'il les exprime explicitement ou qu'ils soient déduits de son comportement.

Les dimensions : Les dimensions retenues peuvent être ventilées en :

- Dimension liée à l'url et à la page consultée. C'est le point de vue **Contenu**. Diverses catégorisations peuvent être faites : simplement par l'extension du fichier mais aussi en introduisant des renseignements sur le site comme le placement du fichier dans une rubrique, une sur- rubrique.
- Dimension liée à la date. C'est le point de vue **Régularité d'accès**. On peut utiliser la hiérarchie habituelle seconde, minute, heure, jour, mois année ou préférer délimiter certaines périodes.
- Dimension liée à la session et à l'utilisateur. C'est le point de vue **Utilisateur**. Une session particulière appartient à un utilisateur, lui même appartenant à un service, un projet, une unité de recherche, un domaine. On peut aussi classifier les sessions sur d'autres critères comme nous le verrons plus loin. On peut alors enri-

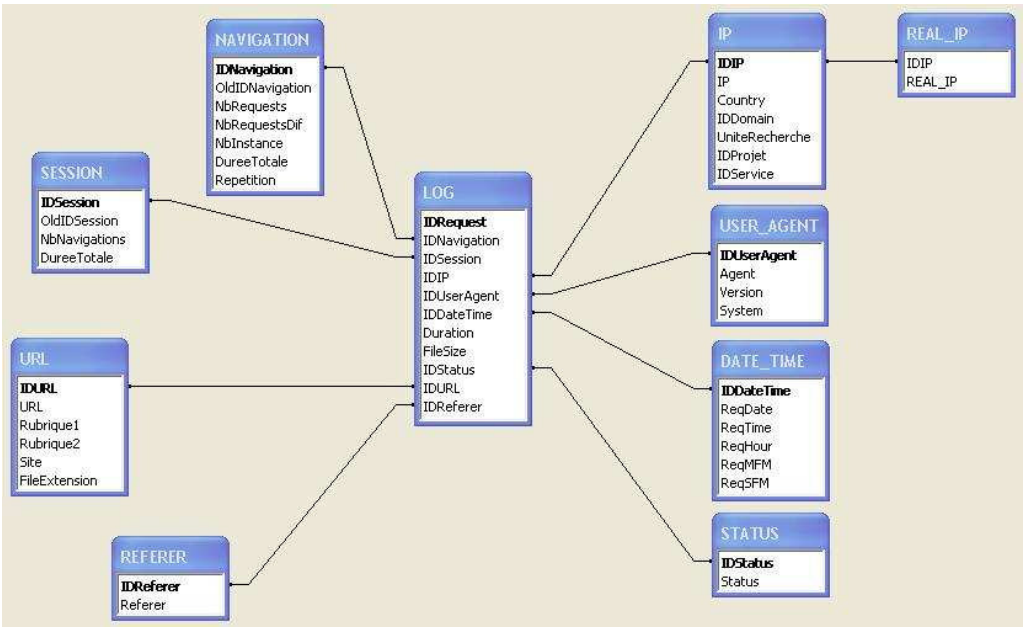


FIG. 2 – *Schéma en étoile de la base de données actuelle*

chir la description de la session en la plaçant dans une classe (puis éventuellement dans une sur-classe).

- Dimension liée au *referer* et à la navigation. C'est le point de vue **Navigation**. Le referer permet (avec quelques incertitudes) de retracer la navigation de l'internaute qui peut être linéaire ou comporter des répétitions.
- Dimension liée au statut de la transaction. C'est le point de vue **Efficacité d'accès**. Les « logs » du serveur donne le statut résultant du traitement de la requête (comme réussite, erreur, redirection, accès interdit). Ici encore, on peut définir un autre type de statut résultant cette fois du contexte dans lequel la requête a été émise (page recommandée dans le cas où l'internaute travaille avec un système de recommandation et où l'on dispose d'un suivi plus précis que les simples traces du serveur).

Les hiérarchies de dimensions : Dans cette application, l'existence de multi-hiérarchies dans les dimensions est essentielle. L'implantation actuelle en Access est surtout motivée par un souci d'efficacité avec l'emploi d'une clé étrangère symbolique vers chaque élément de dimension. Dans le schéma Access, les hiérarchies de dimensions apparaissent simplement sur les regroupements visuels de tables (`session` peut admettre comme supérieur hiérarchique `User Agent` ou `IP`) ou directement dans les attributs (`URL` peut admettre comme supérieur hiérarchique rubrique2 lui-même dominé par rubrique1). La mise en évidence de tels niveaux de généralisation est essentielle pour appliquer les opérations "roll-up" ou "drill-down" qui synthétisent ou détaillent

les résultats issus d'un entrepôt de données.
Nous envisageons de porter le système sous Oracle9i qui implante le concept de multihiérarchies de dimensions et débouche sur des outils d'analyse performants OLAP et ROLAP. De plus, Oracle9i Warehouse Builder permet de disposer des méta-données concernant l'entrepôt sous le format standard Common Warehouse Metamodel (CWM) et d'automatiser les processus d'extraction, transformation, chargement des données (ETL).

2.3 Position par rapport aux travaux existants

Les données de l'usage du web sont souvent exploitées en vue de produire des statistiques ou même dans une optique de fouille de données pour prévoir les pages qui vont être demandées. Elles sont filtrées et stockées selon les nécéssités. Pour cela, on peut distinguer deux modes essentiels :

a) Il y a des structures spécialisées pour mémoriser et organiser les données. Les arbres ou les graphes sont utilisés lorsque l'on recherche surtout des modèles de navigation.

C'est le cas notamment du logiciel WUM (Web Utilisation Miner) [Spiliopoulou *et al.*, 1999] qui utilise des arbres d'agrégation pondérés pour montrer le trafic de navigation sur les chemins tracés par l'organisation logique d'un site. WUM propose un langage de requête MINT qui opère sur ces arbres avec une syntaxe à la SQL. Des structures vectorielles sont également fréquentes.

Dans WEBMiner [Mobasher *et al.*, 2002], une transaction est représentée comme un vecteur dans l'espace des pages visibles. Par ailleurs ce logiciel utilise d'autres informations sur les utilisateurs et les documents qu'il intègre aux traces ; il a un étage de données intégrées interrogeables par un langage de requêtes qu'il reformatte en fonction du travail d'analyse à effectuer.

b) Les données sont uniquement placées dans des fichiers et des tables relationnelles. La plupart des logiciels commerciaux emploient de telles stuctures sous une forme peu élaborée et elles sont rarement décrites. Cependant une base de données relationnelle permet de fédérer tous les aspects utiles à l'analyse de l'usage à la différence des implantations précédentes qui avaient des objectifs plus précis (une représentation pour analyser les navigations statistiquement, une autre pour construire des profils utilisateurs). C'est dans cette perspective que nous nous situons et que nous trouvons les travaux suivants.

[Bonchi *et al.*, 2001] décrit un magasin de données implanté comme une base de données relationnelles avec Microsoft SQL Server. Son but est de développer des stratégies optimales de cache mémoire grâce à l'analyse de ces données. Les faits sont issus des fichiers « log » et comportent des indicateurs calculés comme **LastAccess** et **NextAccess** qui indiquent, en liaison avec une ligne de « log » le nombre de lignes à parcourir en amont ou en aval pour retrouver une demande sur la même url. On trouve également l'indicateur **PageDelay** qui est la distance en micro secondes de la requête à la requête précédente du même utilisateur. Ceci permet de distinguer les pages traversées rapidement comme des liens de celles observées plus longuement pour leur contenu. Une notion de session en est dégagée comme une suite de requêtes « liens » aboutissant

à une requête « contenu ». Cela correspond à un objectif de visite, d'une granularité bien plus fine que notre notion de session. Les dimensions ne sont pas développées dans des tables spécifiques, seule la dimension url est codée pour pallier la lenteur des SGBD dans le traitement des chaînes de caractères. Des informations issues du site sont introduites comme la profondeur du fichier demandé dans l'arborescence du site. Enfin deux points intéressants sont à signaler : le maintien d'un ordre temporel total pour les transactions et l'existence d'un champ Class (ici pour la classe WebLog) qui peut être valorisé a posteriori pour introduire une classification des transactions. Nous utilisons également les séries temporelles de transactions pour la classification. D'autre part, les résultats issus de la classification sont disponibles dans un format XML mais aussi pour valoriser des attributs réservés dans les dimensions.

WebLogMiner [Zaiane *et al.*, 1998] est un logiciel d'analyse de l'usage qui utilise un entrepôt de données et construit un cube multidimensionnel pour appliquer les techniques OLAP. Parmi les dimensions retenues il y a celles issues des traces (l'url, le type et le taille de la ressource, la date, le temps passé, l'utilisateur, l'agent, le domaine et le statut) et deux dimensions supplémentaires **FieldSite** et **Event**. **FieldSite** permet d'introduire la connaissance de la structure hiérarchique du site et **Event** permet de préciser une action type de l'utilisateur (utiliser un V-groupe, ajouter un message, lire un message) Ceci rejoint notre souci de décrire le site et de suivre l'utilisateur plus précisément qu'avec les seuls"logs".
Dans WebLogMiner, le modèle dimensionnel suppose qu'il y a pour seul fait l'existence de "clicks" repérés par un nuple de dimensions (table de faits sans indicateur). Ceci diffère de notre représentation avec des indicateurs comme *DuréeTotale* ou *NbRequests*. WebLogMiner permet de réaliser un tableau de bord avec diverses fréquences et aussi de dégager des modèles d'utilisateurs et de faire de l'analyse de tendance. Ceci est proche de nos objectifs de classification exposés dans les paragraphes suivants.

3 Méthodes de classification

Après les phases de nettoyage et de transformation de données qui permettent de remplir la base de données et de construire un tableau de description des sessions nous abordons la phase d'analyse. L'objectif de cette phase est de découvrir différents comportements d'utilisateurs ou des catégories de comportement de navigation par diverses approches [Säuberlich et Huber, 2002] : Analyse des séquences fréquences, segmentation, modèle prédictifs, réseau neuronal et classification automatique.

3.1 Travaux existants

Il existe de nombreuses méthodes de classification utilisées dans la fouille de données. Cependant, peu de méthodes ont été appliquées aux données du Web : BIRCH dans [Fu *et al.*, 2000], CLIQUE dans [Perkowitz et Etzioni, 1998], EM dans
[Cadez *et al.*, 2000], car il est difficile, voire impossible, d'adapter certaines méthodes aux particularités des données Web compte tenu de la taille de ces tableaux tant pour les sessions que pour les pages différentes.
Dans [Mobasher *et al.*, 2002], les auteurs considèrent deux méthodes de classification,

mais ne prennent pas en compte l'ordre des pages dans les sessions. Les sessions (appelées "transactions" dans l'article) sont représentées par des vecteurs binaires ou pondérés par des vues de pages (pages, dans la suite). Une transaction t est de la forme $t =< w(p_1,t),w(p_2,t),...w(p_n,t) >$ où $w(p_i,t)$ est le poids de la page p_i dans la transaction t. Le poids d'une page dans une transaction peut dépendre de la présence ou l'absence de cette page dans la transaction (1, respectivement 0). Le poids peut être égal à la valeur d'une fonction sur le temps total de chargement et d'affichage de la page (pouvant exprimer l'intérêt de l'utilisateur dans cette page) ou d'une autre fonction sur le type de page.
Dans [Fu *et al.*, 2000] les sessions des utilisateurs sont généralisées en utilisant une induction basée sur les attributs. Cette induction a comme objectif de réduire les dimensions des données. Lors d'une première étape les pages sont organisées dans une hiérarchie. Par exemple la page **www-sop.inria.fr/axis/teaching/stid-projet2.html** est mise dans la hiérarchie suivante : **www-sop.inria.fr** → **axis** → **teaching** . Dans l'étape de généralisation la page est ainsi remplacée par une page plus générale : **www-sop.inria.fr/axis/teaching/**. Cette hiérarchie est construite seulement sur la syntaxe des URLs et pas sur la sémantique des pages. Les données généralisées sont ensuite classées en utilisant un algorithme efficace de classification hiérarchique - BIRCH introduit par [Zhang *et al.*, 1996]. Dans les expérimentations décrites, BIRCH a été capable de proposer une classification des sessions associées à 21 107 utilisateurs. Cependant, d'après les auteurs les performances de l'algorithme BIRCH se dégradent visiblement en augmentant la dimension des données ce qui limite la généralisation des pages à quelques niveaux.
Une classification non-supervisé basée sur un réseau de neurones est utilisée dans [Benedeck et Trousse, 2003] pour grouper les sessions similaires (issues d'un site annuaire thématique) en classes.

3.2 Notre approche

L'objectif est de développer une stratégie qui analyse les relations existantes entre la structure d'un site Web et le fichier log Web du même site. Pour arriver à cela nous appliquons deux méthodes hybrides de classification sur deux types de données Web : continues (numériques) et qualitatives.

3.2.1 Méthode de classification pour les données Web numériques

La première méthode d'analyse est la classification hyride du fichier log. Les objets analysés sont les navigations Web et les variables sont continues. La classification hybride utilise :

1. **l'analyse par composantes principales (ACP)** pour la visualisation des corrélations existantes entre les variables définies sur le fichier log Web
2. **une méthode de classification dynamique** sur les principaux facteurs donnés par l'ACP.

L'objectif de la classification dynamique est de trouver un groupe de navigations homogènes. La méthode de classification dynamique peut être résumée de la façon suivante :

Etape 1: Initialisation
Soit k objets, choisis d'une manière aléatoire dans un ensemble de "navigations", représentant les k classes. Appelons ces objets : A_1,...A_k.
Etape 2: Affectation
Toute navigation X_i est assignée à la classe j ssi $d(X_i,A_j)$ est le minimum de $\{d(X_i,A_j)|j = \overline{1..k}\}$.
Etape 3: Représentation
Une nouvelle représentation A_k est calculée comme étant la moyenne des éléments de la classe k.
Etape 4: Stabilisation
Si toutes les navigations ne changent pas la classe alors Stop, sinon saut à l'étape 2.

Un pré-traitement sur le fichier log est nécessaire pour pouvoir calculer les six variables actives suivantes pour chaque navigation. Ces variables sont utilisées pour l'ACP.

PRequest_SEL	pourcentage des requêtes réussies
NBrequest	nombre total des requêtes
DureeTotale	durée totale de la navigation
Repetition	taux de répetition [Jaczynski, 1998]
MDurée_OK	durée moyenne d'une requête pour une navigation
MSize_OK	dimension moyenne des fichiers demandés dans les requêtes de la navigation

La classification des navigations est suivie par la description des classes et leur placement sur un plan factoriel.

3.2.2 Méthode de classification pour les données Web qualitatives

La deuxième analyse est à nouveau une méthode de classification hybride appliquée sur les données qualitatives. Il s'agit d'une analyse par correspondances multiples (ACM) sur un ensemble de navigations concernant seulement les deux premièrs sites Web : `www` et `www-sop`. Chaque variable correspond à une rubrique sémantique du site Web considéré, c'est une variable qualitative. Nous avons donné la définition des rubriques sémantiques pour le premier niveau des deux sites web (`www` et `www-sop`). A chaque rubrique syntaxique de premier niveau correspond une rubrique sémantique de premier niveau. Prenons par exemple l'URL `http://www-sop.inria.fr/axis/software/index.html`, ‘‘`axis`’’ est sa rubrique syntaxique de premier niveau et “equipe de recherche” est la rubrique sémantique correspondante. Nous avons fixé 58 rubriques sémantiques différentes pour les deux sites web.
Après l'ACM nous utilisons une méthode de classification dynamique avec une métrique *khi*2 et nous obtenons quelques groupes de navigations et leur interprétation.
Pour cette analyse nous avons besoin d'organiser les pages Web en rubriques sémantiques.
Pour la classification, les algorithmes utilisées sont de type Nuées Dynamiques [Diday, 1971]. Ils recherchent simultanément une partition P de E en k classes et un vecteur L de k prototypes minimisant : $\Delta(P^\bullet,L^\bullet) = Min\{\Delta(P,L) \mid P \in P_k, L \in D_k\}$, avec P_k l'ensemble des partitions de E en k classes non vides. Ce critère Δ exprime l'adéquation entre la partition P et le vecteur des k prototypes. Il est souvent défini comme la somme des distances entre tous les objets s de E et le prototype g_i de la classe C_i la plus proche.

Pour cette méthode nous avons défini deux nouveaux paramètres pour les navigations et pour chaque serveur Web :

SyntacticTopic-WebServer	vecteur contenant les rubriques syntaxiques du serveur Web
SemanticTopic-WebServer	vecteur contenant les rubriques sémantiques du serveur Web

4 Application aux données issues de serveurs Web de l'INRIA

Pour valider notre approche, nous avons choisi d'exploiter les données issues de trois des serveurs web de l'INRIA : `www.inria.fr`, `www-sop.inria.fr` et `www-futurs.inria.fr`. Il y a de nombreux liens entre ces serveurs web et l'un des buts de notre analyse a été de trouver des groupes de sujets consultés ensemble.

4.1 Les données brutes

Les fichiers traces des accès http aux serveurs web ont été recueillis sur une période de deux semaines entre le 1 et le 15 Janvier 2003. Certains caractéristiques de ces données sont présentées dans le tableau 1.

Serveurs Web	{`www, www-sop, www-futurs`}`.inria.fr`
Période	1 - 15 Janvier 2003
Nombre de requêtes	6 040 312
Requêtes après pré-traitement	673 389
Nombre de sessions	115 825
Nombre de navigations	174 015

TAB. 1 – *Résumé des données expérimentales*

4.2 Pré-traitements réalisés

Pour la première analyse, nous avons décidé de ne garder que les navigations "longues", c'est à dire celles dont la durée est supérieure à 60 secondes. La seconde contrainte imposée a été que le nombre de pages visitées dépasse le seuil de 10 pages. Enfin, la troisième condition à remplir était que la vitesse d'exploration (le rapport entre la durée de navigation et le nombre de pages visitées) soit de plus de 4 sec./page. Après avoir filtré la base de données à l'aide de ces contraintes, nous avons retenu 9700 navigations. Le nombre de requêtes HTTP valides (ayant le statut entre 200 et 399) dans ces navigations a été de 282 705. Sur cet ensemble de navigations, nous avons appliqué la première technique de classification.

Pour la seconde analyse, nous n'avons retenu, dans le premier ensemble de navigations, que celles dont les requêtes visaient les sites web `www` et `www-sop`. Le nombre de navigations sur le site web `www-futurs` était très restreint (28), ce qui nous a amené à les ignorer. Finalement, le nombre de navigations vérifiant les conditions pour cette analyse a été de 3969.

4.3 Entrepôt de données

En accord avec le modèle relationnel de la figure 2, nous avons dans l'entrepôt de données trois types d'information:

- information sur l'utilisation du site web (requêtes, navigations)
  Le premier type d'information a été déjà présenté dans le paragraphe précédent.
- information sur les utilisateurs
  Durant la période analysée, au total, 115 825 utilisateurs avaient fait des requêtes vers les trois serveurs web. La plupart de ces utilisateurs venaient de France (30,7utilisateurs appartenaient à des unités de recherche de l'INRIA. Dans le cas de l'unité de recherche de Sophia Antipolis (429 utilisateurs) nous connaissions aussi l'équipe de recherche ou le service auquel appartenait l'utilisateur.
- information sur le site web (URLs, sujets)
  Il y a eu 86640 URLs demandées dans les trois sites web. D'après leur syntaxe, les URLs ont été groupées en 58 sujets sémantiques définis par les analystes/spécialistes du contenu du site.

4.4 Les résultats de la première analyse (variables continues)

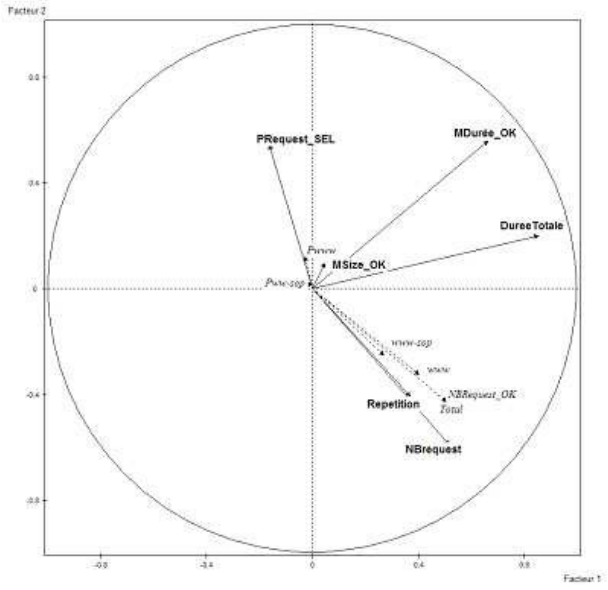


FIG. 3 – *Le cercle de correlation*

Les résultats de l'ACP sont présentés dans le cercle de corrélation (fig. 3) et dans le plan des classes (Fig. 4). Nous avons obtenu 7 classes, mais seulement les classes 1, 2, 4 et 5 sont significatives. La classe 1 contient les navigations qui ont, généralement, une courte durée. La durée moyenne et le nombre de pages visitées sont inférieures aux

valeurs moyennes pour toutes les navigations. Une classe qui est intéressante, même si elle est très petite (seulement 17 navigations), est la classe 6. Elle contient des navigations qui sont très longues (avec une moyenne de 887,76 pages / navigation), venant surtout des domaines avec l'extension `.net`, en majorité faites pendant la nuit et le 1^{er} Janvier. Nous supposons que ces navigations sont effectuées par les robots Web [2], car il est bien connu que ces outils d'indexation sont lancés en début de mois (pour parcourir les sites Web et actualiser les index).

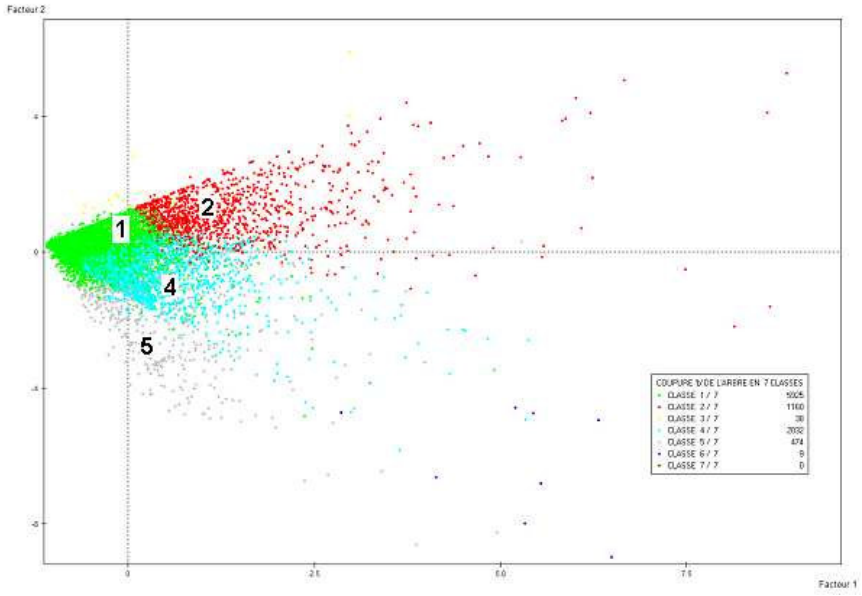


FIG. 4 – *Le résultat de l'analyse ACP*

4.5 Les résultats de la deuxième analyse (variables qualitatives)

Suite à l'exécution de l'ACM nous avons obtenus 11 classes. Par exemple, la classe 10 (tab. 2) représente 19.01% des navigations. Cette classe contient les navigations des utilisateurs qui se sont intéressés aux pages Web avec les rapports INRIA (rapports d'activité, rapports techniques et de recherche). Dans ce cas, l'objectif de ces utilisateurs est clair, l'intérêt du visiteur est de lire les rapports des chercheurs de l'INRIA.

2. Un étude ultérieur vérifiera cette hypothèse.

Etiquette rubrique	V.Test	Classe/Freq	Freq/Classe	Global
www ra	240.13	85.27	66.49	14.82
www rrrt	12.85	28.82	3.00	1.98
www rapports	10.68	44.13	0.56	0.24
sop rapports	7.17	35.31	0.45	0.24

TAB. 2 – *Description de la classe 10*

5 Conclusions et perspectives

Dans ce papier nous avons, tout d'abord, introduit les données d'usage Web qui sont volumineux et diverses (fichiers log et connaissances sur le site et les utilisateurs) et insister sur l'organisation nécessaire de ces données lors de l'étape de pré-traitement. Puis, nous avons proposé deux méthodologies pour classifier les données d'usage Web. Notre approche utilise deux méthodes hybrides de classification (ACP ou ACM suivi d'une classification dynamique) pour analyser les données pré-traitées issues des fichiers log Web. Nous avons testé nos méthodologies sur des fichiers log de plusieurs serveurs Web de l'INRIA et les résultats obtenus sont très intéressants.
A court terme, notre objectif est de réaliser une corrélation entre ces deux analyses. Ceci peut être fait, éventuellement, par la prise en compte de nouveaux paramètres pour caractériser les données log Web. Nous envisageons, aussi, d'introduire une hiérarchie sur les rubriques sémantiques.

Références

[Benedeck et Trousse, 2003] Attila Benedeck et Brigitte Trousse. Visualization adaptation of self-organizing maps for case indexing. In *27th Annual Conference of the Gesellschaft fur Klassifikation, Cottbus Germany, 12-14 March 2003*, 2003.

[Bonchi *et al.*, 2001] Francesco Bonchi, Fosca Giannotti, Cristian Gozzi, Giuseppe Manco, Mirco Nanni, Dino Pedreschi, C. Renso, et Salvatore Ruggieri. Web log data warehousing and mining for intelligent web caching. *Data Knowledge Engineering*, 39(2):165–189, 2001.

[Cadez *et al.*, 2000] Igor V. Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, et Steven White. Visualization of navigation patterns on a web site using model-based clustering. In *In Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 280–284, Boston, Massachusetts, 2000.

[Diday, 1971] E. Diday. La methode des nuees dynamiques. *Revue de Statistique Appliquée*, XIX(2):19–34, 1971.

[Fu *et al.*, 2000] Y. Fu, K. Sandhu, et M. Shih. A generalization-based approach to clustering of web usage sessions. In *Proceedings of the 1999 KDD Workshop on Web Mining, San Diego, CA. Springer-Verlag*, volume 1836 of *LNAI*, pages 21–38. Springer, 2000.

[Jaczynski, 1998] M. Jaczynski. Modèle et plate-forme à objets pour l'indexation des cas par situation comportementales: application à l'assistance à la navigation sur le web, décembre 1998. Thèse de doctorat, Université de Nice Sophia-Antipolis.

[Luotonen, 1995] A. Luotonen. The common log file format. http://www.w3.org/Daemon/User/Config/Logging.html, 1995.

[Mobasher *et al.*, 2002] Bamshad Mobasher, Honghua Dai, Tao Luo, et Miki Nakagawa. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery*, 6(1):61–82, January 2002.

[Perkowitz et Etzioni, 1998] Mike Perkowitz et Oren Etzioni. Adaptive web sites: Automatically synthesizing web pages. In *AAAI/IAAI*, pages 727–732, 1998.

[Spiliopoulou *et al.*, 1999] Myra Spiliopoulou, Lukas C. Faulstich, et Karsten Winkler. A data miner analyzing the navigational behaviour of web users. In *Proc. of the Workshop on Machine Learning in User Modelling of the ACAI'99 Int. Conf.*, Creta, Greece, July 1999.

[Säuberlich et Huber, 2002] F. Säuberlich et K.-P. Huber. A framework for web usage mining on anonymous logfile data. In O. Opitz et M. Schwaiger, editors, *Exploratory Data Analysis in Empirical Research, Proceedings of the 25th Annual Conference of the Gesellschaft für Klassifikation e.V., University of Munich, March 14-16, 2001*, pages 229–239. Springer-Verlag, 2002.

[Tanasa et Trousse, 2003] Doru Tanasa et Brigitte Trousse. Le prétraitement des fichiers logs web dans le "Web Usage Mining" multi-sites. In *Journées Francophones de la Toile (JFT'2003)*, Tours, June - July 2003.

[Zaiane *et al.*, 1998] Osmar R. Zaiane, Man Xin, et Jiawei Han. Discovering web access patterns and trends by applying OLAP and data mining technology on web logs. In *Advances in Digital Libraries*, pages 19–29, 1998.

[Zhang *et al.*, 1996] Tian Zhang, Raghu Ramakrishnan, et Miron Livny. Birch: An efficient data clustering method for very large databases. In H. V. Jagadish et Inderpal Singh Mumick, editors, *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6, 1996*, pages 103–114. ACM Press, 1996.

Summary

In this paper we propose an analysis based on Web Usage Mining data. Apart from the log files, the analysis uses knowledge about the Web site and information about its users. The WUM process contains three steps: pre-processing, data mining and result analysis. First, we give a brief description of the WUM process and Web data, followed in section 2 by the presentation of the pre-processing step and the data warehouse that we employed.

Two hybrid clustering methods based on Principal Components Analysis (PCA), Multiple Classification Analysis (MCA) and Dynamic Clustering, are used for analysing the Web logs taken from INRIA's Web servers. The results obtained after applying these methods and the corresponding interpretations are presented in section 4 of the article.

Recherche par le contenu dans une base d'images fixes : l'intérêt des règles d'association

Anicet Kouomou Choupo*, Annie Morin*, Laure Berti-Équille*

*IRISA Campus de Beaulieu 35042 Rennes-France
{akouomou, Annie.Morin, Laure.Berti-Equille}@irisa.fr
http://www.irisa.fr/texmex/

Résumé. Une base d'images fixes peut être décrite de plusieurs façons, notamment par des descripteurs visuels globaux de couleur, de texture, ou de forme (niveau pixel). Les requêtes les plus fréquentes impliquent et combinent les résultats de plusieurs types de descripteurs : par exemple, "retrouver toutes les images ayant une couleur et une texture semblables à celles d'une image requête donnée". Pour retrouver plus efficacement et plus rapidement une image dans une base d'images fixes, nous exploitons des combinaisons appropriées de descripteurs. Dans ce papier, nous étudions l'intérêt des règles d'association entre clusters de descripteurs lors de la recherche dans une base d'images fixes. Après avoir décrit le système de recherche qui nous sert de cadre d'étude, nous nous focalisons essentiellement sur le temps de réponse à des requêtes puis nous confrontons notre système à l'approche classique de recherche séquentielle avec fusion des résultats.

Mots-clé : règles d'association, recherche par le contenu, image fixe, clustering.

1 Introduction

La problématique de la recherche d'informations multimédias par le contenu est rendue plus complexe lorsque la base devient conséquente. Le temps de réponse aux requêtes n'est plus négligeable et il convient de déployer des techniques d'indexation adaptées aux données multimédias en grandes quantités. Dans ce contexte, nous avons envisagé d'exploiter les résultats de deux techniques de fouille de données dans la mesure où elles nous permettent d'utiliser des informations complémentaires sous forme de règles sur les données traitées. L'extraction de connaissances à partir de données multimédias apporte cependant une complexité nouvelle liée à la représentation des données analysées. La représentation des documents multimédias est soit indépendante du contenu (format, origine des données, date, auteur, conditions de création du document, ...), soit directement liée au contenu (mots-clés, fichiers inversés, descripteurs d'images, ...). Nous nous sommes intéressés à cette dernière forme de représentation pour une base d'images fixes.

Une base d'images fixes peut être décrite de différentes façons, notamment par des descripteurs visuels globaux de couleur, de texture, ou de forme (au niveau pixel). Plusieurs descripteurs sont proposés dans la littérature (Manjunath et al. 2002, Obeid et al. 2001, Tao 1999) et certains sont standardisés comme MPEG-7 (Manjunath et al. 2002). Chacun d'eux est défini selon l'information que l'on souhaite extraire de l'image. On privilégiera par exemple un descripteur de forme si l'on souhaite retrouver toutes les images contenant un clavier d'ordinateur. Par contre, un descripteur de couleur sera indiqué si l'on recherche des images de la mer ou

du coucher du soleil. Les requêtes les plus fréquentes impliquent et combinent les résultats de plusieurs types de descripteurs : par exemple, "retrouver toutes les images ayant une couleur et une texture semblables à celles d'une image requête donnée". Le traitement de ce type de requête exige de considérer plusieurs aspects parmi lesquels le choix des bons descripteurs, la recherche sur chacun des descripteurs choisis, et la définition d'une bonne stratégie de fusion des résultats obtenus sous forme de liste ordonnée. Il est légitime de se demander si l'on peut trouver des relations entre les descripteurs existants et si ces relations permettent une recherche plus rapide et plus efficace d'une image dans une base donnée.

L'objectif de notre travail est de combiner et d'exploiter deux techniques de fouille : le clustering et la recherche de règles d'association pour améliorer la recherche dans une base de données complexes telles les images. Après avoir décrit le système de recherche que nous développons afin de valider notre étude, nous nous focalisons essentiellement sur le temps de réponse à des requêtes, puis nous confrontons notre système à l'approche classique de recherche séquentielle avec fusion des résultats.

Cet article est organisé de la manière suivante : dans la section 2 nous présentons sommairement le principe de description des images fixes ainsi qu'une synthèse de l'état de l'art des différents usages multimédias des règles d'association. Nous présentons ensuite notre méthode de recherche dans la section 3 et discutons de nos résultats. La section 4 conclut l'article et présente nos perspectives à moyen et long terme.

2 Description d'images et usage des règles d'association

2.1 Description d'images fixes

La description d'une image a pour but de rassembler tous les éléments nécessaires à la caractérisation de l'image dans un contexte d'utilisation donné. Dans le domaine médical par exemple, la détection automatique du cancer du sein à partir des images de mammographie passe par l'identification et la localisation des cellules anormales. La nature et le type de descripteurs proposés par la communauté du traitement d'images dépendent totalement et précisément du but de la recherche (par exemple un descripteur pour trouver le visage de G. Bush). Quatre niveaux de description sont cependant envisageables (Zhang et al. 2001) : le pixel, l'objet, le niveau sémantique, et la connaissance. Au niveau pixel, une image est considérée comme un ensemble de points ou de pixels dont on calcule les caractéristiques de couleur, de texture, et/ou de forme. Il peut s'agir par exemple de la détection d'un cercle dans une image. Le but au niveau de l'objet est de pouvoir identifier dans une image des objets comme un ballon. Au niveau sémantique, l'on s'appuie sur la connaissance du domaine (médical, spatial, sportif, ...) pour extraire des concepts de haut niveau à partir des objets identifiés dans une image. Il est ainsi possible de distinguer les images d'un match de football. Le niveau de la connaissance s'appuie sur le niveau sémantique auquel il est rajouté des informations complémentaires décrivant le contenu. À une image de football par exemple, on peut rajouter des mots décrivant les équipes en compétition, les actions de tir au but, et le score du match.

Notre étude est volontairement limitée à la description visuelle des images qui sont traitées comme des ensembles de pixels. Dans ce cadre, étant donnée une image i, la description de i consiste à trouver un vecteur $d = f(i)$ qui résume les caractéristiques de l'image i.

Le descripteur d de l'image i est un vecteur de réels ou d'entiers de dimension n et f la fonction de calcul du descripteur. Cette fonction doit être robuste aux variations des conditions de production de l'image et à l'orientation de celle-ci (Smeulders et al. 2000). Nous limiterons notre étude à 5 descripteurs pour nos images décrivant respectivement : la couleur ($Color\,Layout$, $Scalable\,Color$), la texture ($Homogeneous\,Texture$, $Edge\,Histogram$) et la forme ($Region\ Shape$) ; tous proposés dans le standard MPEG-7 (Manjunath et al. 2002).

2.2 Usage des règles d'association

Le but de la technique des règles d'association est la génération d'associations informatives à partir d'une grande masse de données. Une formalisation de la technique est introduite pour la première fois dans (Agrawal et al. 1993).

Soit $I = \{i_1, i_2, \ldots, i_N\}$ un ensemble de N éléments distincts appelés items, objets ou attributs, et D un ensemble de transactions ayant comme attributs les éléments de I. *Une règle d'association* est une implication de la forme $A \rightarrow B$, où $A, B \subset I$, et $A \cap B = \phi$. A est appelé *antécédent* et B *conséquent* de la règle. Un ensemble d'objets ou d'items est appelé *itemset*. A chaque itemset, on associe une mesure statistique appelée *support*, et notée *supp*. Pour un itemset $A \subset I$, $supp(A) = s$, si la fraction de transactions de D contenant A est égale à s. Une mesure de confiance permet de caractériser la pertinence d'une règle. Pour $A \rightarrow B$, elle est définie par $conf = supp(A \cup B)/supp(A)$.

Déterminer des règles d'association consiste à extraire toutes les règles de la forme $A \rightarrow B$ ayant un support et une confiance supérieurs ou égaux à leurs seuils respectifs $minsupp$ et $minconf$ fixés à l'avance. Plusieurs algorithmes d'extraction des règles d'association tels que $Apriori$ (Agrawal et al. 1993), $Pascal$ (Bastide et al. 2002) sont proposés dans la littérature. Ces méthodes d'extraction marchent très bien avec les bases de données transactionnelles dont la structure est bien définie. Mais leur application aux images nécessite des adaptations. Une idée consiste à transformer les images et à appliquer les techniques classiques d'extraction de règles. Une image sera par exemple identifiée à une transaction et les différentes régions la constituant à des objets (Ordonez 1999). La répétition d'objets peut dans certaines conditions être porteuse d'informations. Un grand nombre de régions de couleur bleue dans une image donne une idée de l'intensité de la couleur bleue. La définition classique de règles d'association ne tient pas compte de cette possibilité de répétition et il est nécessaire dans ce cas de développer de nouveaux formalismes. C'est ainsi que Zaïane et al. (Zaïane et al. 2000) proposent une définition de *règle d'association avec objets récurrents* comme une implication de la forme :

$$\alpha P_1 \wedge \beta P_2 \wedge \ldots \wedge \gamma P_n \rightarrow \delta Q_1 \wedge \lambda Q_2 \wedge \ldots \wedge \mu Q_m$$

où $P_i, i \in [1..n]$ et $Q_j, j \in [1..m]$ sont des descripteurs d'images, et $\alpha, \beta, \gamma, \delta, \lambda, \mu$ sont des entiers indiquant le nombre d'occurrence des descripteurs. L'algorithme $MaxOccur$ décrit dans (Zaïane et al. 2000) est une adaptation de $Apriori$ à l'extraction des règles d'association avec objets récurrents.

Nous proposons une application des règles d'association à l'indexation et à la recherche par le contenu dans une base d'images fixes. Notre approche s'inscrit dans le cadre de la transformation des données multimédias et de l'application des techniques classiques d'extraction de règles. Nous utilisons plusieurs descripteurs et les images sont regroupées en clusters pour chaque descripteur. Nous identifions ensuite une image à une transaction et un numéro de clus-

ter pour un descripteur donné à un attribut. Le problème de répétition d'objets ou d'attributs ne se pose pas puisque les clusters sont identifiés de manière unique par descripteur. Parmi les travaux sur les règles d'association dans un contexte de recherche par le contenu multimédia, Djeraba utilise une technique d'indexation voisine de la nôtre en se sens qu'elle combine la construction des clusters et la détermination des règles d'association (Djeraba 2003). Sa base de travail est formée de plusieurs sous-groupes d'images classées thématiquement (animaux, plantes, ...) et l'idée consiste à caractériser ces groupes par des règles d'association. Lors de la recherche, une analyse de la requête à partir des règles d'association oriente le choix du sous-groupe d'images. Le travail de Djeraba a pour objectif l'amélioration de la qualité de recherche. Il ne s'intéresse pas au problème de fusion des résultats lorsque la recherche se fait suivant plusieurs descripteurs. Le temps de la recherche n'est pas évalué. De plus, l'approche est contrainte par la construction semi-automatique d'un dictionnaire nécessitant une labélisation des clusters. Dans notre approche par contre, nous restons au niveau du pixel et aucun étiquetage n'est nécessaire.

D'autres travaux exploitent les règles d'association (Bouet 2002, Aouiche et al. 2003). Le premier tente de capturer le niveau sémantique par la découverte des relations existantes entre les images d'une base de données pour réduire l'espace de recherche. L'approche d'Aouiche et al. destinée à l'auto-administration d'une base de données relationnelle, fonde le calcul de l'index sur les ensembles d'attributs les plus fréquemment utilisés dans les requêtes. Cette approche détermine des relations entre attributs (sous forme de motifs fréquents) à partir de la fréquence d'utilisation de ceux-ci. Elle fait donc une administration à partir d'un journal d'utilisation de la base (intervention indirecte de l'utilisateur) alors que nous faisons une indexation sans intervention de l'utilisateur. Nous ne supposons aucune organisation préalable de la base d'images fixes et nous ne limitons pas le nombre ainsi que le type de descripteurs utilisables.

3 Description de la méthode d'indexation et de recherche

Nous proposons une méthode utilisant la classification automatique et la recherche de règles d'association pour améliorer la recherche par le contenu dans une base d'images fixes. Notre approche se décompose en deux principales étapes fonctionnelles autour desquelles s'articule l'architecture de notre chaîne de traitement des images (voir Figure 1) : la procédure d'indexation et la procédure de recherche que nous détaillons ci-après.

3.1 L'indexation

L'indexation de la base se fait hors-ligne. Elle est composée de trois modules : le descripteur de données, le classifieur automatique, et le générateur de règles. Le descripteur de données calcule tous les descripteurs à partir d'un ensemble de fichiers d'images stockés dans un répertoire. Nous travaillons avec les descripteurs MPEG-7 de couleur ($ColorLayout$, $ScalableColor$), de texture ($HomogeneousTexture$, $EdgeHistogram$) et de forme ($RegionShape$). Notre système génère un fichier décrivant toutes les images pour chacun des descripteurs utilisés (étape I1 de la Figure 1). Le classifieur automatique organise les fichiers de descripteurs en clusters pour chaque descripteur, construit un index d'accès à chaque groupe formé et le stocke dans un fichier (étape I2). Chaque cluster est identifié par un numéro et la nature du descripteur pour lequel le cluster est calculé.

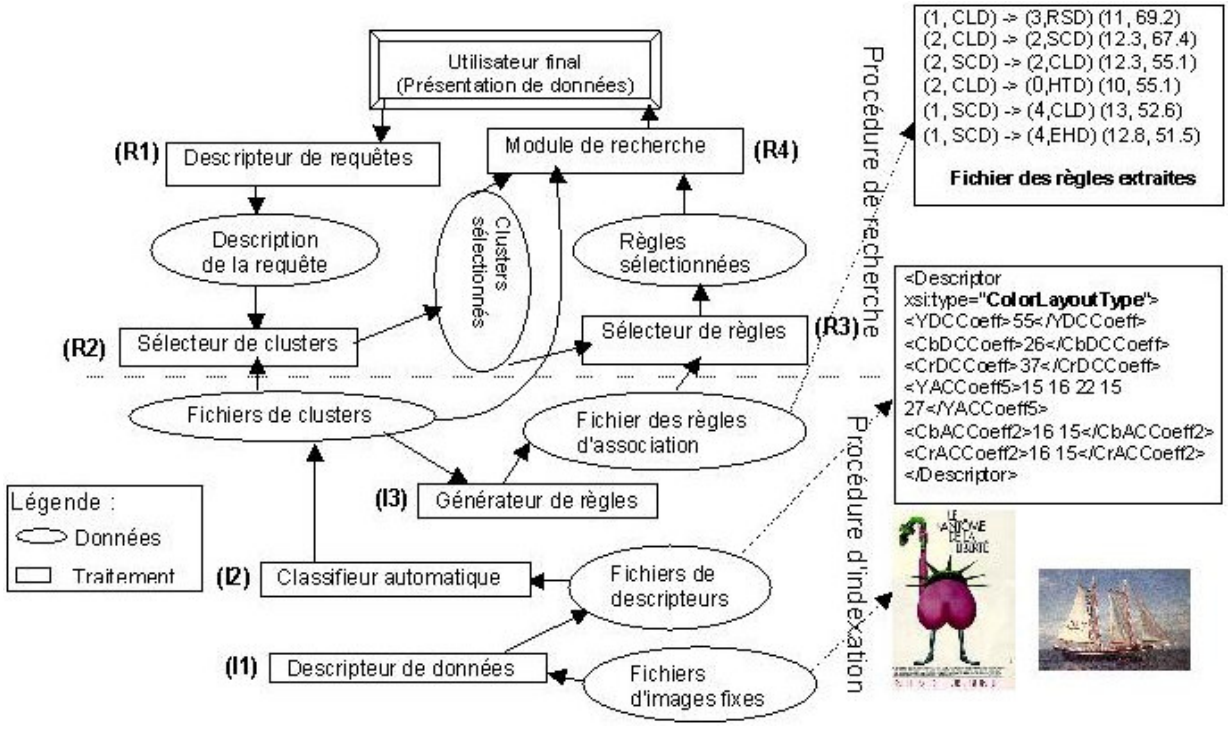


FIG. 1 – *Processus d'indexation et de recherche d'images fixes.*

Le générateur de règles utilise les fichiers de clusters produits par le classifieur automatique pour extraire des relations entre clusters sous forme de règles d'association (étape I3). Une règle d'association ainsi générée est une implication de la forme :

$(< num\ cluster >, < nom\ descripteur >)\ [(< num\ cluster >, < nom\ descripteur >)...] \rightarrow (< num\ cluster >, < nom\ descripteur >)\ (< s >, < c >)$

ayant la sémantique suivante :

$< num\ cluster >$ est le numéro de cluster pour le descripteur $< nom\ descripteur >$; $< s >$ et $< c >$ sont des pourcentages indiquant respectivement le support et la confiance de la règle étudiée. La partie gauche d'une règle est composée d'un ou de plusieurs couples $(< num\ cluster >, < nom\ descripteur >)$ identifiant les clusters. Nous limitons la partie droite à un seul couple.

3.2 La recherche

La recherche se fait en-ligne. L'utilisateur soumet au système une image requête. Son but est de retrouver toutes les images semblables à la requête spécifiée. Deux cas peuvent se présenter : soit l'utilisateur choisit les descripteurs suivant lesquels la recherche doit être faite, soit l'utilisateur n'a aucune idée des descripteurs calculés. Nous penchons pour le second cas

le plus fréquent. Le module de description de requête traite la requête soumise et produit un descripteur de chaque type dans la limite des descripteurs gérés par la procédure d'indexation (étape R1). Le sélecteur de clusters utilise les fichiers de clusters et la description de la requête afin de déduire pour chaque descripteur, le cluster dans lequel la requête se trouve (étape R2).

Nous utilisons les règles d'association pour réduire le nombre de clusters dans lesquels nous faisons de la recherche séquentielle. Le sélecteur de règles choisit parmi les règles disponibles celles qui décrivent les relations entre les clusters fournis par le sélecteur de clusters (étape R3). Une règle est choisie si tous les clusters impliqués dans la règle sont des éléments de l'ensemble des clusters sélectionnés à l'étape R2. Pour faciliter l'écriture, nous adoptons les notations abrégées CLD, SCD, HTD, EHD, et RSD pour les descripteurs respectifs $Color\ Layout$, $Scalable\ Color$, $Homogeneous\ Texture$, $Edge\ Histogram$, et $Region\ Shape$. Les clusters sont numérotés de 0 à 4 pour chaque descripteur. Supposons par exemple une requête I_q et les valeurs suivantes fournies par le sélecteur de clusters : $(2, CLD)$, $(2, SCD)$, $(1, HTD)$, $(3, EHD)$, et $(3, RSD)$. Les règles sélectionnées seront donc $(2, CLD) \rightarrow (2, SCD)$ $(12.3, 67.4)$ et $(2, SCD) \rightarrow (2, CLD)$ $(12.3, 55.1)$ (voir Figure 1 pour l'ensemble de toutes les règles du système).

Les clusters et les règles sélectionnés sont transmis au module de recherche (étape R4). Si aucune règle n'est sélectionnée, alors la recherche séquentielle est faite dans tous les clusters sélectionnés. Dans le cas contraire, la recherche séquentielle se fera uniquement dans les clusters n'apparaissant pas en partie droite de règle. L'exemple précédent est un cas particulier où la recherche séquentielle est faite uniquement dans les clusters $(1, HTD)$, $(3, EHD)$, et $(3, RSD)$. On aurait souhaité que la recherche se fasse aussi dans $(2, CLD)$ ou $(2, SCD)$ (mais pas dans les deux). Cette situation n'est pas encore gérée par le système et fera l'objet d'une étude plus élaborée. Les résultats de la recherche séquentielle dans les clusters sont ensuite fusionnés selon le principe suivant (Nepal 1999) :

1. pour chaque cluster C_i fourni par le sélecteur de clusters, retourner le prochain couple $(I, s_{C_i}(I))$ où I est l'image la plus proche de l'image requête I_q pour la mesure de score s_{C_i} (comprise entre 0 et 1), puis calculer $s_{C_j}(I)$ pour les clusters C_j où $i \neq j$
2. calculer $f_h = f(s_{C_i}(I_i), \ldots, s_{C_m}(I_m))$ où I_i est l'image du cluster C_i à l'itération courante. f est une fonction de synthèse des scores. Nous l'avons prise égale à la fonction min
3. calculer $s_Q(I) = f(s_{C_i}(I), \ldots, s_{C_m}(I))$ pour tout I de l'itération courante et mettre à jour l'ensemble $Y = \{I | s_Q(I) > f_h\}$
4. répéter 1, 2, et 3 jusqu'à ce que Y contienne le nombre d'images désiré, puis retourner $\{(I, s_Q(I) | I \in Y)\}$ à l'utilisateur.

3.3 Discussion

La méthode présentée ci-dessus est en cours d'implémentation en C++ sous Linux. Nous travaillons pour l'instant avec une base de 7727 images fixes mais nous envisageons plusieurs centaines de milliers d'images à long terme. Pour chacun des 5 descripteurs MPEG-7 utilisés, nous regroupons les images en clusters avec un algorithme de type $k - means$. La complexité temporelle du regroupement en clusters d'images ne nous préoccupe pas ici car l'indexation se fait hors-ligne. Nous avons maintenu le seuil de la confiance à 50% pour le calcul des règles d'association entre clusters avec l'algorithme $Apriori$ (Agrawal et al. 1993) afin de garantir

la bonne qualité des implications. La figure 2 montre les variations du nombre de règles en fonction du nombre de clusters pour les trois valeurs suivantes du seuil du support : 0.1%, 1%, et 10%. Nous avons travaillé avec le même nombre de clusters pour tous les 5 descripteurs. L'expérimentation peut être étendue en considérant des nombres de clusters différents pour les descripteurs. L'on constate que plus le nombre de clusters est élevé, moins on obtient de règles. Dans la suite des expérimentations, nous avons fixé le nombre de clusters à 5 pour chacun des descripteurs et le seuil du support à 10%. Ce sont en effet des valeurs pour lesquelles nous avons obtenu des règles de support significatif. Dans ces conditions, le système produit 6 règles dont le support varie entre 10% et 13%. Ce support relativement faible s'explique. En effet, la valeur du support est une fonction décroissante du nombre de clusters choisi par descripteur. Si l'on suppose par exemple une répartition uniforme des images dans chacun des 5 clusters pour chacun des descripteurs, alors le support des règles est majoré par 20%.

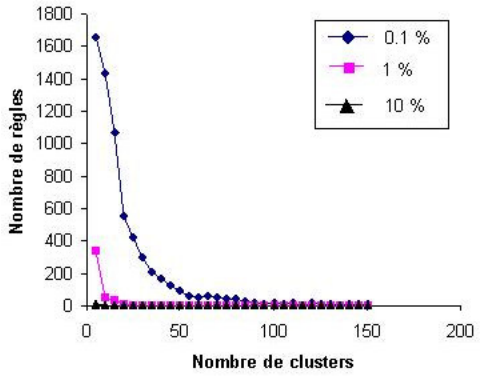


FIG. 2 – *Variations du nombre de règles en fonction du nombre de clusters pour 3 valeurs du seuil du support.*

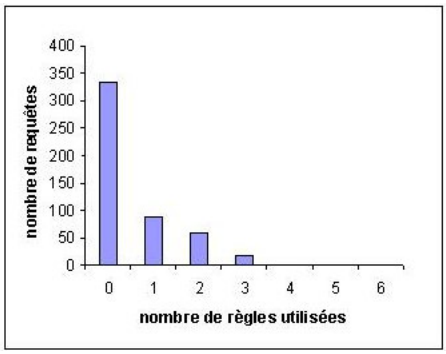


FIG. 3 – *Histogramme d'utilisation des règles.*

Le système conçu a été interrogé par 500 requêtes, toutes déjà présentes dans la base de 7727 images. Pour 165 d'entre elles, le système fait usage des règles d'association calculées, soit un taux d'utilisation de 33%. L'histogramme d'utilisation des règles est présenté à la figure 3. On constate que le système utilise rarement plus de trois règles dans le traitement d'une requête.

De façon générale la méthode proposée présente un temps de recherche inférieur au temps de recherche séquentielle avec fusion des résultats comme présenté dans le tableau 1. D'autre part, l'usage des règles d'association réduit le nombre de descripteurs à explorer et donc le temps recherche aussi.

Le tableau 2 décrit le comportement général de notre méthode et de celle de la recherche séquentielle sur un ensemble de 500 requêtes. L'on note par exemple que le temps moyen de recherche par notre approche reste voisin de celui de la recherche séquentielle. Nous pensons donc qu'il reste encore à faire pour améliorer ce résultat, par exemple en étudiant beaucoup plus finement les stratégies d'exploitation des règles d'association. L'introduction de la sélection des règles à partir d'une fonction de coût de calcul est envisageable. La recherche guidée par des règles d'association offre donc une perspective intéressante à explorer.

	Temps moyen de recherche séquentielle avec fusion	Temps moyen de recherche avec le système proposé
Usage des règles	46.50 secondes	44.77 secondes
Pas d'usage des règles	46.71 secondes	45.25 secondes

TAB. 1 – *Comportement de notre système par rapport à la recherche séquentielle avec fusion des résultats.*

	Recherche séquentielle avec fusion	Recherche avec le système proposé
Temps minimum (secondes)	3.96	2.12
Temps maximum (secondes)	257.82	257.90
Temps moyen (secondes)	46.64	45.09
Écart-type	23.26	23.77

TAB. 2 – *Récapitulatif du temps d'exécution des requêtes.*

Le problème du calcul automatique du nombre de clusters reste entier. Dans ce travail, nous adoptons une approche manuelle, mais la démarche pour déterminer le nombre de clusters reste introuvable. Une tentative de résolution du problème est présentée dans (Fernandez et al. 2002). Mais la complexité du regroupement en clusters est plus grande.

4 Conclusion et perspectives

Nous décrivons le principe de fonctionnement d'un système de recherche d'images par le contenu en cours de développement. Ce système sert de cadre pour l'étude de l'intérêt des règles d'association dans la recherche par le contenu. Dans notre démarche, nous utilisons 5 descripteurs MPEG-7 pour décrire à terme plusieurs milliers d'images fixes. Dans un premier temps, nous déterminons des clusters d'images pour chaque descripteur. Nous générons ensuite sous forme de règles d'association, les relations entre les différents clusters obtenus. Nous montrons que les règles d'association sont envisageables pour améliorer le temps de recherche.

- Nous nous proposons de travailler sur une base plus grande d'images dans la suite de nos expérimentations. Ainsi, le temps d'indexation ne sera plus négligeable et nous essayerons des techniques de clustering plus performantes que le $k - means$.
- D'après notre étude, les règles d'association obtenues sont d'un support faible. Il nous paraît donc intéressant d'explorer d'autres mesures afin d'en déduire les plus adaptées à l'évaluation des règles. Il est également utile d'étudier des stratégies plus fines de sélection et d'exploitation des règles pour en améliorer le taux d'utilisation par le système lors de la recherche.
- Nous nous sommes limité au cours de notre étude à l'évaluation du temps de recherche. La comparaison de la qualité des résultats du système proposé et de celle de la recherche

séquentielle avec fusion sera la prochaine étape de la validation de notre approche. Cette étape passe par une définition des mesures adaptées à l'évaluation de la qualité des résultats.

– Enfin, l'expérimentation des techniques statistiques voisines des règles d'association nous semble indispensable pour évaluer notre approche. Nous pensons par exemple à l'analyse des correspondances multiples.

Références

Agrawal R., Imielinski T., Swami A. (1993), Mining Association Rules Between Sets of Items in Large Databases, ACM SIGMOD International Conference on Management of Data, 1993, pp 207-216.

Aouiche K., Darmont J., Gruenwald L. (2003), Extraction de motifs fréquents pour l'auto-administration des bases de données, Revue des Sciences et Technologies de l'Information, 2003, p 547.

Bastide Y., Taouil R., Pasquier N., Stumme G., Lakhal L. (2002), Pascal : un algorithme d'extraction des motifs fréquents, Technique et science informatique, Vol 21, No 1, 2002, pp 65-95.

Bouet M., Khenchaf A. (2002), Traitement de l'information multimédia : Recherche du média image, RSTI - Ingénierie des systèmes d'information. Bases de données et multimédia, 7(5-6), 2002, pp 65-90.

Djeraba C. (2003), Association and Content-Based Retrival, IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No.1, 2003, pp 118-135.

Fernandez G., Meckaouche A., Peter P., Djeraba C. (2002), Intelligent Image Clustering, Lecture Notes in Computer Science, Vol. 2490, 2002, pp 406-419.

Manjunath B.S., Salembier P., Sikora T. (2002), Introduction to MPEG-7, John Wiley & Sons, 2002.

Nepal S., Ramakrishna M.V. (1999), Query Processing Issues in Image (Multimedia) Databases, Proceedings of the ICDE, 1999, pp 22-29.

Obeid M., Jedynak B., Daoudi M. (2001), Image indexing and retrieval using intermediate features, Proceedings of the 9^{th} ACM/ICM, 2001, pp 531–533.

Ordonez C., Omiecinski E. (1999), Discovering Association Rules based on Image Content, Proceedings of the IEEE Advances in Digital Libraries Conference(ADL'99), 1999, pp 38-49.

Smeulders A.W.M., Worring M., Santini S., Gupta A., Jain R. (2000), Content-Based Image Retrieval at the End of the Early Years, Vol. 22, No.12, 2000, pp 1349–1380.

Tao Y., Grosky W. (1999), Object-Based image retrieval using point feature maps, Proceedings of the International Conference on Database Semantics(DS-8), 1999, pp 59-73.

Zaïane O., Han J., Zhu H. (2000), Mining Recurrent Items in Multimedia with Progressive Resolution Refinement, Proceedings of the 16^{th} IEEE International Conference on Data Engineering(ICDE'00), 2000, pp 461–476

Zhang J., Hsu W., Lee M. L. (2001), Image Mining: Issues, Frameworks and Techniques, Proceedings of the 2^{nd} International Workshop on Multimedia Data Mining, 2001, pp 13-20.

Summary

Fixed image database can be described in several ways by global visual descriptors of color, texture, or form (pixel level). Most frequent queries imply and combine results of several type of descriptors such as: "retrieve all images that have similar color and similar texture to the given example image". To retrieve more efficiently and more effectively an image of a large database, we exploit combinations of descriptors. In this papier, we study the effet of association rules between clusters of descriptors on fixed image content based retrieval. We first describe the system retrieval which is used as our study framework, then we set our focus on query response time. We finally compare our system with the sequential retrieval followed by result fusion.

Keywords: association rules, content based retrieval, fixed image, clustering.

Tableau de Bits Indexé (TBI) pour la Recherche de Séquences Fréquentes - Application à l'enquête MENAGE

Lionel Savary, Karine Zeitouni

Laboratoire PRISM, Université de Versailles
45, avenue des Etats-Unis, 78035 Versailles Cedex, France
Lionel.Savary@prism.uvsq.fr, Karine.Zeitouni@prism.uvsq.fr

Résumé. La recherche de séquences fréquentes est utile dans l'étude des comportements et habitudes d'une population. Notre application se situe dans le domaine de l'analyse de la mobilité de la population et plus précisément sur les données de l'enquête MENAGE. Dans cet article, nous proposons une nouvelle méthodologie pour la découverte de séquences fréquentes dans les programmes d'activités L'algorithme proposé recherche des motifs fréquents en respectant l'ordre des articles. Ses performances ont été optimisées grâce aux structures et aux index proposés. En effet, il n'effectue qu'une seule lecture dans la base de données tout en ayant une représentation des données peu gourmande en mémoire et performante lors de la recherche de séquences fréquentes.

1. Introduction

La recherche de séquences fréquentes a été introduite en 1995 par Agrawal [Agrawal et al. 1995] dans le contexte de l'analyse de la consommation. Elle exploite des bases de données dites transactionnelles décrivant les achats de produits par client et par transaction. Depuis, différents algorithmes ont été proposés dans ce domaine, pour améliorer les coûts de recherche en terme de performance et d'allocation d'espace mémoire [Jay et al. 2002], [Zaki 1998]. Puis les travaux se sont principalement orientés vers la recherche de séquences fréquentes d'achats de produits par le même client dans des transactions successives, de recherche de séquences fréquentes dans les séquences d'ADN [Han et al. 2001] ou d'occurrences fréquentes de mots en fouille de textes [Ahonen 1999].

Dans ce papier, nous nous intéressons à une application particulière[1] portant sur les données de l'enquête MENAGE [CERTU 1998] et plus particulièrement sur l'étude des déplacements journaliers effectués par la population de la ville de Lille, leurs modes de transport. Dans cette étude, nous possédons des données sur les activités quotidiennes (ex. : partir de la maison, déposer les enfants à l'école, aller au travail, retourner à la maison) effectuées par chacun des habitants de la ville de Lille. La ville a été découpée en 60 zones et chaque zones sont elle mêmes découpées en sous zones. Ainsi pour chaque personne habitant dans une zone de Lille, nous possédons des informations concernant : chaque personne d'un ménage (âge, sexe, emplois,…), son programme d'activités [Wang et al. 2001], les différents modes de transports utilisés entre chaque activité. Par exemple au cours d'une journée, une personne peut : partir de chez-elle, déposer les enfants à l'école, aller au travail, récupérer les

[1] Ce travail est partiellement financé par le projet européen HEARTS du 5° PCRD – référence du contrat : QLK4-CT-2001-00492

enfants à l'école et rentrer chez elle. Nous obtenons ainsi une suite d'activités appelée programme d'activités effectué par cette personne au cours d'une journée. Ce que nous pouvons résumer par : (Maison-Ecole-Travail-Ecole-Maison). Le programme d'activités de différentes personnes peut bien entendu être le même ou présenter des similitudes. Si nous ramenons le programme d'activité de chaque individu à une suite de séquences d'activités où chaque activité serait codé par une valeur (ex. : M, E, T, E, M codant Maison, Ecole, Travail, Ecole, Maison), il nous serait alors possible de déterminer les séquences fréquentes d'activités effectuées par une majorité d'habitants de Lille. Ceci permet d'analyser la mobilité de la population urbaine de Lille. De même, si on code plutôt les modes de transport, les horaires ou les durées, on peut déterminer une typologie des modes de transport utilisés, ou des horaires, etc. .

Dans ce papier, nous proposons un nouvel algorithme pour la recherche de séquences d'activités fréquentes. L'article est organisé comme suit : la deuxième partie présente un état de l'art des travaux se rapportant le plus à ce domaine, la troisième partie détaille l'algorithme proposé et enfin une conclusion générale résume la contribution et trace les perspectives.

2. Travaux liés

La plupart des travaux sur la recherche de séquences fréquentes se situent dans le domaine de l'analyse de la consommation. A l'origine l'algorithme *Apriori* de [Agrawal et al 1994] ne considérait pas des séquences de transactions mais juste des transactions. Il est en outre peu performant car il effectue de multiples parcours de la base de données pour déterminer les sous-ensembles d'articles fréquents. Trois algorithmes traitant les séquences de transactions – listes d'ensembles d'articles par client - sont développés et comparés dans [Agrawal et al. 1995] : *AprioriAll*, *AprioriSome* et *DynamicSome*. L'algorithme *AprioriAll* est une adaptation de l'algorithme *Apriori* pour les séquences où la générations de candidats et les calculs de supports diffèrent. Contrairement à *AprioriAll*, *AprioriSome* ne compte que les séquences fréquentes maximales. Cet algorithme utilise une fonction *next()* qui retourne la taille de séquences candidates à générer pour la prochaine passe. Les séquences fréquentes maximales de tailles inférieures qui n'ont pas été calculées sont déterminées dans une phase *retour*. La valeur de *next(k)* augmente avec Pk = |Lk|/|Ck|, où Lk représente l'ensemble de séquences fréquentes de taille k et Ck l'ensemble de candidats de taille k générés. L'algorithme *DynamicSome* fonctionne selon le même principe qu'*AprioriSome* mais utilise un saut par multiple de *pas* donné par l'utilisateur. Par exemple si *pas* = 3, alors les séquences fréquentes de tailles 1,2 et 3 sont calculées puis les séquences de tailles 6,9 et 12 sont générées à partir des séquences de tailles 3, 6 et 9.

L'algorithme SPAM de [Jay et al. 2002] utilise une représentation bitmap des séquences de transactions, après avoir chargé toute la base de données dans un arbre lexicographique. L'inconvénient est qu'il considère que ces structures de données résident entièrement en mémoire principale. L'algorithme n'effectue donc ici qu'un seul parcours de la base de données pour la charger en mémoire, si celle-ci est de taille suffisante.

3. Algorithme tableau de bits indexé (TBI)

On s'intéresse maintenant au cas particulier où les séquences considérées sont basiques car composés d'articles et non d'ensembles d'articles comme dans les séquences de transactions vues ci-dessus. En effet, ce qui nous importe dans l'application est de rechercher

les séquences fréquentes dans les programmes d'activités des personnes sondées. Ce qui revient à rechercher des enchaînements d'activités, de modes, d'horaires… répétitifs et pouvant caractériser un groupe d'individus.

3.1 Principe de l'algorithme

Notons qu'une séquence est dite fréquente si elle est « incluse » dans un nombre de séquences de la base supérieur au seuil de support donné par l'utilisateur. L'inclusion entre deux séquences s1= (a1, …, an) et s2= (b1, …, bm) : s1 ⊂ s2, est définie par :

∃ bi1=a1, …, bin=an tel que i1 < i2< …<in

L'algorithme que nous proposons est basé sur un tableau de bits indexé (TBI). A ce tableau est associé un vecteur permettant de coder toutes les séquences enregistrées dans la base. La longueur du vecteur de séquences (VS) détermine donc la largeur du tableau (voir figure ci-dessous). Dans ce tableau, ne sont représentés que les séquences distinctes de la base de données. La longueur du tableau est donc limitée au nombre de séquences distinctes, augmentant le taux de compacité de cette représentation. A titre d'exemple, dans notre base de données, sur 13054 individus, il n'existe que 3429 séquences distinctes.

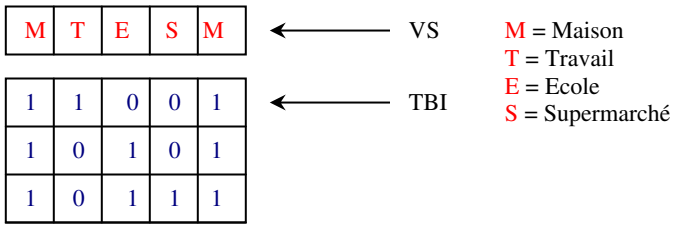


FIG. 1 - *Tableau de bit indexé*

Dans la figure 3, le vecteur de séquences est constitué de 5 activités dans l'ordre (M,T,E,S,M). Dans cet exemple, nous supposons que la base de données est composée de trois séquences distinctes de tailles 5 codées dans le TBI. Les bits du TBI placés à 1 indiquent les articles présents dans une séquence particulière, en correspondance avec le VS. Dans cet exemple, les 3 séquences distinctes sont : (M, T, M), (M, E, M) et (M, E, S, M).

TBI (Seuil-support)
- Gen-vecteur-séquences () //génère le vecteur de séquences
- Coder et Insérer chaque séquence dans le TBI
- **Pour** k = 1 à Tmax //Tmax = taille maximale des séquences de la base
  - Générer Ck
  - Gen-séquences-fréquentes (Seuil-support)
- **Fin Pour**

FIG. 2 - *Algorithme général de TBI*

L'algorithme n'effectue qu'une seule lecture dans la base de données, lors de laquelle le nombre total de séquences, les fréquences par séquence distincte ainsi que le nombre de séquences par taille sont calculés. Ce qui permet de calculer le support de chaque séquence générée. Les séquences sont classées par taille décroissante dans le TBI. Un index, associé au TBI, permet un accès direct aux séquences selon leur taille et donc de comparer efficacement

les séquences d'une certaine taille directement avec les séquences de même taille ou de tailles supérieures (voir figure ci-dessous).

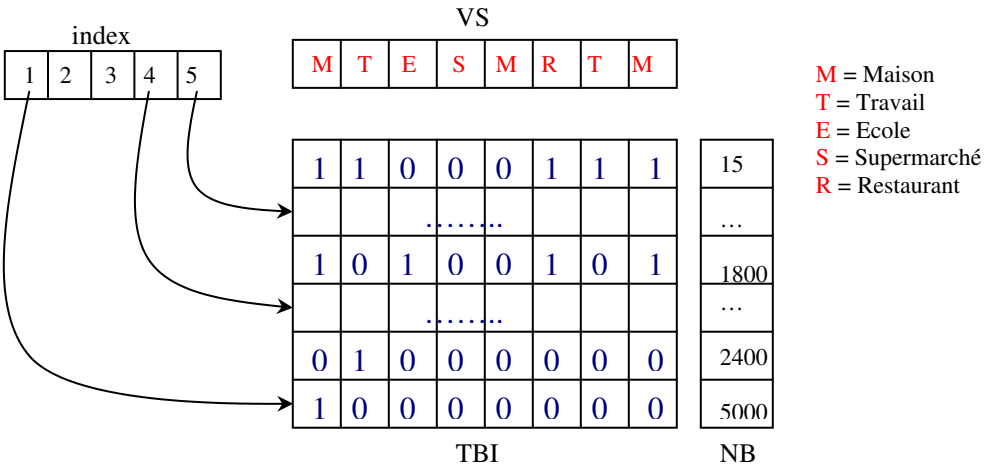


FIG. 3 - *Structure et représentation des données*

Dans l'exemple de la figure 3, le TBI contient l'ensemble des séquences distinctes de la base. Celle-ci contient des séquences de taille allant de 1 à 5. Le tableau NB associé au TBI renseigne la fréquence de la séquence. Ainsi la séquence (M, T, R, T, M) de taille 5 se retrouve 15 fois dans la base de données.

Une fois l'index, VS, le tableau NB construits et les dimensions du TBI déterminées, les séquences distinctes peuvent alors être codées dans le TBI suivant leur taille.

3.2 Génération du vecteur de séquences (VS)

Le vecteur de séquences est créé lors de la première lecture de la base suivant l'algorithme de la figure 4 ci-dessous :

Gen-vecteur-séquences() :
- VS = ϕ //VS vide au départ
- **Pour** chaque séquence *s* de la base de données
  - **Pour** chaque article *a* de *s*
  - **Si** *a* ∉ VS
    - **Si** ∃ *b* ∈ *s* tel que *b* ∈ VS et indice(*b*) > indice(*a*) dans *s* et (indice(*b*) > indice(dernier article de *s* inséré)) dans VP
      - Insérer *a* avant *b*
    - **Sinon** insérer *a* à la fin de VS
  - **Fin Pour**
- **Fin Pour**

FIG. 4 - *Génération du vecteur de séquences*

Ainsi toutes les combinaisons de séquences rencontrées dans la base sont représentées dans le vecteur VS. En prenant l'exemple des quatre séquences de la figure 3, supposons que nous voulons construire le vecteur de séquences avec les séquences dans l'ordre : (MTRTM), (MERM), (T), puis (M). Le vecteur de séquences VP étant vide au départ, on commence par insérer la première séquence. Ce qui nous donne VP = (MTRTM). Puis pour la deuxième séquence, le premier article M se trouvant déjà dans VP, on ne fait rien. Pour le deuxième article (E), celui-ci n'appartenant pas à VP, on regarde s'il existe dans cette séquence un article *b* d'indice supérieur à E et appartenant à VP tel que l'indice de *b* dans VP est supérieur à l'indice de M (dernier article de cette séquence inséré dans VP). Nous constatons que R appartient à VP et indice(R) > indice(E) dans *s* et indice(R) > indice(M) dans VP. On insère donc E avant R. Ce qui nous donne VP = (MTERTM). Le troisième et le quatrième articles de la deuxième séquence se trouvant déjà dans VP et situés après l'article E, on ne fait rien. Pour la troisième et la quatrième séquence respectivement, les articles T et M appartenant à VP, on ne fait rien. On obtient finalement VP = (MTERTM) représentant la combinaison de ces quatre séquences.

3.3 Génération de candidats et recherche de séquences fréquentes

Les candidats sont générés suivant le même principe que dans l'algorithme *Apriori* sauf qu'ici, l'ordre ainsi que la répétition d'articles dans les séquences générées sont pris en considération (voir figure 5 ci-dessous).

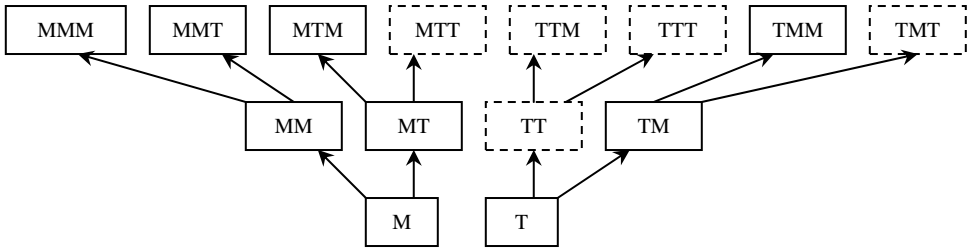


FIG. 5 - *Génération de candidats*

Dans cet exemple, les candidats de taille 2 et 3 sont générés à partir des articles fréquents {M} et {T}. Si la séquence générée (TT) de taille 2 est non fréquente, alors les séquences (MTT), (TTM), (TTT) et (TMT) ne peuvent être fréquentes et ne seront pas prises en compte. De plus, l'ordre étant important, les séquences (MT) et (TM) sont perçues comme deux séquences différentes.

Pour la recherche de séquences fréquentes, on commence par rechercher les articles fréquents codés dans le TBI ; puis on génère les séquences de taille 2 à partir des articles fréquents trouvés précédemment et ainsi de suite. Une fois les candidats de taille K générés, il suffit de parcourir le VS pour déterminer l'emplacement exact de chaque candidat dans le TBI. La valeur index[K] du tableau index indique la ligne du TBI à partir de laquelle les séquences de tailles égales ou supérieures à K sont représentées (voir algorithme ci-dessous).

Gen-séquences-fréquentes (Seuil-support) **:**
- Seuil-support // seuil de support en entrée
- Hmax //hauteur du tableau
- Ck // ensemble des séquences candidates générées de tailles k
- Lk // ensemble des séquences fréquentes de tailles k
- TBI // tableau de bits indexés
- index // tableau d'indexes
- NB // tableau contenant les fréquences de chaque séquence
- Lk = ∅ // ensemble des séquences fréquentes de tailles k
- **Pour** toutes les séquences $s \in$ Ck
  - **Pour** toutes les lignes *l* du TBI de la ligne index[k] à Hmax
  - **Si** $s \subset l$
    - *s*.count = *s*.count + NB[k] // fréquence de *s*
  - **Fin Pour**
  - **Si** *s*.count >= Seuil-support
    - Lk= Lk ∪ *s*
- **Fin Pour**

FIG. 6 - *Génération des séquences fréquentes*

Conclusion et perspectives

Cet article propose un algorithme TBI adapté à la recherche de séquences fréquentes. Il a été développé en vue de son application à la recherche de motifs fréquents dans les programmes d'activités dans les bases de données sur la mobilité de la population en zone urbaine et plus précisément aux données de l'enquête MENAGE. Cet algorithme n'effectue qu'un seul parcours de la base de données. Seules les séquences distinctes sont représentées et codées en format binaire en mémoire principale. Ce qui a pour avantage d'occuper peu d'espace mémoire et de permettre une comparaison rapide par opérateur binaire entre les séquences codées dans la table indexée et les séquences générées.

Comme perspective, l'implémentation permettra la validation expérimentale de cet algorithme et de tester son intérêt dans le domaine cité. D'autres domaines pourront être explorés tels que la recherche de séquences fréquentes dans les séquences d'ADN ou l'extension aux séquences de transactions.

Références

[Agrawal et al. 1994] Agrawal R., Srikant R.: Fast Algorithms for Mining Association Rules. In Proc. of the 20th Int. Conf. Very Large Data Bases (VLDB), Santiago, Chile, Septembre 1994.

[Agrawal et al. 1995] Agrawal R., Srikant R.: Mining sequential patterns. In Proc. of the 11th Int'l Conference on Data Engineering, Taipei, Taiwan, March 1995.

[Ahonen 1999] Ahonen H. Finding all maximal frequent sequences in text. ICML 1999, Machine Learning in text Data Analysis Workshop. Bled, Slovenia.

[CERTU 1998] Ministère de l'Equipement, des Transports et du Logement. L'enquête ménages déplacements « méthode standard ». Collections du Certu. Octobre 1998. ISSN 1263-3313.

[Jay et al. 2002] Jay A., Johannes .G, Tomi .Y, Jason F. Sequential Pattern Mining using A Bitmap Representation. SIGMOD pp429-435, July 2002, Edmonton, Alberta, Canada. Isbn 1-58113-567-X.

[Wang et al. 2001] Wang D., Tao C., A spatio-temporal data model for activity-based transport demand modeling.. International Journal of Geographical Information Science, 2001, vol.15, no 6, 561-585.

[Han et al. 2001] Han, J., Jamil, H. M., Lu, Y., Chen, L., Liao, Y. and Pei, J. DNA Miner: A system prototype for mining DNA sequences. In the proc. Of the ACM SIGMOD International Conference on the management of data, Day 21-24, 2001, Santa Barbara, CA, USA.

[Zaki 1998] Zaki M. J., Efficient Enumeration of Frequent Sequences. Int. Conference on Information and Knowledge Management, November 1998, Washington DC.

Sélection automatique d'index dans les entrepôts de données

Kamel Aouiche, Jérôme Darmont, Omar Boussaid

Laboratoire ERIC, Université Lumière – Lyon 2
5 avenue Pierre Mendès-France
69676 Bron Cedex
{kaouiche, jdarmont, boussaid}@eric.univ-lyon2.fr

Résumé. L'efficacité de l'interrogation d'un entrepôt de données est liée à sa conception physique. Cette conception repose sur la sélection d'index pertinents et leur combinaison avec les vues matérialisées. La sélection d'index est un problème NP–complet car le nombre d'index est exponentiel en nombre total d'attributs dans la base. Il faut donc concevoir et mettre en œuvre des méthodes permettant de réduire cette complexité afin de recommander un ensemble d'index (configuration). Dans cet article, nous proposons une méthode pour la sélection d'index basée sur la fouille de données (recherche des motifs fréquents, classification). Le contexte d'extraction de connaissances est construit après l'analyse des requêtes du journal des transactions exécutées. Les index de la configuration obtenue sont ensuite créés après une étape d'optimisation liée aux spécificités du SGBD utilisé.

1 Introduction

L'administrateur d'un Système de Gestion de Base de Données (SGBD) prend plusieurs décisions concernant des tâches d'administration, telles que la conception logique ou physique des bases de données, la gestion de l'espace de stockage, le réglage de performance (*performance tuning*), etc. L'une des plus importantes est la conception physique des bases de données, qui inclut l'organisation des données et l'amélioration de l'accès à ces données. Pour améliorer les temps d'accès, l'administrateur emploie en général des index pour rechercher rapidement les informations nécessaires à une requête sans parcourir toutes les données. Cependant, la mise à jour des données doit être propagée sur les index. Cela engendre un coût de maintenance supplémentaire.

La sélection d'index est difficile car leur nombre est exponentiel en nombre total d'attributs dans la base. C'est pourquoi nous nous intéressons au problème de sélection automatique d'un ensemble d'index (configuration) minimisant le coût d'exécution des requêtes. Dans ce sens, nous avons proposé une méthode de sélection d'index dans les bases de données basée sur des techniques de fouilles de données [ADG03a, ADG03b].

Par ailleurs, le problème de sélection d'index est aussi posé dans les entrepôts de données, mais certaines adaptations sont nécessaires. En effet, les entrepôts contiennent en général de très gros volumes de données. L'indexation de ces données donne donc lieu à des index volumineux. De plus, la sélection d'index se fait en conjonction avec la matérialisation des vues. L'espace nécessaire aux vues et aux index est de ce fait

très grand. La sélection d'index doit donc se faire en prenant en compte l'espace de stockage disponible. D'autre part, les rafraîchissements de l'entrepôt sont réalisés périodiquement quand le système est hors ligne. Le coût de maintenance des index doit donc être pris en compte de manière différente que dans le cas des bases de données.

Dans cet article, nous étendons dans un premier temps notre approche grâce à une analyse plus fine des requêtes de la charge, à la détermination des techniques d'indexation les plus appropriées aux index, à l'établissement d'un ordre des attributs constituant les index multi–attributs et à la prise en compte des spécificités du SGBD utilisé lors de la création des index. Par la suite, nous proposons des adaptations dans le contexte des entrepôts de données afin de prendre en compte la contrainte sur l'espace disponible pour stocker les index et de sélectionner des techniques d'indexation propres aux entrepôts.

Cet article est organisé comme suit. Après un état de l'art sur les solutions proposées pour le problème de sélection d'index (Section 2), nous détaillons l'approche que nous proposons (Section 3). Nous terminons par une conclusion et des perspectives de recherche (Section 4).

2 État de l'art

2.1 Sélection d'index dans les bases de données

Le problème de sélection d'un ensemble d'index pour une base de données a été étudié depuis les années 70. Il est NP–Complet [Com78]. En effet, ce problème réduit à une seule table de n attributs candidats peut théoriquement être résolu en évaluant le coût de 2^n ensembles d'index possibles. Cela induit un coût de calcul exponentiel. Il s'agit donc de trouver un ensemble d'index proche de l'optimal sans évaluer le coût de tous ces ensembles.

On peut distinguer deux grandes familles de travaux proposant des solutions au problème de la sélection d'index. Dans la première famille, une fonction de calcul de coût basée sur un modèle mathématique est élaborée pour estimer le coût d'un ensemble d'index [ISR83, BMT85, BP90, CBC93a, CBC93b, KLT03, FR03]. Par exemple, Kratica *et al.* utilisent une optimisation par un algorithme génétique [KLT03]. Le système DINNER [FR03] exploite une base de connaissances et représente les requêtes d'une charge sous forme de graphes de solutions. Une solution décrit les chemins d'accès (avec index ou non) parcourus pour exécuter une requête. Il utilise ensuite plusieurs heuristiques pour élaguer les mauvaises solutions.

Dans la deuxième famille, l'optimiseur de requêtes du SGBD est utilisé pour évaluer le coût de chaque index [FST88, FON92, CN97, CN98, VZZ$^+$00, ACN00]. Frank *et al.* proposent un outil d'aide à la décision pour l'administrateur qui, à partir d'une charge et d'un ensemble d'index donnés, fournit une estimation du coût global de chaque index grâce à une communication constante avec l'optimiseur de requêtes [FON92]. L'outil de sélection d'index IST (*Index Selection Tool*) [CN97, CN98, ACN00] développé par Microsoft au sein du SGBD SQL Server, exploite une charge et en extrait une configuration d'index candidats mono–attributs. Un algorithme glouton permet de sélectionner les meilleurs index de cette configuration grâce à des estimations de coût effectuées

par l'optimiseur de requêtes. Le processus est ensuite réitéré pour générer des index sur deux attributs à partir des index mono-attributs, et ainsi de suite pour les index multi-attributs de taille supérieure. Pour DB2, un algorithme de sélection d'index génère des index candidats pour chaque requête séparément, en injectant dans la base des index dits virtuels [VZZ+00]. Ces index virtuels sont des sous–ensembles de tous les index possibles choisis par un algorithme dont les détails ne sont pas mentionnés. Les index proposés sur une table donnée sont combinés en les sélectionnant dans un ordre décroissant du ratio gain–espace de stockage.

2.2 Sélection d'index dans les entrepôt de données

Dans les entrepôts de données, deux familles d'algorithmes de sélection d'index existent : des algorithmes pour optimiser le temps de maintenance [LQA97] et d'autres pour optimiser le temps d'exécution des requêtes [GHRU97, ACN01, GRS02]. Dans les deux cas, l'optimisation est réalisée sous la contrainte de l'espace de stockage.

Gupta *et al.* proposent un algorithme glouton utilisant un modèle de coût [GHRU97]. Cet algorithme parcourt un multi–graphe bipartie pour recommander un ensemble de vues et d'index minimisant le coût d'exécution des requêtes. Ce multi–graphe relie les index et les vues aux requêtes où ils sont présents. Agrawal *et al.* proposent un outil implanté sous SQL Server [ACN01]. Ce dernier commence par énumérer tous les index et vues pouvant contribuer à la réduction du temps d'exécution d'un ensemble de requêtes donné. Une réduction du nombre de candidats (index et vues) est ensuite effectuée par l'optimiseur de requêtes. Un ensemble final d'index et de vues matérialisées est alors proposé. Une approche heuristique utilisant un algorithme glouton a également été proposée [GRS02]. Cet algorithme admet en entrée un ensemble de vues et d'index pour choisir progressivement les index les plus avantageux. Le choix est réalisé en comparant le coût des plans d'exécution possibles, générés par l'optimiseur, de chaque requête de la charge.

2.3 Motivation et discussion

La sélection d'index soit utilise une fonction mathématique, soit fait appel à l'optimiseur de requêtes du SGBD. Une fonction de coût est basée sur des hypothèses théoriques, par exemple, "la fréquence d'insertion et de suppression des n–uplets d'une table est telle que le nombre total des n–uplets dans cette table reste constant pour deux choix consécutifs d'un ensemble d'index", ou "le nombre d'index n'est pas limité par table", qui ne sont pas toujours réalisables dans la pratique. Par contre, l'optimiseur de requêtes utilise des statistiques et un modèle de coût inhérent à un SGBD donné. La sélection d'index est donc dépendante du SGBD et doit être adaptée pour un autre SGBD. De plus, le coût de communication avec l'optimiseur peut être important si le nombre d'index candidats à évaluer est grand.

L'approche de sélection d'index que nous proposons dans cet article est similaire à celle de Microsoft. Cependant, l'ensemble d'index candidats n'est pas obtenu par une heuristique qui fait constamment appel à l'optimiseur de requêtes mais en appliquant des techniques de fouille de données à la charge. Le résultat est directement une configuration d'index candidats dont le coût peut être évalué soit grâce à une fonction de coût

soit par l'optimiseur de requêtes. Dans le second cas, le coût de communication avec l'optimiseur est réduit car le nombre d'index candidats à évaluer est moins important.

Dans les entrepôts, le problème devient plus crucial du fait de la grande volumétrie des données. Dans ce contexte, notre approche contribue efficacement à la sélection d'index en tenant compte de l'espace de stockage disponible.

3 Fouille de données pour la sélection d'index

L'approche que nous proposons (Figure 1) exploite le journal des transactions (ensemble de requêtes résolues par le SGBD) pour recommander une configuration d'index (ensemble d'index) améliorant le temps d'accès aux données.

La méthode proposée analyse la charge pour chercher les attributs pouvant être utiles lors de l'exécution d'une requête s'ils sont indexés. Cette analyse génère également un ensemble d'attributs sur lesquels il n'est pas souhaitable de construire des index. Les attributs trouvés précédemment sont stockés dans les matrices M_{Select} et M_{update}, respectivement. La combinaison de ces matrices permet de construire une matrice "requêtes–attributs". Dans les entrepôts de données, la matrice M_{update} n'existe pas car le rafraîchissement des entrepôts se fait hors ligne (*off line*). La matrice "requêtes–attributs" est donc construite directement. Parallèlement à l'analyse de la charge, nous déterminons la technique d'indexation appropriée à chaque index candidat. Les techniques d'indexation diffèrent suivant l'environnement utilisé : base ou entrepôt de données. La matrice "requêtes–attributs" correspond au contexte d'extraction utilisé pour générer les index de la configuration en utilisant les techniques de fouilles de données telles que la recherche des motifs fréquents ou la classification. Enfin, l'ordre des attributs dans les index multi–attributs est établi et un processus d'élagage est proposé avant de créer les index de la configuration. Nous détaillons chacune des ces étapes dans les sections suivantes.

3.1 Analyse de la charge

Les requêtes présentes dans la charge (journal des transactions) sont traitées par un analyseur de requêtes SQL afin d'extraire tous les attributs susceptibles d'être des supports d'index. Dans un environnement de base de données, la charge extraite est divisée en deux parties : une partie ne contenant que les requêtes d'interrogation (SELECT), notée $Part_{select}$ et une deuxième partie ne contenant que les requêtes de mises à jour (UPDATE, INSERT, DELETE), notée $Part_{update}$. Dans les requêtes des deux parties, le choix d'un attribut à indexer n'est pas dû au hasard. En effet, une requête d'interrogation comportant des jointures ou des sélections peut être très coûteuse si elle opère de manière séquentielle (sans utilisation d'index). Il est alors judicieux de construire des index sur les attributs utilisés pendant la recherche. Pour ces mêmes raisons, il est recommandé de créer des index sur les attributs servant à la mise à jour. Par contre, il n'est pas souhaitable de créer des index sur les attributs mis à jour (par exemple, les attributs d'une clause SET) car ils engendrent un coût de maintenance supplémentaire. Partant de ce constat, nous avons établi des règles pour recommander des attributs à indexer suivant chaque type de requête.

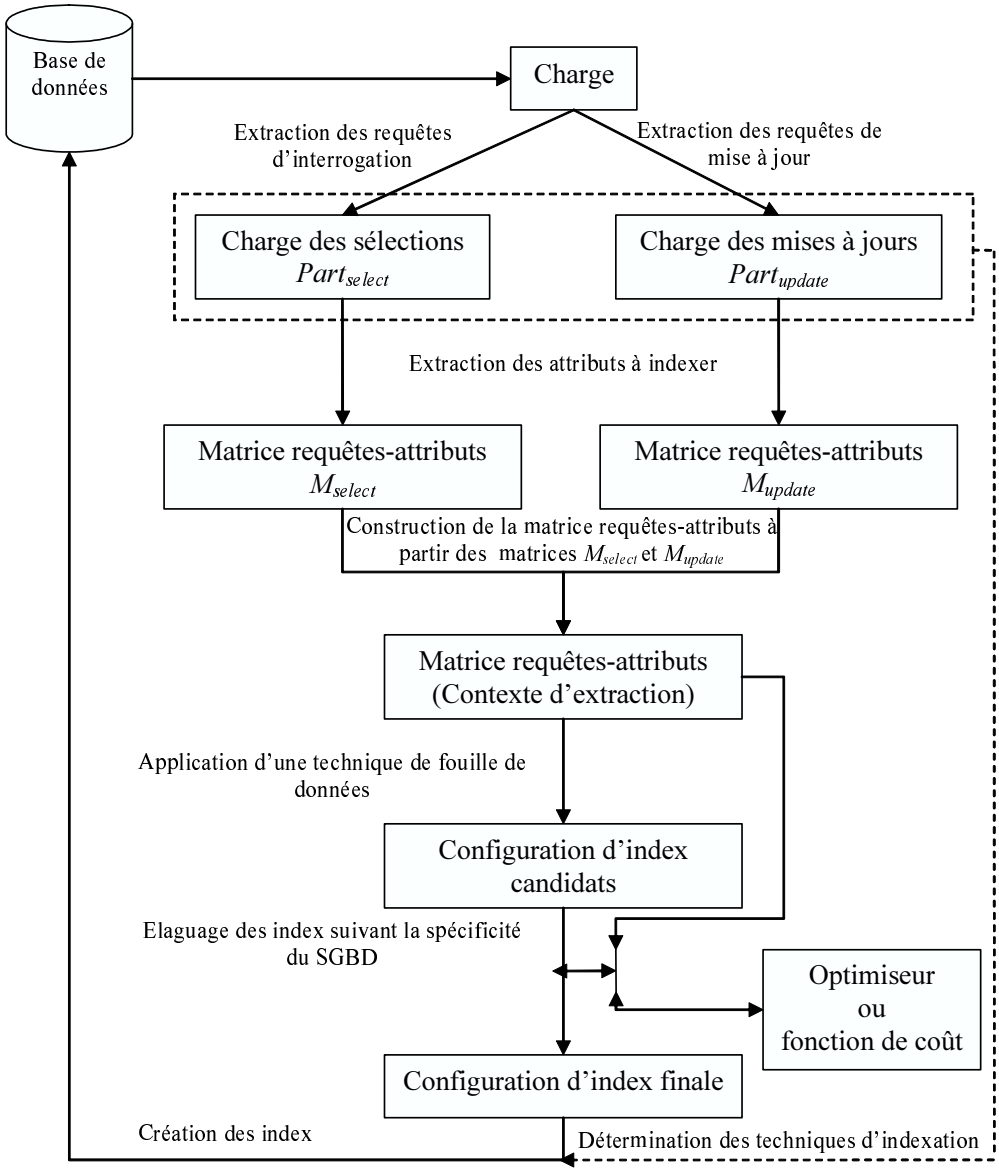


FIG. 1 – Démarche proposée pour la sélection d'index

Dans les entrepôts de données, les requêtes sont de type SELECT en lecture seule (*read only*) et les rafraîchissements (*batch update*) sont réalisées périodiquement [VG99]. Dans ce cas, nous construisons directement la matrice "requêtes–attributs".

Illustrons quelques règles à travers les exemples suivants. Soit un prédicat de restriction qui a l'une des formes suivantes : $t.a$ `between` v_1 `and` v_2, $t.a\,\theta\,v_2$, $t.a$ `like` v_1 `ou` $t.a$ `in` $(v_1, v_2, ...)$

où :

- t est un nom ou un alias d'une table,
- a est un attribut de cette table,
- θ est l'un des opérateurs $=$, $\neq$, $<$, $>$, $\leq$ ou $\geq$,
- v_1, v_2 sont des valeurs constantes ne contenant pas une spécification d'une table (nom ou alias d'une table).

Plusieurs cas peuvent conduire à proposer des index "inutiles", c'est–à–dire qui ne seront pas utilisés pour répondre aux requêtes précédentes.

1. Un prédicat de la forme $E(t.a) = v_1$ où $E(t.a)$ est une expression en $t.a$, par exemple $t.a^2+3t.a$. Dans un tel cas, $t.a$ n'est pas l'opérande immédiat de l'opérateur de comparaison. Le SGBD n'utilise pas d'index sur l'attribut a pour chercher les tuples vérifiant ce prédicat.
2. Un prédicat de la forme $t.a \neq v_1$ n'utilise pas un index construit sur a car toutes les données sont parcourues sauf v_1 si elle existe dans la table.
3. Un prédicat de comparaison de chaînes de caractères utilisant `like` n'exploite pas l'existence d'un index pour chercher les n–uplets vérifiant ce prédicat si la recherche s'effectue sur une sous–chaîne (par exemple, dans le cas d'utilisation d'un joker au début `like '%chaîne'`) plutôt que sur la totalité de la chaîne.

3.2 Détermination des techniques d'indexation

Dans une base de données, les attributs des clauses `<attribut>=<constante> | <sous-requête>`, `<attribut> in <liste_de_valeurs> | <sous--requête>` donnent lieu à la création d'index de hachage. Les attributs des clauses restantes et les attributs de jointure donnent lieu à la création d'index structurés en b–arbres ou leurs variantes (b^+–arbre, b^*–arbre). La raison est qu'un index de hachage est plus rapide qu'un index en b–arbre pour la recherche d'un enregistrement. Cependant, l'utilisation d'un index de hachage statique sur une table qui ne l'est pas (la fréquence des insertions et des suppressions est importante) peut dégrader substantiellement les performances. Pour pallier ce problème, l'analyse des différentes requêtes du contexte d'extraction peut nous indiquer les tables qui ne sont pas statiques.

Par ailleurs, plusieurs techniques d'indexation sont utilisées dans les entrepôts : index en b–arbre, index *bitmap*, index de jointure et index de projection [VG99]. Dans ce cas, plusieurs facteurs servent à la détermination de la technique d'indexation la mieux adaptée. Ces facteurs sont liés aux caractéristiques des attributs indexés et à leur usage dans une requête SQL.

Les facteurs liés aux caractéristiques des attributs indexés sont :

- la cardinalité d'un attribut (le nombre de valeurs distinctes de cet attribut) ;

– la distribution d'un attribut, qui détermine la fréquence des occurrences des valeurs distinctes de cet attribut ;
– la portée (*value range*) d'un attribut, qui est la différence entre les valeurs maximum et minimum d'un attribut.

Par exemple, pour un attribut de grande cardinalité et de petite portée, une technique d'indexation basée sur un *bitmap* est souhaitable (un b–arbre dégraderait les performance).

De plus, l'emploi d'un index candidat dans une clause d'une requête (jointures, prédicats de restriction, GROUP BY, etc.) peut aider à déterminer le choix de la technique d'indexation à adopter. Par exemple, il est intéressant de créer un index de jointure sur deux attributs fréquemment utilisés ensemble dans une condition de jointure.

3.3 Construction du contexte d'extraction

A partir des attributs extraits dans l'étape précédente, nous construisons une matrice "requêtes-attributs" qui a pour lignes les requêtes de la charge et pour colonnes les attributs à indexer. Cette matrice est le résultat de la combinaison de deux matrices : la matrice M_{select} (construite à partir de $Part_{select}$), dont les lignes représentent seulement les requêtes d'interrogation se trouvant dans la charge, et la matrice M_{update} (construite à partir de $Part_{update}$), dont les lignes représentent les requêtes de mise à jour. La partie $Part_{update}$ est divisée à son tour en deux groupes : G_1 et G_2. G_1 contient les attributs mis à jour et G_2 contient les attributs servant à la mise à jour (par exemple, les attributs de la clause WHERE d'un UPDATE). Par conséquence, la matrice M_{update} est partitionnée verticalement en deux parties : un groupe de colonnes correspondant aux attributs de G_1 et un deuxième groupe correspondant aux attributs de G_2. Ces matrices sont construites de telle façon à garder un lien entre chaque attribut candidat et la requête de la charge où cet attribut est présent. Nous illustrons la construction de ces matrices à travers cet exemple.

Soit une base de données composée de trois tables : `Item (catalog-number, name, price, supplier-code, delivery-time)`, `Warehouse (warehouse-code, warehouse-name, warehouse-manager)`, `Item-Warehouse (catalog-number, warehouse-code, quantity, emergency-quantity)`. La Figure 2 représente un extrait de la charge composé de sept requêtes sur ces tables. Les Figures 3 et 4 montrent les matrices M_{select} et M_{update} respectivement obtenues.

M_{select} peut être diminuée des attributs de $Part_{update}$ qui sont très fréquemment mis à jour et servent rarement à la recherche (par exemple, dans une clause WHERE d'un SELECT ou d'un UPDATE). Un ratio entre la fréquence d'utilisation d'un attribut dans la partie "attributs mis à jour" de la matrice M_{update} et la fréquence du même attribut dans les matrices M_{update} (la partie "attributs de recherche") et M_{select} permet d'élaguer cet attribut. Ce ratio est comparé à un seuil défini par l'administrateur. Il est également possible de prévoir une série de tests pour estimer empiriquement ce seuil.

La matrice "requête–attributs" obtenue par la combinaison des matrices M_{select} et M_{update} représente le contexte d'extraction des connaissances pour recommander un ensemble d'index. Nous pensons que l'utilité d'un index est fortement corrélée avec

```
(1)     Select name, price from Item
        where catalog-number like '999%'
        and price > 1000 order by name
(2)     Select * from Item
        where delivery-time > 100 or price > 100
(3)     Select Item.name from Item, Item-Warehouse
        where Item-Warehouse.catalog-number = Item.catalog-number
        and Item-Warehouse.quantity > 10000
        and Item.supplier-code = 4 and Item.price > 1000
(4)     Select Item.name, Warehouse.name, Item-Warehouse.quantity,
        Item.prace from Item, Warehouse, Item-Warehouse
        where Item.catalog-number = Item-Warehouse.catalog-number
        and Item-Warehouse.warehouse-code =
        Warehouse.warehouse-code
        and Item.price>1000 and Warehouse.Warehouse-code like '%12'
        and Item-Warehouse.quantity<1000
(5)     Update Item  set delivery-time = delivery-time - 1
(6)     Update
        Item-Warehouse set quantity = 50
        where catalog-number = 10
(7)     Delete from Item where delivery-time = 0
```

FIG. 2 – Exemple de requêtes extraites d'une charge

la fréquence de son utilisation dans l'ensemble des requêtes de la matrice "requête–attributs". A priori, la recherche des motifs fréquents ou la classification sont des techniques appropriées pour mettre en évidence cette corrélation et faciliter le choix des index. Nos premiers travaux [ADG03a, ADG03b] se basent sur la recherche des motifs fréquents à l'aide de l'algorithme CLOSE [PBTL99] et apportent des premiers résultats encourageants. La classification peut également être intéressante [Roc03]. L'idée est de regrouper des requêtes similaires. Le but est de ne garder que les index candidats appartenant à un groupe de requêtes ayant une fréquence totale supérieure ou égale à un seuil donné. Ce seuil peut être défini par l'administrateur de la base ou empiriquement.

3.4 Ordre des attributs dans les index multi-attributs

Les index de la configuration peuvent être mono–attributs tels les index de projection ou multi–attributs tels les index de jointure. Le contexte d'extraction peut être exploité pour établir un ordre dans les attributs constituant les index multi–attributs. Un choix judicieux de cet ordre peut réduire le nombre d'index de la configuration. En effet, soit un index i_1 sur les attributs $a_1...a_n$ recommandé pour les requêtes de l'ensemble Q_1. L'index i_1 peut être éliminé s'il existe un autre index i_2 sur les attributs $a_1..a_n a'_1...a'_m$ recommandé pour les requêtes de l'ensemble $Q_2 \supset Q_1$. Si $Q_1 = Q_2$, l'index i_1 est conservé et i_2 est éliminé car il occupe plus d'espace.

Par ailleurs, soient trois index i_1 sur les attributs $a_1...a_n$, i_2 sur $a'_1...a'_m$ et i sur

Tables	Item				
Requêtes	price	name	delevery-time	catalog-number	supplier-code
(1)	1	1	0	0	0
(2)	1	0	1	0	0
(3)	1	0	0	1	1
(4)	1	0	0	1	0

Tables	Item-Warehouse			Warehouse
Requêtes	catalog-number	quantity	warehouse-code	warehouse-code
(1)	0	0	0	0
(2)	0	0	0	0
(3)	1	1	0	0
(4)	1	1	1	1

FIG. 3 – Exemple de matrice M_{select}

$a_1...a_n a'_1...a'_m$ (ou i sur les attributs $a'_1...a'_m a_1...a_n$) recommandés pour les requêtes des ensembles Q_1, Q_2 et Q, respectivement. Si $Q \subseteq Q_1 \cup Q_2$, l'index i peut être éliminé.

Pour illustrer ces idées, prenons l'exemple suivant. Soient deux requêtes : `select * from t where a=4 and b=5` et `select * from t where b=3`. Un index `(b,a)` peut être utilisé par ces deux requêtes, alors qu'un index `(a,b)` peut seulement être utilisé par la première requête. Nous pouvons donc nous contenter de créer l'index `(b,a)`. Ajouter une troisième requête `select * from t where a=3` rendrait l'index `(b,a)` inutilisé par cette requête. Dans ce cas, les index `a` et `b` peuvent être utilisés par l'ensemble des trois requêtes.

L'application de ces règles a deux conséquences intéressantes : (1) garantir qu'un index est utilisé par un plus grand nombre de requêtes ; (2) gérer plus efficacement l'espace de stockage. Le deuxième point est particulièrement intéressant dans les entrepôts de données.

3.5 Élagage d'index de la configuration

Le nombre d'index candidats obtenu est d'autant plus important que la charge en entrée est volumineuse. En pratique, la création de tous les index de la configuration peut ne pas être réalisable car le système n'autorise qu'un nombre d'index prédéfini par table. Ce nombre est différent d'un SGBD à l'autre. Par exemple, l'optimiseur d'Oracle autorise jusqu'à seize index par table depuis la version 6. Il faut donc réduire le nombre d'index à générer.

Nous pensons utiliser une des méthodes suivantes. La première consiste à mettre en œuvre un modèle de coût permettant d'estimer le coût d'utilisation d'un index. La deuxième consiste à faire appel à l'optimiseur de requêtes pour estimer ce coût. Dans les deux cas, les index les moins avantageux (ayant un coût plus élevé) dans la configuration sont éliminés. Pour cela, le coût moyen d'utilisation d'un même index dans les différentes requêtes de la charge est calculé. Les index sont ensuite groupés

	Attributs mis à jour				
Tables	Item				
Requêtes	catalog-number	name	price	supplier-code	delivery-time
(5)	0	0	0	0	1
(6)	0	0	0	0	0
(7)	1	1	1	1	1

	Attributs mis à jour	Attributs de recherche	
Tables	Item-Warehouse	Item	Item-Warehouse
Requêtes	quantity	delivery-time	catalog-number
(5)	0	0	0
(6)	1	0	1
(7)	0	1	0

FIG. 4 – Exemple de matrice M_{update}

selon leur table d'appartenance et triés suivant leur coût moyen décroissant. Les k (k est le nombre maximum d'index autorisé) meilleurs index sont conservés et les autres sont éliminés.

Dans les entrepôts de données, la sélection est réalisée sous la contrainte de l'espace de stockage disponible. Si l'utilisation de deux index différents a le même coût, l'index occupant le moins d'espace est privilégié.

4 Conclusion et perspectives

Nous avons proposé dans cet article une méthode basée sur la fouille de données pour la sélection automatique d'index dans les entrepôts de données. Nous complétons en premier lieu nos travaux antérieurs par une analyse plus fine de la charge, une méthode pour la détermination des techniques d'indexation, l'établissement d'un ordre dans les attributs des index multi–attributs et la prise en compte des spécifités du SGBD utilisé. Nous avons également apporté des adaptations pour la sélection d'index dans le contexte des entrepôts de données. Ces adaptations consistent dans le choix des attributs indexables, les facteurs déterminant les techniques d'indexation typiques des entrepôts et la contrainte sur l'espace de stockage disponible.

Actuellement, nous sommes en train de mettre en œuvre ces idées au sein d'un SGBD. D'autre part, nous envisageons dans nos travaux futurs de traiter le problème de la sélection des vues à matérialiser. En effet, la classification effectuée sur le contexte d'extraction permettrait de construire des groupes de requêtes ayant de fortes similarités. Chaque groupe de requêtes peut alors être un point de départ pour la génération des vues matérialisées.

Par ailleurs, l'exploitation de l'ensemble des motifs fréquents pourrait nous permettre de mieux cibler les index de jointure en étoile partiels à construire en fonction de la charge à traiter. Nous éviterions ainsi la constitution d'un seul index de jointure

en étoile complet qui serait très coûteux en terme d'espace.

Enfin, il sera indispensable de valider cette approche. Dans un premier temps, nous vérifierons que les performances de notre première approche sont bien améliorées par nos adaptations. Nous comparerons aussi les index que nous proposons avec ceux construits par un expert. Finalement, nous mènerons une étude comparative par rapport aux autres méthodes de sélection d'index, dans la mesure du possible.

Références

[ACN00] S. Agrawal, S. Chaudhuri, and V. R. Narasayya. Automated selection of materialized views and indexes in SQL databases. In *26th International Conference on Very Large Data Bases (VLDB 2000), Cairo, Egypt*, pages 496–505, 2000.

[ACN01] S. Agrawal, S. Chaudhuri, and V. R. Narasayya. Materialized view and index selection tool for Microsoft SQL Server 2000. *ACM SIGMOD International Conference on Management of Data*, 2001.

[ADG03a] K. Aouiche, J. Darmont, and L. Gruenwald. Frequent itemsets mining for database auto-administration. In *7th International Database Engineering and Application Symposium (IDEAS 2003), Hong Kong, China*, pages 98–103, 2003.

[ADG03b] K. Aouiche, J. Darmont, and L. Gruenwald. Vers l'auto–administration des entrepôts de données. *Revue des Nouvelles Technologies de l'Information*, (1) :1–12, 2003.

[BMT85] R. Bonano, D. Maio, and P. Tiberio. An approximation algorithm for secondary index selection in realtional database physical design. *The Computer Journal*, 28(4) :398–405, 1985.

[BP90] E. Barcucci and O. Pinzani. Optimal selection of secondary indexes. *IEEE Transactions on Software Engineering*, 16(1) :32–38, 1990.

[CBC93a] S. Choenni, H. M. Blanken, and T. Chang. Index selection in relational databases. In *5th International Conference on Computing and Information (ICCI 1993), Ontario, Canada*, pages 491–496, 1993.

[CBC93b] S. Choenni, H. M. Blanken, and T. Chang. On the selection of secondary indices in relational databases. *Data and Knowledge Engineering*, 11(3) :207–238, 1993.

[CN97] S. Chaudhuri and V. R. Narasayya. An efficient cost-driven index selection tool for microsoft SQL server. In *23rd international Conference on Very Large Data Bases (VLDB 1994), Athens, Greece*, pages 146–155, 1997.

[CN98] S. Chaudhuri and V. R. Narasayya. Autoadmin 'what-if' index analysis utility. In *ACM SIGMOD International Conference on Management of Data, Seattle, USA*, pages 367–378, 1998.

[Com78] D. Comer. The difficulty of optimum index selection. *ACM Transactions on Database Systems*, 3(4) :440–445, 1978.

Application aux images hyperspectrales d'une nouvelle méthode de sélection d'attributs pour la classification d'objets complexes

Alexandre Blansché, Pierre Gançarski

LSIIT, UMR 7005 CNRS-ULP
Parc d'Innovation, Boulevard Sébastien Brant
67412 ILLKIRCH
{blansche,gancarski}@lsiit.u-strasbg.fr
http ://lsiit.u-strasbg.fr/afd/

Résumé. Une nouvelle méthode de sélection d'attributs non supervisée qui va déterminer une pondération sur les attributs, différente mais optimale pour chacune des classes cherchées, a été proposée au cours de nos travaux. Cette recherche se fait par coévolution coopérative. Il a pour cela été nécessaire de définir un nouveau critère de qualité d'une classification, car chaque classe est définie à partir d'une métrique différente.

Dans ce papier, nous allons montrer que notre méthode s'applique parfaitement à la classification d'images hyperspectrales, qui est souvent problématique et dans lequelles il est nécessaire de choisir correctement quelles bandes utiliser pour obtenir une classification pertinente. En effet, dans ce type d'images, chaque pixel représente les réponses radiométriques sur de nombreuses bandes spectrales, celles-ci pouvant être fortement corrélées, bruitées ou non pertinentes.

1 Introduction

Dans le cadre notre travail, nous nous intéressons à classification d'objets complexes [4], où chaque objet est représenté par un nombre important d'attributs, qui peuvent être de natures variées (valeurs numériques, valeurs symboliques, histogrammes, etc.). Contrairement à de nombreuses méthodes, nous pensons que lors d'une classification, l'importance d'un attribut dépend de la classe que l'on désire mettre en évidence. Notre idée de base est donc, que pour chaque classe à extraire d'un ensemble de données, il est nécessaire de définir une combinaison de pondérations sur les attributs des objets la plus optimale possible pour cette classe.

Or bien souvent, ces pondérations ne sont pas connues a priori. Le but de notre méthode est de déterminer cet ensemble de combinaisons de façon non supervisée. Pour cela, pour m classes recherchées, nous définissons m classifieurs, chacun chargé d'extraire une classe à partir d'une pondération ad'hoc des attributs. Grâce à une coévolution coopérative introduite dans notre méthode, les pondérations utilisées par les classifieurs évolueront, lors de l'apprentissage, de sorte que l'ensemble des classes extraites forment une partition de l'espace des données et que cette partition soit la meilleure possible suivant un critère de qualité.

En parallèle, nous travaillons dans le domaine de la télédétection où l'utilisation des images hyperspectrales, dans lesquelles chaque pixel représente la réponse radiométrique sur de nombreuses bandes spectrales (de l'ordre d'une centaine), est de plus en plus fréquente (la figure 1 représente la signature spectrale des principaux éléments composant une image de télédétection). Or l'analyse d'une telle image est souvent problématique car, bien que toutes les données pour chaque pixel soient de même nature, leur nombre fait qu'il existe de nombreuses corrélations entre elles, ce qui donne plus de poids aux bandes corrélées par rapport à celles qui sont indépendantes. De plus, très souvent les bandes sont bruitées ou non pertinentes. Il est alors nécessaire de bien choisir les bandes à utiliser pour procéder à l'analyse des données.

Il existe de nombreuses méthodes de sélection de bandes comme les méthodes de projections linéaires (analyse des composantes principales, analyse des composantes indépendantes, etc.) ou non linéaires (analyse des composantes curvilignes) [9]. Mais, qu'elles soient supervisées ou non,

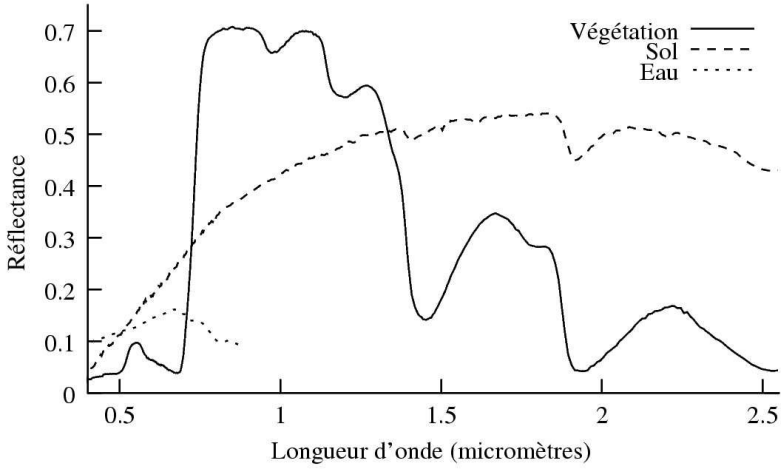


FIG. 1 – Signature spectrale typique de la végétation, d'un sol nu et de l'eau

et quelque soit la technique utilisée, elles consistent actuellement toutes à chercher à maximiser la qualité globale selon un certain critère de qualité. Elles ne se distinguent que sur le degré d'utilisation (booléen ou réel) des attributs et sur l'algorithme d'apprentissage.

En fait, nous pensons que notre méthode s'applique parfaitement et directement à la sélection (pondération) de bandes dans les images hyperspectrales. Dans la section 2 nous présenterons donc notre algorithme de sélection d'attributs dans le cas général. Dans la section 3 nous verrons comment l'appliquer à la sélection de bandes dans les images hyperspectrales. Enfin, nous verrons les perspectives pour nos travaux dans la section 4.

2 Sélection d'attributs : cas général

2.1 Méthode proposée

Nous proposons une méthode de classification utilisant une sélection automatique d'attributs, c'est-à-dire, une recherche non supervisée d'un ensemble de coefficients sur les attributs pour chaque classe cherchée. Nous utilisons plusieurs instances de classifieurs, avec des méthodes quelconques, paramétrées différement, chaque classifieur étant spécialisé dans un type d'objets, c'est à dire qu'il permettra d'extraire une classe « correcte » de l'ensemble des objets analysés. Ainsi, chaque classe sera indépendante des autres et devra être complémentaire de toutes les autres.

Il est inenvisageable de chercher ces pondérations pour chaque classe indépendamment. En effet, il est impossible d'obtenir des classes totalement disjointes à partir des mêmes données sans un apport de connaissances supplémentaires. Il faut donc que chaque classifieur puisse interagir avec les autres afin de pouvoir recouvrir l'ensemble des données.

Nous proposons donc de mettre un place un mécanisme d'apprentissage par coévolution génétique dans lequel les pondérations utilisées par les classifieurs vont « évoluer » de façon à ce que la répartition des données dans les classes soit la meilleure possible. Pour pouvoir définir la processus d'apprentissage, il est nécessaire d'éclaircir les trois points suivants :

- Comment est définie la classe extraite par un classifieur et comment est-elle trouvée à partir des objets (cf. 2.2) ?
- Quel est le critère de qualité à utiliser pour l'algorithme génétique (cf. 2.3) ?
- Comment faire évoluer les classifieurs pour que les pondérations soient les meilleures possibles (cf. 2.4) ?

Remarque Nous travaillons principalement avec l'algorithme K-means [11] et Cobweb [5]. Les coefficients sur les attributs seront utilisés pour le calcul de la distance globale (définition 2.1).

D'une manière similaire le calcul de la prédictivité pour la méthode Cobweb est effectué en utilisant les mêmes pondérations.

Définition 2.1 (Distance globale) *La distance globale entre deux objets o et o' utilisant n caratéristiques est définie par :*

$$d(o,o') = p_1 \times d_1(o,o') + p_2 \times d_2(o,o') + \ldots + p_n \times d_n(o,o')$$

où les d_i sont les distances sur chacune des n caractéristiques et les p_i sont les n pondérations.

2.2 Choix d'une classe dans un classifieur

Le choix de la classe obtenue par un classifieur peut se faire en choisissant la meilleure classe obtenue par ce classifieur. Pour cela on utilise le critère défini dans [16] (définition 2.2).

Définition 2.2 (Critère de compacité) *On définit la compacité r_k d'une classe C_k par :*

$$r_k = \begin{cases} 0, \textit{ si } \frac{1}{n_k} \sum_{l=1}^{n_k} \frac{d(x_{k,l}, g_k)}{d(x_{k,l}, g)} > 1 \\ 1 - \frac{1}{n_k} \sum_{l=1}^{n_k} \frac{d(x_{k,l}, g_k)}{d(x_{k,l}, g)}, \textit{ sinon} \end{cases}$$

avec n_k le nombre d'objets appartenant à la classe C_k, $x_{k,l}$ le l-ième élément de C_k, g_k le centre de gravité de C_k et g le centre de gravité différent de g_k le plus proche de $x_{k,l}$ (c'est donc le centre de gravité de la classe la plus proche de la classe C_k).

Plus r_k est grand, meilleure est la classe. La classe C_k telle que $r_k = \max\{r_i\}$ est choisie.

2.3 Qualité d'un classifieur

Dans le cas des méthodes supervisées, il est relativement aisé de définir la qualité d'une classification : il suffit de compter les objets placés dans la bonne classe dans l'ensemble d'apprentissage et l'ensemble de test. Mais dans le cas non supervisé, il n'est possible d'utiliser que des critères statistiques comme la compacité d'une classe et la séparabilité entre les classes. De nombreux indices ont ainsi été définis [3, 7, 8, 12]. Il existe aussi des méthodes plus complexes. Par exemple, certaines méthodes de validation consistent à tester la stabilité de l'algorithme de classification [1, 10, 15]. De telles méthodes sont utilisées pour chercher les paramètres optimaux des algorithmes, en particulier le nombre de classes.

Mais les méthodes classiques de validation s'appliquent difficilement pour nos classifieurs spécialisés, car d'une part il est impossible d'utiliser la distance entre les classes, chacune utilisant une métrique différente (les pondérations étant différentes) et d'autre part, comme chaque classe est indépendante des autres, il peut y avoir des objets non classifiés et des intersection entre les classes, alors que les classes sont compactes et que chacun des résultats est stable.

Le critère de qualité de l'algorithme génétique (fonction de fitness) doit dépendre de la qualité de la répartition (cf. définition 2.3) que forment les classes obtenues. Ce critère doit tenir compte du partitionnement obtenu (cf. 2.3.1) et de l'homogénéité des tailles des classes qui la composent (cf. 2.3.2).

Définition 2.3 (Répartition) *Une répartition est un ensemble $P = \{P_1, \ldots, P_m\}$ de m parties de l'ensemble des n objets observés $E = \{o_1, \ldots, o_n\}$. Les intersections entre les P_i ne sont pas forcément vides et $\bigcup_{1 \leq i \leq m} P_i$ n'est pas forcément égal à E.*

2.3.1 Qualité de partitionnement

Nous avons défini un critère de qualité de sorte que si l'ensemble des classes obtenues forme une partition de l'ensemble des objets à classifier, la qualité est maximale, et que celle-ci baisse si des objets appartiennent à aucune classe ou à plusieurs classes (on parlera d'objets K-classifiés pour les objets appartenant à K classes). Pour cela, nous avons d'abord défini une qualité pour chaque objet (définition 2.4).

Définition 2.4 (Qualité relative à un objet) *On définit la qualité relative à un objet o_k dans une répartition P par :*

$$Q_o(o_k, P) = 1 - \left| Card\{P_i \mid o_k \in P_i, P_i \in P\} - 1 \right|$$

Ainsi, si un objet o_k appartient à un et un seul ensemble de P alors $Q_o(o_k, P) = 1$ et o_k est dit *1-classifié*, sinon $Q_o(o_k, P) \leq 0$.

Nous pouvons alors définir la qualité sur l'ensemble des objets (définition 2.5). C'est ce critère que nous utiliserons pour évaluer nos classifications.

Définition 2.5 (Qualité de partitionnement) *On définit la qualité de partitionnement d'une répartition P par :*

$$Q_P(P) = \max\left(0, \frac{1}{n}\sum_{k=1}^{n} Q_o(o_k, P)\right)$$

On voit facilement que $Q_P(P) = 1$ si et seulement si P est une partition de E. En effet, si P est une partition, $Q_o(o_k, P) = 1$ pour tous les objets o_k, donc $\sum_{k=1}^{n} Q_o(o_k, P) = n$ et $Q_P(P) = 1$. Si P n'est pas une partition, il existe un object o_k tel que $Q_o(o_k, P) < 1$, donc $\sum_{k=1}^{n} Q_o(o_k, P) < n$ et $Q_P(P) < 1$. De plus, si tous les objets appartiennent à plusieurs classes ou à aucune, la somme des qualités relatives à chacun des objets sera négative ou nulle, et donc Q_P sera nul.

Sur l'exemple de la figure 2, trois classes sont cherchées. Chaque classifieur propose une classe (fig. 2(b)). La qualité Q_o de chaque objet est calculée (fig. 2(d)) en fonction du nombre de classes qui ont extrait chacun des objets (fig. 2(c)). En utilisant la définition 2.5, on obtient une qualité de partitionnement égale à $\frac{9}{16} = 0,5625$ pour la répartition proposée.

Cependant si l'on n'utilise que ce critère de qualité, il est possible qu'une classe englobe la quasi-totalité des objets et que le résultat forme une partition parfaite, mais sémantiquement incohérente. Nous avons, donc, arbitrairement décidé que les classes obtenues doivent toutes être de tailles comparables.

2.3.2 Qualité d'homogénéité

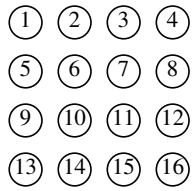
Nous avons défini ce second critère pour que les classes obtenues soient de tailles comparables et ainsi éviter qu'une classe contiennent une trop grande partie des objets. Il nous a semblé plus pertinent de définir ce critère en utilisant uniquement les objets 1-classifiés.

Définition 2.6 (Objets 1-classifiés) *On définit $E_{cl}(P) \subseteq E$, l'ensemble des objets 1-classifiés dans une répartition P, c'est-à-dire les objets qui ont été trouvés par un et un seul classifieur, par :*

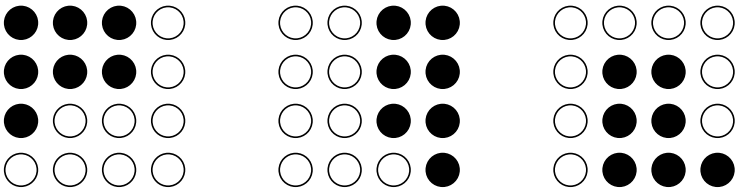
$$E_{cl}(P) = \left\{o_k \in E \mid Q_o(o_k, P) = 1\right\}$$

De plus, nous définissons nb_j comme le nombre d'objets 1-classifiés dans chaque classe de P par :

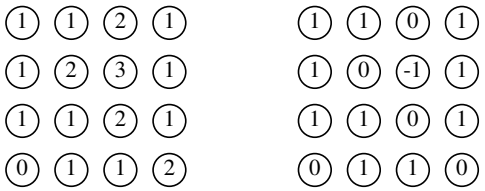
$$nb_j = Card\left\{o_k \mid o_k \in P_j \cap E_{cl}(P)\right\}$$



(a) L'ensemble de données



(b) Résultat de chaque classifieur (la classe trouvée est en noir)



(c) Le nombre de classes pour chaque objet

(d) La qualité relative à chaque objet

FIG. 2 – Calcul de la qualité de partitionnement

Définition 2.7 (Homogénéité d'une répartition) *On définit l'homogénéité d'une répartition par :*

$$Q_H(P) = \frac{\prod_{j=1}^{m} nb_j}{\left(\frac{Card(E_{cl})}{m}\right)^m}$$

Cependant, ce critère est trop restrictif et biaise les résultats. Sa maximisation permet bien qu'une classe n'englobe pas la quasi-totalité des données. Par contre, il peut engendrer une dégradation des résultats pour lequel un déséquilibre entre les tailles des classes est nécessaire ou dans le cas où une petite classe doit être mise en évidence. Il faudra donc définir un autre critère permettant, par exemple, de conserver l'homogénéité des tailles des classes, tout en acceptant des petites classes « pertinentes ». À notre avis, ce critère dépendra du domaine d'application.

2.3.3 Qualité totale d'une répartition

Il nous est maintenant possible de définir un critère de qualité d'une répartition (définition 2.8) à partir des définitions précédentes.

Définition 2.8 (Qualité totale d'une répartition) *On définit la qualité totale d'une répartition par :*

$$Q_t(P) = p_P Q_P(P) + p_H Q_H(P)$$

avec p_P, coefficient de partitionnement, *et* p_H, coefficient d'homogénéité, *tels que* $p_P + p_H = 1$. p_P *et* p_H *sont donnés par l'expert en fonction des données à traiter. Par défaut nous proposons de prendre* $p_P = 0,6$ *et* $p_H = 0,4$.

2.4 Évolution des classifieurs

Une première approche consisterait à utiliser un individu pour représenter un ensemble de classifieur, le chromosome représentant cet individu serait alors consitué de l'ensemble des poids sur chacun des attributs pour chacune des classes. Mais nous n'avons pas retenu cette méthode car le nombre de gènes sur chaque chromosome est très important et il y a donc beaucoup de risques de rester bloqué dans un maximum local. Nous pensons qu'il est plus judicieux de diviser le problème et de faire évoluer des objets plus simples.

Nous avons donc préféré une autre approche, que nous étudierons ici plus en détail. Chaque individu représente un classifieur. Un chromosome est constitué des coefficients $p_i \in [0,1]$ sur chaque caractéristique pour le classifieur qu'il représente. Le but évident est de faire évoluer plusieurs individus en collaboration afin que les classes obtenues par un ensemble d'individus forment une bonne classification de l'ensemble des données.

Notre méthode d'évolution : Nous proposons donc d'utiliser plusieurs populations, une par classe cherchée. Les répartitions se construiront en prenant un classifieur de chaque population. La méthode consiste alors à faire évoluer plusieurs populations en collaboration.

Chaque population évolue dans un environnement qui dépend des autres populations et qui donc évoluera aussi. D'où, si une population s'est adapté à son environnement à une génération g, celui-ci peut changer à la génération $g+1$ et il sera peut-être nécessaire de recommencer l'apprentissage. C'est ce que l'on appelle l'effet « Red Queen », en référence au personnage d'*Alice au pays des merveille* qui, bien qu'elle court jusqu'à en être exténuée, reste toujours au même endroit. FLOREANO et NOLFI [6] et PAREDIS [13] ont étudié l'évolution à populations multiples, mais dans le cas d'une interaction prédateur-proie, où deux populations sont en compétition et non en collaboration. POTTER et DE JONG ont étudié la coévolution coopérative [14] où plusieurs *espèces* cohabitent dans un *écosystème* et où chaque espèce occupe une *niche* de cet écosystème.

Notre méthode est donc basée sur ce type d'apprentissage par coévolution, appliqué à la classification. Comme elle est non supervisée, la niche (c'est-à-dire la classe) de chaque population n'est pas connue a priori, mais sera découverte en fonction des données qu'il reste à « occuper ».

Cependant, trouver le meilleur ensemble de classifieurs en étudiant toutes les combinaisons possibles comportant un classifieur de chaque population pose un problème important. Si l'on cherche m classes avec p individus dans chaque population, alors $m \times p$ classifications et p^m calculs de qualité sont effectués à chaque génération. Le temps de calcul devient rédhibitoire pour un nombre important de données.

Afin d'accélérer l'évaluation de la qualité des classifieurs, nous proposons de définir la qualité d'un individu par rapport à un ensemble de référence (définition 2.9).

Définition 2.9 (Ensemble de référence) *Pour une génération* g *donnée* ($g \neq 1$)*, on définit l'ensemble de référence comme le meilleur ensemble de classifieurs trouvé au cours des générations précédentes, il est noté*

$$M(g) = \{M_1(g), \ldots, M_i(g), \ldots, M_m(g)\}$$

où $M_i(g)$ *est spécialisé dans la* i*-ème classe, c'est-à-dire appartenant à la* i*-ème population.*
$M(g+1)$ *est obtenu en prenant la meilleure répartition possible à partir des* $M_i(g)$ *et du meilleur classifieur de chaque population de la génération* g.

On peut alors donner une définition de la qualité d'un classifieur (définition 2.10) illustrée par la figure 3.

Définition 2.10 (Qualité d'un classifieur) *On définit la qualité d'un classifieur* Cl_k^i*, spécialisé dans la* i*-ème classe, en calculant la qualité de la répartition obtenue en remplaçant le* i*-ème classifieur de* $M(g)$ *par* Cl_k^i*. Ainsi,* $Q\left(Cl_k^i\right) = Q_t\left(P_k^i\right)$*, où*

$$P_k^i = \{M_1(g), \ldots, Cl_k, \ldots, M_m(g)\}$$

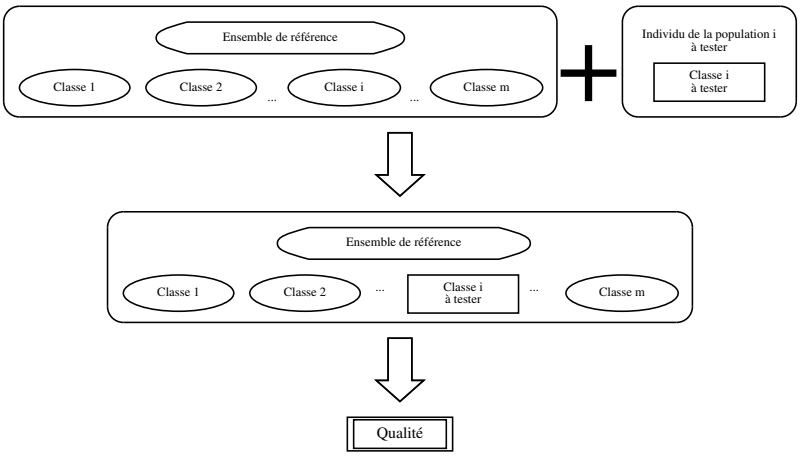


FIG. 3 – Évaluation d'un classifieur

Ainsi, seulement $m \times p + 2^m$ calculs de qualité sont effectués à chaque génération. Le gain obtenu est important. En effet, p est fréquement proche de 100 voire 200, alors que m est rarement supérieur à 15, ce qui nous fait un rapport de 1 à plusieurs milliers.

Mais cela pose un nouveau problème, qui est l'évaluation à la première génération, car $M(1)$ n'est pas défini. Pour pallier ce problème, nous pouvons tester les p^m répartitions pour la première génération ou de n'en prendre qu'un échantillon choisi au hasard. Nous avons choisi la deuxième solution, car même sur une seule génération, le temps de calcul pour tester toutes les répartitions est excessivement long.

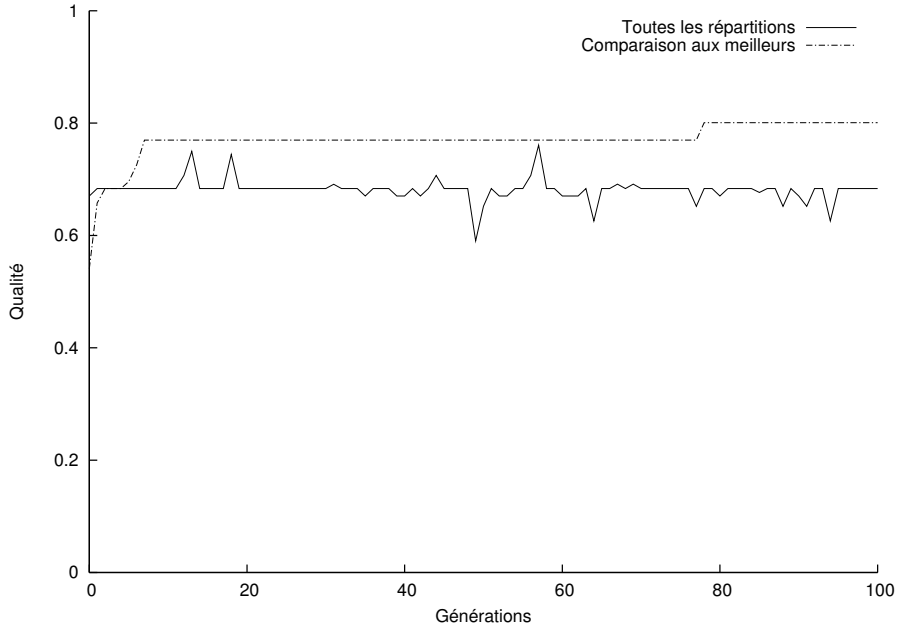


FIG. 4 – Comparaison des deux méthodes d'évaluation

Il reste enfin à savoir si, avec cette méthode d'évaluation, l'apprentissage est performant. En effet, celle-ci teste beaucoup moins de répartitions. Nous avons donc comparé les deux méthodes.

Pour cela nous avons utilisé un jeu de données très petit, car la recherche dans l'ensemble de toutes les solutions possibles est trop longue pour les données sur lesquelles nous travaillons normalement. Les données retunues ne sont pas significatives, cependant il est possible de comparer l'évolution de l'apprentissage, même s'il n'y a pas convergence vers des résultats sémantiquement corrects. La figure 4 montre l'évolution des deux méthodes sur 100 générations. L'apprentissage se fait avec trois populations de 50 individus chacune. Ainsi, en cherchant parmi toutes les solutions, 125 000 calculs de qualité sont effectués à chaque génération, alors qu'avec notre méthode, seulement 158. Il est à noter que les valeurs obtenues varient d'une exécution à l'autre, mais restent relativement similaires.

On voit qu'à la première génération, la première approche donne une meilleure répartition, avec une qualité de 0,67 contre 0,54 pour notre méthode, ce qui correspond bien à nos attentes, étant donné qu'il y a moins de répartitions évaluées avec notre algorithme.

Mais ensuite, on peut voir très clairement, que notre méthode ne nuit absolument pas aux résultats. Au contraire, l'apprentissage est même amélioré.

La meilleure qualité pour la première méthode est 0,76. Cette valeur est atteinte à la 57^{e} génération. On remarque également que la qualité oscille beaucoup au cours des générations. Ceci est dû à l'effet « Red queen ».

La meilleure qualité pour notre méthode est 0,8, atteinte à la 78^{e} génération. Ici, il y n'y a pas d'oscillation car cette méthode d'évaluation atténue l'effet « Red queen », puisque les changements d'une génération à l'autre sont moins importants. En effet, on conserve toujours les meilleurs classifieurs et l'évaluation se fait en fonction de ceux-ci, donc, tant qu'il n'y a pas eu d'amélioration, l'environnement auquel doivent s'adapter les objets ne change pas.

3 Sélection de bandes dans les images hyperspectrales

Sur la figure 5, nous avons représenté 5 bandes d'un extrait d'une image hyperspectrale d'un quartier de Strasbourg. Il s'agit d'une image DAIS de 152×156 pixels dont on a extrait 44 bandes sur 79. Ainsi, la bande 18 (FIG. 5(a)) semble être pertinente et non bruitée. Par contre, les bandes 24 et 26 (FIG. 5(b) et FIG. 5(c)) apparaissent fortement corrélées. Enfin, la bande 29 (FIG. 5(d)) semble ne porter aucune information, alors que la bande 31 (FIG. 5(e)) est visiblement très bruitée. On voit bien qu'il est nécessaire d'effectuer une sélection de ces bandes pour obtenir une classification pertinente.

(a) Bande 18 (b) Bande 24 (c) Bande 26 (d) Bande 29 (e) Bande 31

FIG. 5 – Quelques bandes spectrales de l'image

Nous avons donc appliqué notre méthode sur cet extrait. L'algorithme a été paramétré pour rechercher 5 classes. Chacune des populations est composée de 150 individus. L'apprentissage s'est déroulé sur 50 générations.

3.1 Résultats

La méthode proposée à été validée une première fois dans le cas de la classification de régions obtenues par une segmentation d'une image [2]. Il est possible de caractériser ces régions par de nombreux critères (texture, forme, etc.). Notre méthode s'est avérée très adaptée pour la sélection des caractéristiques les plus pertinentes, malgré une méthode de segmentation relativement basique.

Sur la figure 6, nous pouvons observer l'évolution du critère de qualité défini en 2.8. Nous remarquons que l'évolution présente 4 phases :

- de la génération 0 à 12, une forte amélioration de la qualité de la classification ;
- de la génération 13 à 28, une amélioration très lente, voire une stagnation de cette qualité ;
- de la génération 29 à 37, une nouvelle augmentation très forte de la qualité ;
- de la génération 38 à 50, à nouveau une augmentation très lente.

Sur la figure 7 sont représentés les résultats correspondant à ces différents points d'inflexions. Le résultat du i-ème classifieur représente la meilleure classe extraite par un individu de la i-ème population. Sur la table 1 est indiqué le pourcentage de pixels 0-classifiés, c'est-à-dire, les pixels qui ne sont dans aucune classe.

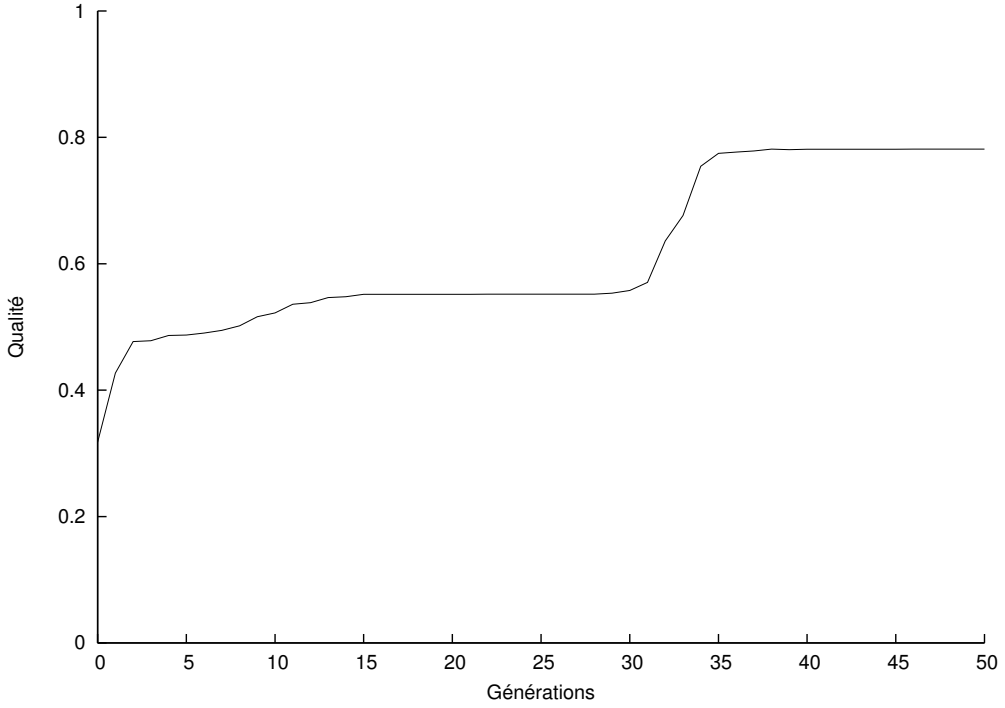


FIG. 6 – Évolution de la qualité

Génération 12	Génération 22	Génération 29	Génération 31	Génération 46
75,67%	72,24%	70,13%	58,78%	26,05%

TAB. 1 – Pourcentage de pixels 0-classifiés

Nous expliquons cette évolution de la manière suivante :

- dans la première période (de 0 à 12), l'amélioration provient principalement de la spécialisation de chacune des populations de classifieurs ;
- dans la seconde période (13 à 28), les spécialisations trouvées dans l'étape précédente n'ont plus permis d'évolution significative de la qualité, ceci étant vraisemblablement dû à la présence d'un maximum local ;
- à la 29^{e} génération, le 3^{e} classifieur vient de se spécialiser dans une classe radicalement nouvelle par rapport à la génération précédente, mais aussi, et surtout, par rapport aux autres classifieurs de la génération 29. Ceci explique la première inflexion de la courbe après cette phase de stagnation. En effet cette nouvelle spécialisation a permis de classifier des objets qui ne l'étaient pas encore (baisse du nombre d'objets 0-classifiés de 72,24% à 70,13% en une génération) ;

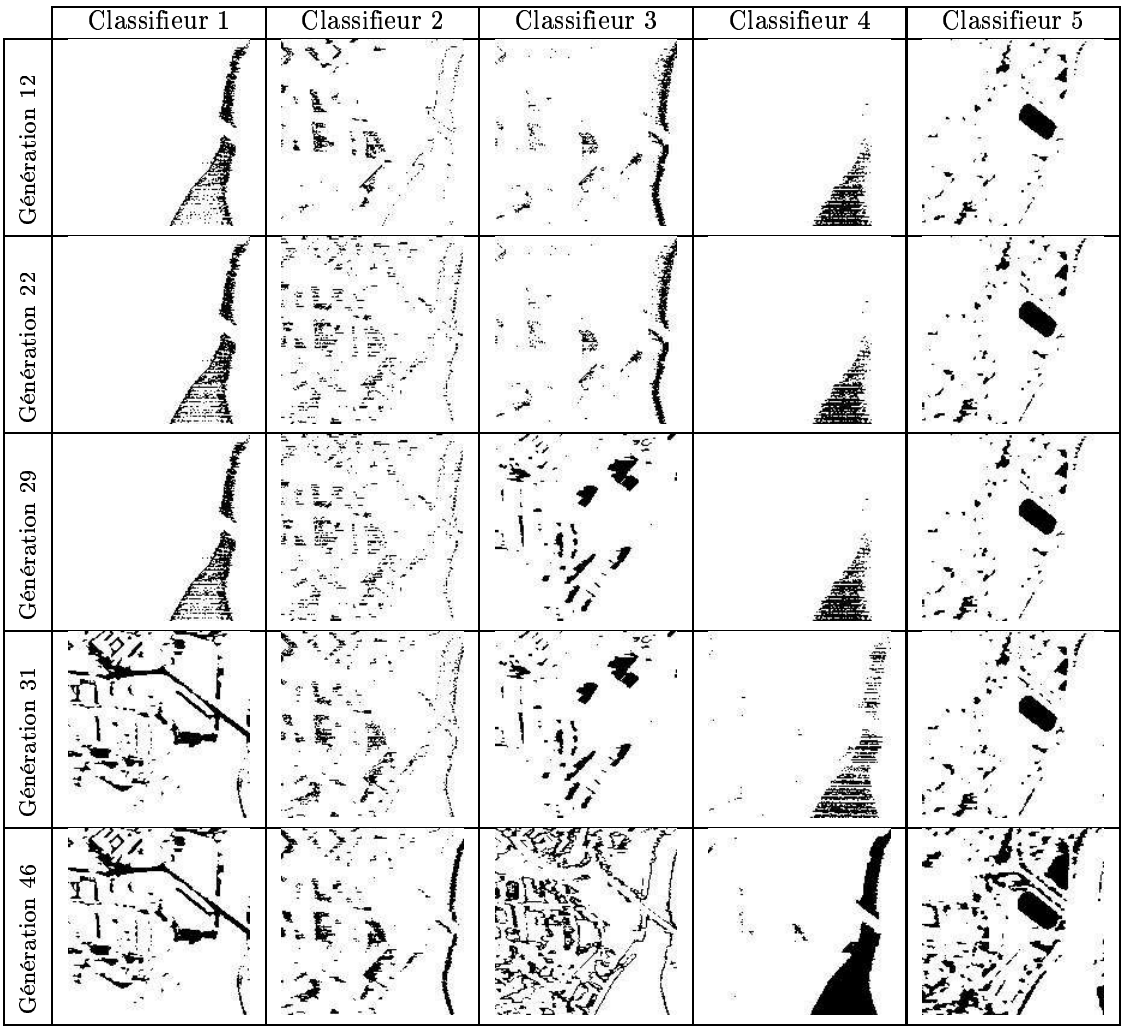


FIG. 7 – Résultats de la classification

- cette amélioration est immédiatement suivie à la 31^{e} génération par une nouvelle amélioration encore plus forte due au fait que le 1er classifieur propose lui aussi une classe radicalement différente. Cette spécialisation a de plus supprimé un conflit entre le classifieur 1 et le classifieur 4 ;
- Les classifieurs ont conservé ces nouvelles spécialisations jusqu'à la fin de l'apprentissage qui n'a plus consisté alors qu'à affiner les classes proposées. Par exemple le 4^{e} classifieur a pu regrouper l'ensemble des pixels que nous savons être de l'eau.

D'une façon plus générale, nous observons à la 46^{e} génération que des classes intéressantes ont été découvertes. Ainsi, le 1er classifieur a bien « identifié » les routes, le 2^{e} les ombres, le 4^{e} l'eau et le 5^{e} la végétation (un stade et et des espaces verts). Le 3^{e} a mis en évidence les pixels de bordures, difficiles à classifier.

Les objets non classifiés sont des objets difficiles à mettre en évidence car ils sont peu présents dans la scène.

Nous avons également testé si les pondérations obtenues pour chaque classifieur étaient identiques à chaque exécution de l'algorithme. L'algorithme converge chaque fois vers la même classification globale (même si deux classes identiques d'une exécution à l'autre ne sont pas obtenues par la même population), mais, contrairement à nos attentes, très peu de similitudes ont été observées quant aux pondérations des attributs. Cela est principalement dû aux dépendances entre les caractéristiques qui produisent des redondances d'informations. Il est alors possible d'avoir plusieurs distances très différentes qui produisent des classes identiques. Il est également possible qu'une

caractéristique ne soit que très peu significative pour un type d'objets, et donc que quelque soit le poids donné à cette caractéristique, la classification ne changera pas.

Ces résultats sont très prometteurs et montre le bien-fondé de notre méthode. En effet, même s'il y a encore des erreurs de classification, notre méthode a pu découvrir les classes importantes de manière totalement non supervisée.

Nous sommes en train de tester quelle est la performance de notre méthode par rapport à une méthode n'utilisant pas de sélection d'attributs. Les premières comparaisons, purement statistiques semblent prometteuses, mais nous proposons surtout de faire valider notre approche par des experts du domaine.

3.2 Limites

Outre les limites inhérentes aux algorithmes évolutifs, et plus particulièrement aux méthodes par coévolution, la méthode que nous proposons présente quelques faiblesses auxquelles il sera nécessaire de remédier.

Le risque majeur de notre méthode est de converger vers une partition correcte, mais sémantiquement incohérente, ce qui est malheureusement le lot commun des méthodes non supervisées. Il est en effet possible d'obtenir des ensembles qui regroupent des objets de natures très différentes mais qui forment une bonne partition de l'ensemble des données.

Un autre point faible est qu'il est nécessaire de connaître a priori le nombre de classes à trouver dans l'image. Ce problème est commun à de nombreux algorithmes de classification non supervisée, comme K-means qui est pourtant une des méthodes les plus utilisées. Certains procédés de validation permettent de tester si le nombre de classes obtenues est optimal, mais cela nécessite de relancer l'algorithme plusieurs fois avec un nombre de classes cherchées différent à chaque fois. La méthode que nous sommes en train d'implanter pour pallier ce problème consiste à ajouter ou supprimer dynamiquement des classifieurs :

- si le nombre d'objets non classifiés est trop important, un nouveau classifieur est ajouté (et donc une nouvelle population) ;
- si le nombre d'objets mis dans plusieurs classes est trop grand, le classifieur spécialisé dans la classe qui produit le plus de conflits est alors détruit.

4 Conclusion et perspectives

La classification d'objets présentant un grand nombre d'attributs (tels que des bandes radiométriques d'une image hyperspectale) pouvant être bruités, corrélés, non pertinents et éventuellement hétérogènes est un travail complexe qui empêche souvent d'avoir de bons résultats, en particulier dans le cas des méthodes non supervisées. Nous avons réussi à contourner ce problème en le divisant en plusieurs problèmes plus simples : au lieu d'essayer de classifier toutes les données par une approche monolithique, nous utilisons plusieurs classifieurs spécialisés chacun dans un type d'objets. L'apprentissage de ces classifieurs se fait en collaboration par un algorithme génétique coévolutif. Il a été nécessaire de définir un nouveau critère de qualité d'une classification pour pallier le fait que chaque classe a été obtenue indépendamment des autres. Les résultats que nous avons obtenus nous ont montré la validité de notre méthode : contrairement aux méthodes usuelles qui n'arrivent pas à utiliser les différents attributs de manière pertinente, des classes « sémantiquement correctes » ont pu être indentifiées. Ainsi une application de notre méthode à des images hyperspectrales nous a permis d'utiliser au mieux les informations radiométriques. Une validation plus poussée de nos résultats est en cours au sein du LIV[1]

Il reste cependant beaucoup à faire pour obtenir des résultats pleinement satisfaisants. En effet, notre méthode présente un risque de rester piégé dans un maximum local de la fonction de qualité et donc de converger vers une mauvaise classification. Il faudrait donc réussir à initialiser les populations avec des classifieurs permettant d'obtenir un résultat proche du maximum global. Parallèlement, une meilleure définition de ce qu'est une « bonne classification » serait également bénéfique, car cela permettrait de définir une meilleure fonction de qualité. Nous travaillons actuellement sur une approche floue qui devrait permettre d'arriver à des résultats plus satisfaisants.

[1] Laboratoire Image et Ville, ULP/CNRS UMR 7011

Références

[1] G. Bel Mufti and P. Bertrand. Validation d'une classe par rééchantillonnage. In *Cinquième rencontres de la Société Francophone de Classification, Lyon*, pages 251–254, 1997.

[2] A. Blansché. SCRITCH : un système de classification de région dans une image de télédétection basée sur des caractéristiques hétérogènes. In *RJCIA'2003*, 2003.

[3] N. Bolshakova and F. Azuaje. Cluster validation techniques for genome expression data. In *Signal processing*, volume 83, pages 825–833, 2003.

[4] A. Blansché et P. Gançarski. Sélection d'attributs et classification d'objets complexes. In *EGC 2004*, 2004.

[5] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine learning*, 2 :139–172, 1987.

[6] D. Floreano and S. Nolfi. God save the red queen ! competition in co-evolutionary robotics. In *Genetic Programming 1997 : Proceedings of the Second Annual Conference*, pages 398–406, 1997.

[7] S. Günter and H. Burke. Validation indices for graph clustering. In *Proc. 3rd IAPR- TC15 Workshop on Graph-based Representations in Pattern Recognition*, pages 229–238. J.-M. Jolion, W. Kropatsch, M. Vento, 2001.

[8] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3) :107–145, 2001.

[9] M. Lennon. *Méthodes d'analyse d'images hyperspectrales.* PhD thesis, Université de Rennes I, 2002.

[10] E. Levine and E. Domany. Resampling method for unsupervised estimation of cluster validity. *Neural Computation*, 13(11) :2573–2593, 2001.

[11] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley, CA, 1965. University of California Press.

[12] N. Nicolayannis, M. Terrenoireand, and D. Tounissoux. Pertinence d'une classification. In *Cinquième rencontres de la Société Francophone de Classification, Lyon*, pages 257–259, 1997.

[13] J. Paredis. Coevolving cellular automata : Be aware of the red queen ! In *7th Int. Conference on Genetic Algorithms (ICGA 97)*, pages 393–400, 1997.

[14] M.A. Potter and K.A. De Jong. Cooperative coevolution : an architecture for evolving coadaptative subcomponents. *Evolutionary Computation*, 8 :1–29, 2000.

[15] R. Tibshirani, G. Walther, D. Botstein, and P. Brown. Cluster validation by prediction strength. Technical report, Department of Biostatistics, Stanford University, 2001.

[16] C. Wemmert. *Classification hybride distribuée par collaboration de méthodes non supervisées.* PhD thesis, Université Louis Pasteur, 2000.

DISDAMIN: ALGORITHMES DE DATA MINING DISTRIBUES

Valérie FIOLET*,**, Bernard TOURSEL*

*LIFL, Cité Scientifique, 59655 Villeneuve d'Ascq cedex
Fiolet, Toursel@lifl.fr,
**Institut d'Informatique générale, UMH, 7000 MONS (BELGIQUE)
Valerie.Fiolet@umh.ac.be

Résumé. Nous présentons ici des algorithmes de data mining visant une exécution en environnement de type grille. Un environnement de ce type comporte des spécificités d'exécution différentes d'une exécution sur machine parallèle de type SMP, essentiellement pour ce qui concerne les aspects communications et synchronisations. Les algorithmes parallèles existants se focalisant sur des environnements de type CC-NUMA ou SMP plutôt que sur des environnements de type réseaux de stations, nous suggérons de nouvelles méthodes heuristiques de résolution des problèmes de recherche de règles d'association et de clustering sur grille.
Une solution pour la recherche de règles basée sur un découpage en profils est présentée avant une solution pour le clustering basé sur une approche dimension par dimension du problème.
Mots clefs: Data Mining, Traitement Distribué, Heuristique

1 Introduction

Les travaux présentés ici prennent place dans le projet DisDaMin (Distributed Data Mining) qui visait dans un premier temps un traitement distribué de la recherche de règles d'association, avec en perspective la proposition de méthodes hautes performances (distribuées et parallèles) pour différentes tâches du Data Mining pour des bases de données de grande taille (de données complexes ou structurées). Le traitement de bases de grande taille suppose de grandes quantités de travail d'où la nécessité de recours au parallélisme. De fait, au fur et à mesure des avancées concernant la recherche de règles d'association, des besoins et des idées se sont présentés en relation avec le clustering.

Ainsi le projet se concentre pour l'instant sur ces deux tâches (règles d'association et clustering) qui n'ont pas pour l'instant de solution acceptable en environnement distribué de type grille, en se limitant à des données classiques (mais pourrait être adapté à des données complexes en ajoutant une phase de structuration). Dans une première partie les notions de règles d'association et de clustering sont rappelées, puis les solutions proposées dans le projet DisDamin sont présentées (recherche de règles et clustering "incrémental"), avant de présenter des résultats expérimentaux.

En entrée : la base de données à traiter, le seuil de support (minsup)
En sortie : F_k : l'ensemble des k-itemsets fréquents
(1) F_1 = 1-itemsets fréquents
(2) **pour**$(k = 2; F_{k-1} \neq \emptyset; k++)$**faire**
(3) C_k = Génération des candidats de taille k à partir de F_{k-1}
(4) **pour** chaque élément c de C_k **faire**
(5) calculer le support de c.
(6) **fin pour**
(7) $F_k = \{c \in C_k | c.support \geq minsup\}$
(8) **fin pour**
(9) **retourner** $\cup F_k$

TAB. 1 – *Algorithme APriori*

2 La recherche de règles d'association

Définition et algorithmes existants

Une application bien connue de la recherche de règles d'association est "le problème de l'analyse du panier de la ménagère" qui consiste à identifier des relations entre produits qui tendent à être achetés ensemble. L'un des intérêts de la méthode est la clarté des résultats produits.

Soit un ensemble de n enregistrements (composés d'items - ou attributs) et m items distincts($\in I$), la recherche de règles d'association consiste à produire des règles entre ces items, de manière à prédire l'occurrence d'autres items. Un règle d'association est donc une implication de la forme $\mathbf{X => Y}$, où X et Y sont des ensembles d'items de I et où X $\cap$ Y= $\emptyset$ [1]. (X est la condition de la règle; Y est la conclusion ou conséquence). Nous appellerons k-itemset un ensemble de k items.

Le nombre de règles d'association est $\mathbf{O}(m2^m)$. La complexité de traitement est $\mathbf{O}(nm2^m)$. Dans l'idée de réduire cette complexité, des mesures statistiques sont associées au traitement pour réduire la taille de l'espace de recherche, en particulier le support (nombre d'enregistrements contenant chaque item). La recherche de règles d'association est alors limitée aux règles de supports et confidences supérieurs aux seuils fixés par l'utilisateur. La recherche des itemsets de support suffisant (itemsets fréquents, à partir desquels seront générées les règles) est la phase la plus coûteuse de la méthode. L'algorithme de référence pour leur génération est l'algorithme Apriori (voir Table 1) proposé par Agrawal et Srikant ([2], [16]). Le principe de APriori est de générer les 1-itemsets fréquents, puis à partir de ceux-ci, les 2-itemsets fréquents et ainsi de suite.

La méthode se base sur la règle suivante: **"Tout sous-ensemble d'un itemset fréquent doit être fréquent"**. Ce principe est utilisé pour éliminer de nombreux itemsets candidats.

Les algorithmes parallèles

De nombreuses versions parallèles de APriori existent et peuvent être classées en

1. Formulation du problème proposée par Agrawal and al. [1] et [2]

deux sous-groupes:

- celles qui proposent une duplication des itemsets candidats (à être fréquents), rendant des synchronisations nécessaires entre chaque itération sur k pour échanger les supports locaux: Count Distribution [3][18], Parallel PARTITION [14], PDM [13];

- celles qui proposent un partionnement des itemsets candidats, rendant nécessaires des échanges d'enregistrements ainsi que des synchronisations entre chaque itération sur k: Data Distribution [3][18], Intelligent Data Distribution [10], DMA [5].

Avantages d'une version distribuée

La recherche des règles d'association est un processus coûteux en calcul (coût exponentiel par rapport au nombre d'items). Le parallélisme pourrait offrir une solution pour contrôler ce coût de calculs et permettre ainsi le traitement de bases de données plus larges, et ce de deux manières: pour stocker la base de données, pour générer les itemsets fréquents.

Les versions parallèles existantes, l'adaptation parallèle vise des machines de type CC-NUMA ou SMP, c'est-à-dire disposant à la fois d'une mémoire commune et d'un réseau d'inter connexion interne rapide (Parallel DataMiner pour IBM-SP3). Des machines de ce type sont très coûteuses. Ces méthodes utilisent de nombreuses communications et synchronisations entre chaque étape.

Une autre approche consiste à envisager l'utilisation des algorithmes sur grappes de stations (existantes et disponibles pour les calculs). Ce type d'environnement utilise des traitements parallèles, mais en distribuant les données sur la grappe. Dans ce contexte, on ne dispose pas de mémoire commune partagée et le réseau d'inter connexion est lent. Une vision distribuée du problème consiste donc à trouver de nouvelles approches dans ce paradigme distribué.

3 Le clustering

La tâche de clustering consiste à identifier des groupes de données, sans connaissance à priori. Les groupes formés doivent être tels que les données au sein d'un groupe soient les plus similaires, les données de groupes différents les moins similaires.
Il existe de nombreux algorithmes de clustering, nous nous contenterons de distinguer deux méthodes majeures pour résoudre ce problème: le clustering **agglomératif** (voir Sokal [15]) et la méthode des **k-Moyennes** (voir McQueen [12] et Forgy [8]).

Le clustering agglomératif

Le clustering agglomératif est une approche itérative consistant à considérer au départ chaque donnée comme un groupe. A chaque itération, les deux groupes les plus "proches" sont identifiés puis agglomérés (voir Table 2), et ce jusqu'à atteindre un critère de convergence qui pourra être: avoir atteint un nombre de groupes souhaités; avoir atteint une limite de distance entre groupes;...

La méthode fournit de bons résultats en terme de similarité et dissimilarité, mais possède une complexité liée au nombre de données en entrée. Ainsi, à chaque itération,

Input : les données, critère d'arrêt
Output : groupes de données
(1) considérer chaque donnée comme un groupe
(2) faire
(3) rassembler les deux groupes les plus proches
(4) jusqu'au critère d'arrêt
(5) **retourner les groupes identifiés**

TAB. 2 – *Principe du clustering agglomératif*

il faut identifier les deux groupes à agglomérer en calculant les distances des groupes deux à deux. Ces calculs doivent être refaits à chaque itération ou stockés et mis à jour partiellement. Dans un cas comme dans l'autre (calcul ou stockage), les performances de la méthode sont vite limitées quant au nombre de données traitables.

La méthode des K-Moyennes

La méthode des K-Moyennes est une approche itérative fournissant un résultat heuristique de regroupement. On initialise k centres de groupes au départ et on affecte chaque donnée au groupe dont elle est le plus proche (voir Table 3). A chaque itération on effectue une affectation des données aux groupes, puis un nouveau calcul des centres (par moyenne) jusqu'à un critère d'arrêt qui peut être: avoir atteint une convergence (aucune donnée n'a changé de groupe entre deux itérations); avoir effectué un nombre limite d'itérations; avoir atteint des critères de qualité pour les groupes formés (en rapport avec les distances les séparant);...

Input : les données, le nombre de groupes à identifier (k)
Output : les groupes de données
(1) choisir k centres initiaux
(2) faire
(3) (re)affecter chaque objet au groupe le plus proche
(4) recalculer les centres de groupes
(5) jusqu'au critère de convergence
(6) **retourner les groupes identifiés**

TAB. 3 – *Principe de l'algorithme KMeans*

L'algorithme peut fournir de bons résultats pour peu que les centres soient bien initialisés au départ (qu'ils recouvrent bien l'espace de valeurs). Un des désavantages de la méthode, est qu'il faut fixer le nombre de groupes, or n'ayant pas de connaissance préalable sur les données, il est difficile de fixer la valeur de k. Toutefois des méthodes itératives sur k permettent d'ajuster cette valeur.

Les versions parallèles

Des versions parallèles existent pour les deux méthodes présentées (méthode agglomérative et k-moyennes) sur les deux types de distribution possibles pour les données (distribution verticale: toutes les données liées à un attribut sur un même site, pour tous

les enregistrements; ou horizontale: distribution des enregistrements entre les sites).

Les deux méthodes présentées nécessitent l'accès à toutes les données. Dans les versions parallèles, on doit donc se baser sur l'existence d'une mémoire commune ou accepter de nombreuses communications entre chaque itération pour conserver une cohérence dans le traitement. Une fois encore, ces communications sont plutôt adaptées à une machine parallèle coûteuse plutôt qu'à un réseau de stations de travail ou à une grille (support d'exécution visés par le projet DisDaMin).

Avantages d'une version distribuée

Une version distribuée vise à pouvoir obtenir un gain en temps de traitement en effectuant des opérations en parallèle. La spécificité d'un déploiement sur grille nous oblige à envisager une indépendance des traitements à effectuer, et ce afin de limiter les synchronisations et donc les communications.
Cependant, tout comme pour le problème de recherche de règles d'association (voir section 2), le clustering nécessite l'accès à toutes les données (tous les enregistrements, tous les attributs).

4 La solution DisDaMin pour la recherche de règles d'association

4.1 Contexte des travaux

Les travaux du projet DisDaMin (Distributed Data Mining) visaient initialement un traitement distribué de la recherche de règles d'association. Les algorithmes parallèles de recherche de règles d'association permettent le traitement de bases de données de grande taille en terme du nombre d'enregistrements à considérer, mais restent limités quant au nombre d'items (attributs) que peuvent posséder les bases. Notre proposition est de s'orienter vers une solution permettant le traitement de grandes bases de données en terme d'enregistrements mais surtout en terme du nombre d'items, la limitation actuelle de traitement se situant à ce niveau.

Partant de la constatation qu'un découpage vertical des données (séparation des attributs entre les sites) ne permettrait pas d'évaluer tous les itemsets possibles sans de nombreuses communications, il apparaît que le choix de distribution doit être réalisé par enregistrements. Un découpage horizontal aléatoire ne permettrait quant à lui pas de traitements indépendants valables.

Se basant sur les travaux de Gupta ([9]), Lent ([11]) ou encore Toivonen ([17]) dont l'idée directrice est de construire des groupes de données à partir des règles d'association générées sur de ces données, l'idée retenue dans le projet DisDamin pour la distribution des données (et donc du traitement associé) correspond à un **découpage des données en groupes** (profils ou clusters dans le domaine du data mining), puis en un traitement indépendant sur chaque fragment de données pour un regroupement final des résultats.
Un schéma de traitement de ce problème consiste donc dans un premier temps en une distribution "intelligente" par découpage horizontal de la base, basé sur des profils,

puis en traitements indépendants des fragments de base obtenus (voir Figure 1).

Gain attendu

La complexité des algorithmes de recherche de règles d'association (ou du moins de recherche d'itemsets fréquents) étant liée au nombre d'items à considérer, on souhaite de cette façon avoir un gain de complexité sur chaque fragment à considérer. La distribution se faisant sur la similarité en terme d'attributs (ou items) contenus, la complexité de traitement de chaque fragment est diminuée par rapport à un traitement global, dès les premières itérations de l'algorithme APriori, et ce par élimination des items non présents ou infréquents dans le profil considéré. En complément, on peut écarter du traitement une partie des items considérés triviaux pour le fragment considéré, diminuant de cette manière le nombre de candidats à évaluer, pour les réinsérer au final dans les résultats.
On peut ainsi obtenir un traitement optimal des fragments de base (gain de complexité de traitement mais également résultats obtenus sur les fragments intéressants en eux-mêmes).

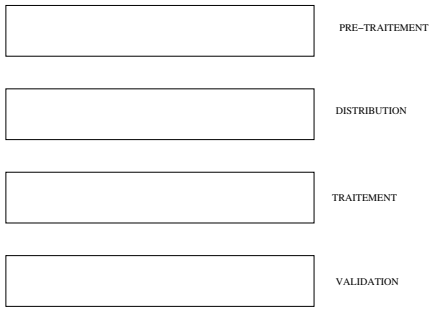


FIG. 1 – *Différentes étapes du schéma*

4.2 Les différentes étapes

Pré-traitement

Le pré-traitement consiste en un formatage des données pour le problème posé. La base visée initialement par le projet DisDaMin étant brute, il fut décider d'inclure cette phase de pré-traitement dans le schéma général (celle-ci pouvant éventuellement fournir des informations utiles pour la suite du traitement). Le pré-traitement consiste à classifier chacun des attributs, indépendamment les uns des autres, par une méthode non supervisée. Cette phase correspond donc à un clustering sur chaque attribut. Il a été choisi d'effectuer ce pré-traitement par l'algorithme K-Means de manière à ne pas être limité quant au nombre de données traitables (voir Section 3) . La base ainsi formatée contient dès lors des items, et des informations statistiques sur chacun de ces items sont d'ores et déjà connus (en particulier leurs supports).

Distribution

Le schéma complet repose sur cette étape de distribution des données qui consiste à identifier des profils d'enregistrements. Différentes pistes ont été étudiées pour effectuer cette étape (voir [6] pour des détails concernant les pistes précédemment étudiées) et les travaux ce sont finalement orientés vers la recherche d'un clustering en environnement distribué . Cette étape sera dès lors expliquer plus loin (Section 5).

Traitement effectif pour la recherche de règles d'association

A ce stade du schéma, la base est distribuée en fragments représentant des groupes d'enregistrements similaires en terme d'items contenus. Le traitement effectif peut alors être effectué de manière indépendante sur chacun des fragments en environnement distribué en utilisant un algorithme séquentiel classique tel qu'APriori (voir [6] pour des détails sur le traitement effectif).

Les étapes de communication

Les algorithmes parallèles existants nécessitent de nombreuses communications synchronisées entre chaque étape. Notre approche tend à minimiser ces communications, et en particulier à les limiter aux déplacements des données entre les sites (pour la distribution des enregistrements) et à la récupération centralisée des résultats.

Il s'avère que pour certaines données, il peut être nécessaire de ne pas les communiquer sur réseau (données confidentielles d'ordre médical, biologique ou autre). En supposant que l'on dispose au départ d'une répartition des données en multibase (découpage vertical), la phase de pré-traitement ne nécessitera pas de faire transiter les données sur le réseau, et la confidentialité sera alors assurée.

Puisque, dans le schéma, les données sont formatées lors du pré-traitement, les communications de distribution de données auront lieu sous forme "cryptée" interne au schéma et on peut dès lors considérer que ces communications ne perturberont pas la confidentialité des données.

5 Concept du clustering Incrémental

5.1 Contexte

Partant de l'idée de distribution "intelligente" des données pour le schéma général et après avoir tester différentes méthodes d'identification de groupes (voir [6]) adaptées à un environnement distribué, il s'est avéré nécessaire d'utiliser un véritable clustering. On se ramène donc pour cette phase de distribution à un problème de clustering sur les enregistrements. Or les méthodes de clustering existantes ne s'adaptent dans un contexte distribué qu'au coût de très nombreuses communications de synchronisation, non envisageables dans le contexte d'une grille. Un clustering centralisé n'est pas plus adapté puisqu'il constituerait un goulot d'étranglement pour le schéma général, avec nécessité d'accès à toutes les données (toute la base). Une deuxième étape pour le projet DisDaMin consiste ainsi, en la proposition d'un algorithme de clustering distribué.

Gardant à l'esprit le schéma général de recherche de règles, une méthode de clustering est recherchée. Le schéma de traitement complet pour le recherche de règles d'association inclut une phase de pré-traitement des données pour les adapter (les

formater) au problème posé. Ce pré-traitement consiste en un clustering de chaque attribut indépendamment (clustering unidimensionnel). Un clustering des enregistrements complets consiste en un clustering prenant en compte la totalité des attributs, donc un clustering multidimensionnel. La question est alors de savoir s'il est possible de construire un clustering multidimensionnel à partir de résultats de clustering unidimensionnels. C'est ce que nous appellerons un clustering Incrémental.

L'algorithme CLIQUE (voir [4]) consiste en un clustering de données multidimensionnelles par projection dans chaque dimension, avec identification de régions denses dans les sous-espaces visités. Partant de cette idée et du schéma général, nous proposons une méthode de clustering multidimensionnel à partir de résultats de clustering unidimensionnels (disponibles dans le schéma général de recherche de règles). Le schéma comporte trois étapes (voir Figure 2): **clustering unidimensionnel; regroupement; limitation de données**. Ces étapes sont expliquées dans les paragraphes suivants. Pour simplifier les explications, on se contentera, dans les exemples, d'un clustering sur données de dimension 2 à partir de clusterings de dimension 1.

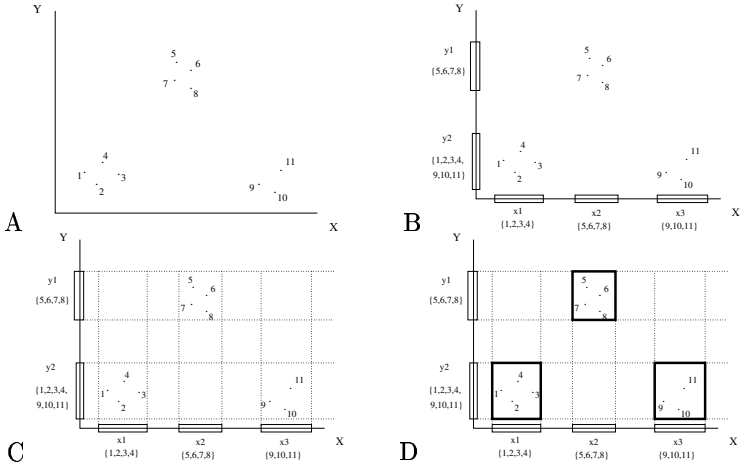


FIG. 2 – *Etapes du clustering Incrémental*

5.2 Les différentes étapes

Clustering unidimensionnel

L'étape de clustering unidimensionnel consiste simplement à traiter chacun des attributs indépendamment par un algorithme de clustering. On travaille sur des projections des données multidimensionnelles dans chaque dimension. On obtient ainsi, une identification de groupes sur chaque attribut.

Pour un traitement de données en 2 dimensions (2 attributs X et Y), disposées dans l'espace comme montré sur la Figure 2 A, le pré-traitement fournit trois groupes (x1, x2 et x3) sur l'attribut X et deux groupes (y1 et y2) sur l'attribut Y (voir Figure 2 B). En sortie du pré-traitement, on a pour chaque groupe: une valeur centre; la valeur

minimale et la valeur maximale pour le groupe; les identifiants des enregistrements associés au groupe.

A noter que cette identification de groupes dans chaque dimension (sur chaque attribut) constitue une connaissance à part entière sur les données, qui peut potentiellement être intéressante en dehors du contexte de clustering multidimensionnel.
On a donc pour l'instant le résultat de clustering sur chaque attribut. Il convient maintenant de regrouper ces résultats pour obtenir un résultat multidimensionnel.

Regroupement des dimensions

On se propose maintenant d'utiliser ces résultats de clustering sur chaque dimension pour obtenir un résultat multidimensionnel. Pour se faire, il faut effectuer un regroupement. On effectue donc un croisement des groupes formés sur chaque dimension, à partir des informations obtenues lors de la première phase (coordonnées des centres, liste des identifiants associés à chaque groupe). Les coordonnées des centres de groupe seront composées des coordonnées des centres sur chaque dimension, la liste des identifiants d'enregistrements étant obtenue par jointure des listes d'identifiants sur chaque dimension.

Sur l'exemple présenté sur la Figure 2 C, on obtient six groupes par ce croisement (croisement des trois groupes identifiés sur X avec les deux groupes identifiés sur Y). On peut dès lors éliminer les groupes vides (ceux pour lesquels la jointure des listes d'identifiants aura fourni un résultat nul). Ainsi, sur l'exemple, à partir de x1, x2, x3, y1 et y2, on obtient trois groupes en dimension 2 (voir Figure 2 D).

Clustering de limitation

On peut considérer que les groupes résultants de la phase de regroupement par croisement sont des "candidats". Plus on va considérer de dimensions, et plus on risque d'avoir un grand nombre de groupes issus du croisement. De fait, les jointures risquent de s'en trouver simplifiées lors de chaque ajout de dimensions, mais le nombre de croisements à effectuer va aller en grandissant au fur et à mesure. On aura à la fin une répartition des enregistrements en un grand nombre de groupes de très petite taille non adaptée au problème visé à l'origine (à savoir distribuer les données pour la recherche de règles d'association). Un algorithme multidimensionnel de clustering n'effectuerait sûrement pas une répartition aussi fine pour peu que l'on utilise des méthodes pour identifier le nombre de groupes le plus significatif sur l'ensemble des données.

On peut donc imaginer une étape de correction, visant à limiter ce nombre de groupes résultants. Ainsi, nous proposons de rassembler certains de ces groupes "candidats" (issus de la phase de croisement): les plus proches. On décide donc ici de faire un clustering des "candidats". Les groupes issus du croisement sont considérées comme des données d'entrée à un algorithme de clustering. Finalement les groupes identifiés pour ces deux dimensions pourront être considérés comme résultats de clustering sur dimension inférieure pour les croisements suivants.

Algorithmes de clustering utilisés

La méthode du clustering "incrémental" utilise les algorithmes de clustering clas-

siques à deux niveaux:

- pour le clustering unidimensionnel;
- pour le clustering de limitation du nombre de groupes.

Pour chacune de ces étapes on pourra utiliser au choix le clustering de type agglomératif ou le clustering de type k-moyennes. Des comparaisons des différentes combinaisons possibles sont présentées dans la section Résultats (voir section 7.3).

5.3 Améliorations

Différents scénarios

Le schéma se base sur le croisement de résultats de clustering dans des sous-espaces, qui constituent des résultats intermédiaires à la méthode. On peut imaginer plusieurs possibilités de "croisement" de ces résultats intermédiaires.

Concernant l'ordre dans lequel on va considérer ces résultats (les sous-espaces), deux modèles apparaissent incontournables: une structure en arbre (voir figure 3 A), ou une structure d'ajout un par un (voir figure 3 B).

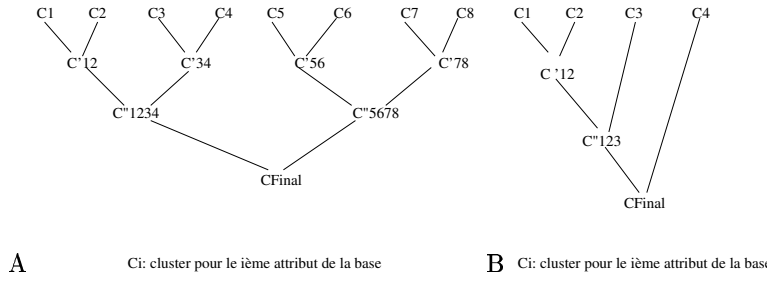


FIG. 3 – *Scenarii de croisement*

Digression macro itérative

Dans la section précédente (voir Section 5.2), on considère au départ un traitement initial unidimensionnel (voir les Xi et Yj sur la figure 2 A). On pourrait partir de résultats de clustering sur un ensemble d'attributs disponibles sur un site (clustering pluridimensionnel) plutôt que de les considérer un par un au départ (voir section 5.4). L'influence de cette possibilité sur les résultats sera décrite dans la section Résultats (voir section 7.3).

On considère également un croisement des groupes à partir de deux ensembles de groupes (à partir des Xi et Yj sur la Figure 2; à partir de C1 et C2 → C'12; C'12 et C'34 → C"1234; ou C'12 et C3 → C'123 sur la Figure 3). On pourrait envisager un croisement sur plus de deux sous-clusterings et ainsi effectuer les croisements de dimensions X, Y et Z en même temps, plutôt que de croiser X et Y puis d'en croiser le résultat avec Z.

5.4 Place dans le projet DisDamin

Déploiement sur une grille

Pour la phase de clustering, les dimensions devant être traitées indépendamment, on peut utiliser un découpage vertical des données (contexte multibase comme pour le schéma général de recherche de règles). La digression macro itérative proposée (voir section 5.3) peut dès lors être utilisée sur l'ensemble des attributs présents sur un site.

Pour le phase de croisement (que ce soit en arbre ou en considérant les résultats intermédiaires un par un), plutôt que d'imposer un ordre de croisement, on peut utiliser les résultats selon l'ordre de leurs disponibilités.

On devra bien entendu assurer au maximum la répartition de charges dans la grille en fonction du nombre de machines, de leurs puissances respectives...

Distribution des données

L'idée du clustering incrémental est apparue du fait de la nécessité d'une méthode de clustering distribué pour la distribution des données dans le schéma de recherche de règles d'association.

Dès lors, la méthode présentée a sa place dans le projet DisDaMin en tant que phase de distribution des données, d'autant que ce sont les particularités du schéma des règles d'association que les étapes de clustering incrémental sont inspirées.

La limitation à apporter sur le nombre de groupes résultants du clustering incrémental pourra dès lors être liée au nombre de machines disponibles pour le pré-traitement, mais devra également tenir compte du fait que l'on ne doit pas avoir une granularité trop fine dans le contexte de la recherche de règles.

Réutilisation des résultats du Pré-traitement

Les données initialement visées par l'application étant des données médicales (symptômes et conséquences du diabète), un formatage des données est nécessaire. En particulier les attributs de la base sont de tout type, avec une prépondérance d'attributs à valeurs réelles. La base étant totalement brute, une premier travail consiste à identifier des classes de valeurs qui seront ensuite considérés comme items dans le contexte des règles d'association. Une phase de pré-traitement consiste donc à réaliser un clustering sur chaque attribut de la base de façon à classifier les valeurs sur chaque attribut et permettre de générer des règles d'association intéressantes (pour lesquels les items n'ont pas une granularité trop fine).

Ainsi, les résultats de clustering unidimensionnel sont déjà disponibles dans le schéma général (pour le formatage des données) et ne nécessiteront donc pas de traitement supplémentaire. Ils sont déjà effectués dans un contexte de grille.

6 Expériences pour le schéma général de recherche de règles

Pour ces expériences des processus de pipeline ont été implémentés entre les différentes phases, afin d'utiliser au mieux les ressources distribuées et de recouvrir les communications autant que possible. La phase de distribution par clustering n'ayant pas encore été

insérée dans le schéma général, les résultats présentés ici sont basés sur une distribution par clustering centralisé des distances au modèle (voir [6]).

Lors du pré-traitement, un modèle d'enregistrements est identifié, la distance de chaque enregistrement à ce modèle est calculée, et la décision de distribution est prise par un clustering de ces distances.

Les tests ont été réalisés sur données réelles, une base de données médicale composée de 182 attributs continus avant pré-traitement, et de 30.000 enregistrements. Après la phase de pré-traitement, on obtient 480 items. La largeur de la base a été augmentée, mais de nombreux items sont d'ores et déjà identifiés comme infréquents et peuvent être écartés immédiatement ou dès le début du traitement effectif.

Même avec une phase de distribution non optimale (basé sur des distances à un modèle et n'assurant pas de répartition de charge), on obtient un taux de conservation des itemsets fréquents de **95%**. C'est-à-dire que 95% des itemsets générés fréquents dans une exécution normale (par un algorithme classique exact), sont identifiés dans le schéma heuristique. Compte tenu des gains de traitement offerts par la méthode distribuée (parallélisme des tâches donc gain de temps, possibilité de stockage accrue, diminution de complexité liée au nombre d'items à considérer...), ce taux de conservation est plus qu'acceptable.

On peut dès lors espérer que l'utilisation d'un mode de distribution plus adéquate (tel que le clustering incrémental présenté) apportera encore des gains de complexité, mais peut être également de qualité, les enregistrements étant répartis par une véritable méthode de clustering (sans modèle), on peut espérer que les groupes/profils identifiés seront de meilleure qualité, d'où une meilleure qualité des résultats.

7 Expériences pour le clustering incrémental

7.1 Principe des tests

Les tests ont été réalisés sur des données de 2 à 25 dimensions générées de manière à ce que des groupes existent réellement dans les données (utilisation d'une variance autour de centres de groupes), en assurant un déséquilibre de taille entre les groupes ainsi qu'une couverture de l'espace. Le nombre d'enregistrements a été limité à 1000 afin de pouvoir appliquer le clustering agglomératif, et donc de pouvoir confronter les différentes versions possibles de la méthode.

Les résultats présentés sont des moyennes réalisés sur 50 jeux de test de données de dimension 25 (et ce afin de se baser sur un nombre suffisant de dimensions pour les performances des méthodes), le but étant à terme de traiter des données de grandes dimensions (les résultats sur des données avec un nombre de dimensions inférieur sont similaires à ceux présents ci-dessous).

Enfin, les résultats sur les méthodes de clustering incrémental ont été comparés aux résultats des méthodes de clustering multidimensionnel par l'algorithme des K-Moyennes et la version agglomérative. Les comparaisons ont porté sur les temps d'exécution, les groupes formés (sont-ils comparables aux groupes initiaux connus et à ceux identifiés par un clustering multidimensionnel?) et sur une qualité de distance (critère de qualité d'un clustering).

Dans les résultats présentés, MA et MK désigneront respectivement les clusterings agglomératif et par k-moyennes multidimensionnels (versions classiques). Les UXCYZ représenteront les versions de clustering incrémental avec: X, le type de clustering unidimensionnel utilisé (A pour agglomératif, K pour k-moyennes); Y, le type de clustering de croisement utilisé (A ou K); et Z le scénario d'insertion utilisé (par structure en arbre B ou attribut par attribut O - voir Figure 3). Enfin, les MUXCYZ représenteront les versions macro itératives (traitement pluridimensionnel de départ, sur un bloc d'attributs, plutôt qu'unidimensionnel sur chaque attribut).

7.2 Mesures et Critères utilisés

Pour la qualité de distance, on prend en compte la distance moyenne d'un enregistrement au centre du groupe auquel il est affecté, permettant de juger de la compacité des groupes.

Les mesures de temps ont été réalisées de manière à identifier la durée de chacune des étapes de la méthode de clustering incrémental (voir Figures 4 A et 4 B). Ainsi, la phase de clustering unidimensionnel (Tpreat), la phase de croisement et regroupement des dimensions (Ttreat) peuvent être comparées à un algorithme multidimensionnel. Cependant, et afin de se rapprocher du schéma de règles d'association dans lequel doit s'intégrer la méthode, un temps supplémentaire (Tsupplt) a été ajouté aux versions macro itératives (cette phase correspond au pré-traitement sur chaque dimension qui devra de toute manière être réalisé, et qui correspond donc à un surcoût dans les versions macro itératives).

Pour la comparaison des groupes formés, et ayant comme référentiel les groupes initiaux (qui sont confirmés dans les tests - voir plus loin), la mesure consiste simplement à vérifier si les groupes formés sont similaires quant aux identifiants d'enregistrements qu'ils contiennent. Le but étant de savoir si la méthode de clustering incrémental permet d'arriver à une même répartition des enregistrements qu'un clustering multidimensionnel classique qui retrouve les groupes ayant servis de base à la génération des données.

Enfin, un tableau récapitulatif prend également en compte les possibilité de parallélisation des meilleures versions par rapport à ces critères.

7.3 Résultats

Groupes formés

Le premier critère à prendre en compte pour juger de l'intérêt de la méthode de clustering incrémental consiste à vérifier qu'on obtient bien les mêmes résultats qu'un algorithme classique (multidimensionnel) de clustering.

Ainsi, les tests ont permis de déterminer que les groupes initiaux (servant de base à la création des données) été ré-identifiés par un clustering agglomératif multidimensionnel (méthode exacte), de même que par un clustering k-moyennes multidimensionnel (les centres étant initialisés de manière à bien recouvrir l'espace des valeurs, voir section 3). On peut donc bien considérer l'ensemble des groupes initiaux comme référentiel pour juger de la pertinence des groupes générés par les versions de clustering incrémental.

Ainsi, pour toutes les versions utilisant un clustering agglomératif (que ce soit pour le pré-traitement ou le croisement, que ce soit avec un pré-traitement unidimensionnel ou macro itératif, UACYZ, MUACYZ, UKCAZ, MUKCAZ), les groupes générés représentent un mélange complet des groupes de référence. En particulier, les versions UACAZ fournissent les résultats les plus mauvais concernant le mélange des groupes. Les groupes générés ne peuvent en aucun cas être considérés comme similaires aux résultats corrects (ceci sera expliqué plus loin).

Pour les versions basées sur un pré-traitement par k-moyennes (UKCYZ et MUKCYZ), on retrouve à peu près les groupes initiaux, avec quelques regroupements. Ainsi, pour 10 groupes initiaux, les versions de clustering incrémental basées sur un pré-traitement par k-moyennes ne fournissent que 7 ou 8 groupes (des groupes voisins ont été rassemblés pour n'en fournir plus qu'un). On peut cependant considérer la solution comme acceptable en tant qu'heuristique, surtout si on prend en compte la possibilité de distribuer ce clustering incrémental (ce qui n'est pas possible de manière satisfaisante pour un clustering multidimensionnel classique).
La façon dont les résultats intermédiaires sont insérés dans la solution final (structure en arbre ou dimension par dimension), ne semble quant à elle pas apporter de différence.

Pourquoi une version incrémentale basée entièrement sur le clustering agglomératif fournit les moins bons résultats en terme de qualité de groupes?

La méthode de clustering agglomératif est une méthode exacte qui va à chaque étape prendre la meilleure décision de regroupement. Une fois une décision prise, elle ne pourra plus être modifiée, on ne pourra pas revenir en arrière. Ainsi, lors des décisions prises sur chaque dimension (pré-traitement unidimensionnel ou pluridimensionnel - macro itératif), la méthode prend des décisions de regroupement des données sur cette (ces) dimension(s). Ces décisions définitives sont transmises lors du croisement et influeront sur la phase de clustering de limitation de ce croisement.

Pour la version basée sur la méthode des k-moyennes, les centres sont "ajustés" jusqu'à convergence pour chaque dimension, mais les décisions effectuées pourront être ajustées lors des croisements en ajustant les centres de groupes par rapport aux données issues de plusieurs dimensions. La méthode permet ainsi de corriger des erreurs de décisions qui auraient été prises dans les dimensions, mais qui ne sont plus valables si on prend en compte plusieurs dimensions au lieu d'une seule. On a ici, la vérification que la combinaison de plusieurs méthodes "heuristiques" (telle les k-moyennes) amène de meilleurs résultats que la combinaison de méthodes exactes (qui ne laissent pas une part d'aléatoire dans les choix effectués).

Temps

La Figure 4 A permet immédiatement de constater que les versions agglomératives sont les plus coûteuse en temps. Le clustering agglomératif a une complexité liée au nombre de données à considérer (voir Section 3).

La version MA fournit des résultats corrects, mais elle se trouve être de loin plus coûteuse en temps que la version MK ou que les versions UKCYZ ou MUKCYZ (qui fournissent elles aussi des résultats corrects, voir précédemment les groupes formés).

Les versions à base de pré-traitement agglomératif UACYZ et MACYZ (qui ne

fournissaient pas de résultats corrects quant aux groupes formés) se trouvent avoir des temps d'exécution inapplicables à de grandes quantités de données et ce du fait de la phase de clustering agglomératif unidimensionnel ou "macro itératif".

Les temps d'exécution observés permettent de confirmer le choix réalisés quant aux groupes formés, à savoir écarter les versions UACYZ et MUACYZ. Ces temps viennent également confirmer la rapidité des versions par k-moyennes par rapport à la méthode de clustering agglomératif.

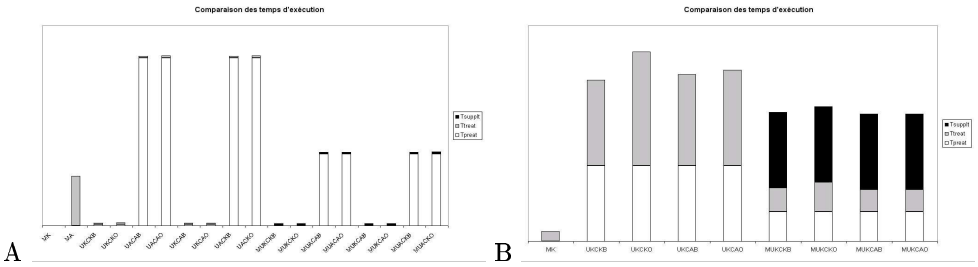


FIG. 4 – *Comparaison des temps d'exécution des différentes versions d'Incrémental clustering*

La Figure 4 B permet de comparer les temps d'exécution des versions basées sur la méthode des k-moyennes (MK, UKCYZ et MUKCYZ).

On peut remarquer que les versions de clustering incrémental (UKCYZ) ont des temps d'exécution en moyenne 15 fois supérieurs à un clustering multidimensionnel par la méthode des k-moyennes, mais il ne faut pas oublier que ces versions sont parallélisables tant au niveau de la phase de pré-traitement (clusterings unidimensionnels) qu'au niveau des croisements. De plus l'utilisation d'un clustering multidimensionnel n'est pas possible de manière satisfaisante en environnement distribué (nécessité de communications de synchronisations qui viendraient handicaper très fortement l'algorithme). Dans le contexte du projet DisDaMin pour la recherche de règles, on devrait de toute manière ajouter un surcoût de pré-traitement unidimensionnel.

Les versions macro itératives fournissent des temps d'exécution en moyenne 6 fois supérieurs au clustering multidimensionnel, mais fournissent également des possibilités de parallèlisations pour la phase de pré-traitement (multidimensionnel) ainsi que pour la phase de croisement.

A noter que pour ce qui concerne les temps d'exécution , un croisement par méthode agglomérative fournit des résultats similaires à un croisement par la méthode des k-moyennes, et ce parce que le nombre de données à traiter est limité à chaque croisement (par le clustering de limitation voir Section 5.2). On effectue donc peu de rapprochements de groupes et donc d'itérations de la méthode, tandis que la méthode des k-moyennes demande un certain nombre d'itérations pour atteindre sa convergence.

En écartant le clustering multidimensionnel (MK), puisqu'il nécessite l'accès à toutes les données (voir Section 3), on constate de façon évidente que la digression

macro itérative fournit des temps d'exécution nettement meilleurs que la version de base du clustering incrémental (basée sur des pré-traitements unidimensionnels), et que même en ajoutant le surcoût d'un pré-traitement unidimensionnel dans le schéma (zones noires du graphique 4 B qui correspondent aux clusterings unidimensionnels sur chaque attribut), on garde un temps d'exécution meilleur que les versions de base, avec des possibilités de parallélisation à plusieurs endroits (pré-traitements unidimensionnels et pluridimensionnels; croisements).

C'est donc vers ces versions macro itératives que doit s'orienter le choix de méthode de clustering incrémental pour le schéma de recherche de règles d'association.

On peut également remarquer que le croisement en structure arborescente offre une rapidité d'exécution plus importante que l'ajout des dimensions une par une. On aurait pu attendre un facteur de un demi pour la phase de croisement avec la structure en arbre par rapport à l'ajout dimension par dimension (on effectue deux fois moins de croisement). On n'obtient pas ce rapport de un demi du fait des jointures à réaliser, qui sont plus complexes (puisque sur des données plus nombreuses) dans la version en arbre.

Distance

Les mesures de distance ne permettent pas de tirer de réelles conclusions pour ces tests. En moyenne, on obtient des distances à peu près similaires pour toutes les versions. Ceci vient du fait des rapprochements de groupes effectués dans les versions de clustering incrémental à base de pré-traitement par les k-moyennes (ce qui augmente en moyenne la distance des enregistrements au centre, du fait des enregistrements éloignés de ce centre). Tandis que pour les versions basées sur un pré-traitement agglomératif, les groupes formés n'étant pas significatifs les distances sont toutes intermédiaires. En moyenne, on ne peut plus comparer les solutions par cette mesure.

7.4 Bilan sur les résultats

Le tableau 4 fournit un résumé de l'influence du choix de l'algorithme de clustering utilisé (agglomératif ou k-moyennes) pour les différentes phases (pré-traitement ou croisement) sur les résultats et les temps de traitement. Les versions utilisant un clustering agglomératif sont donc bien à écarter du fait de la perte de résultats mais également des temps de traitement obtenus.

Croisement / Unidimensionnel	Kmeans		Agglomératif	
	Avantages	Inconvénients	Avantages	Inconvénients
KMeans	Temps Unidimensionnel Temps Croisement Groupes non Mélangés	Moins de groupes	-	Temps Croisement **Groupes Perdus**
Agglomératif	Temps Croisement	Temps Unidimensionnel **Groupes Perdus**	-	Temps Croisement Temps Unidimensionnel **Groupes Perdus**

TAB. 4 – *Avantages et Inconvénients de méthodes*

En ce concentrant uniquement, pour les comparaisons, sur les versions à base de clustering par k-moyennes (pour le pré-traitement et le croisement), du fait des résultats présentés ci-dessus, le tableau 5 fournit un comparatif des qualités des méthodes et des possibilités de parallélisation qu'elles offrent.

Prenant en compte les différents aspects liés à un déploiement en grille (ceux présentés dans le tableau 5), il apparaît que la solution macro itérative est la plus adaptée au problème posé au départ. Cette solution permet la conservation des résultats (ce qui est bien entendu l'enjeu principal), mais apporte également de fortes possibilités d'amélioration des temps d'exécution par les parallèlisations offertes.

Le choix du type d'insertion des résultats intermédiaires (par arborescence ou dimension par dimension) devra être testé en environnement distribué, avec prise en compte des disponibilités des résultats intermédiaires et des communications nécessaires, et ceci afin d'évaluer si l'une des méthodes apporte un réel gain en temps d'exécution.

Version	Temps séquentiel	Qualité des groupes	Parallélisation offerte
Multidimensionnel agglomératif	- - - -	++	0
Multidimensionnel K-moyennes	+++	++	0
Incrémental (unidimensionnel)	+	++	+++
Incrémental (macro itératif)	++	++	+++

TAB. 5 – *Comparatif des aspects liés à une distribution sur grille*

8 Conclusions et Perspectives

Les premières conclusions qui ont pu être réalisées sur le schéma général de recherche de règles d'association apparaissent prometteuses, et la recherche d'un mode optimal pour la phase distribution des enregistrements devrait permettre d'en améliorer encore la qualité.

La méthode de clustering incrémental proposée en tant qu'heuristique de traitement de clustering pour données de dimensions multiples apparaît acceptable au vue de la qualité des résultats, des temps d'exécution, mais aussi des gains possibles au niveau des temps d'exécution du fait des capacités de parallélisations de la méthode. La méthode a été démontrée valable en tant qu'heuristique de clustering, mais les réels avantages sont liés à son déploiement sur grille et en particulier (but initial des travaux) dans le schéma de recherche de règles.

De nombreuses améliorations en environnement distribué sont encore possibles tant pour le schéma initial de traitement des règles d'association que pour la phase de clustering incrémental. Les travaux vont maintenant se concentrer sur l'insertion de la phase de clustering incrémental dans le schéma général, puis sur l'optimisation du déploiement distribué, afin d'assurer un maximum de recouvrement de communications.

Références

[1] R. Agrawal, T. Imielinski, and A. Swami. Database mining : A performance perspective. Special issue on learning and discovery in knowledge-based databases IEEE 5(6):914-925 (December 1993).

[2] R. Agrawal and R. Srikant. Fast algorithms for mining associations rules in large databases. In Proc. of VLDB'94, pages 478-499 (September 1994).

[3] R. Agrawal and J.C. Shafer. Parallel Mining of Association Rules. IEEE Trans. on Knowledge and Data Eng., 8(6):962-969 (December 1996).

[4] R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining application. In Proc. of the ACM Conf. on Managemnt of Data, 94-105. (1998).

[5] D.W. Cheung, J. Han, V. T. Ng, A. Fu and Y Fu. A fast distributed algorithm for mining association rules. In Proc. of the 4th Int. Conf. on Parallel and Distributed Information Systems, pages 31-42 (1996).

[6] V. Fiolet and B. Toursel. Distributed Data Mining. In Proc. of ISPDC'02, pages 349-365 (July 2002).

[7] V. Fiolet M. S. Mezmaz and B. Toursel. Incremental Distributed clustering. (2003).

[8] E. Forgy. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. Biometrics, 21:768, (1965).

[9] G. K. Gupta, A. Strehl, and J. Ghosh. Distance based clustering of association rules In Proc. of ANNIE 1999), vol. 9, pp. 759-764, ASME Press (November 1999).

[10] E-H. Han and G. Karypis. Scalable Parallel Data Mining for Associations Rules IEEE Trans. on Knowledge and Data Eng. , 2(3):337-352 (May-June 2000).

[11] B. Lent, A. Swami and J. Widom. clustering Association Rules In Proc. of ICDE'97, pages 220-231 (April 1997)

[12] J. McQueen. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297, (1967).

[13] J. S. Park, M.-S. Chen, and P. S. Yu. Efficient parallel data mining for association rules. In Proc. of the 4th Int. Conf. on Information and Knowledge Management, pages 31-36 (1995).

[14] A. Savasere, E. Omiecinski and S. Navathe. An efficient algorithm for mining association rules in large databases. In Proc. of VLDB'95, pages 432-444 (September 1995)

[15] R. Sokal and P. Sneath. Numerical Taxonomy. San Francisco: Freeman, 359. pp., (1963).

[16] R. Srikant. Fast algorithms for mining association rules and sequential patterns. PhD thesis, University of Wisconsin (1996).

[17] H. Toivonen, M. Klemettinen, P.Ronkainen, K. Hatonen and H. Mannila. Pruning and Grouping Discovered Association Rules. In ECML-95 Workshop on Statistics, Machine Learning and Knowledge Discovery in Databases. pages 47-52, Heraklion, Greece, (April 95).

[18] Zaki. Parallel and Distributed Association Mining : A survey. (1999).

Summary

Data Mining algorithms in a distributed environment such as grid are introduced. This kind of environment admits specifities, essentially concerning communication and synchronization. Existing parallel algorithms are more adapted to costly parallel machines. Heuritic methods for association rules generation and clustering problems in distributed environment are suggested. First a solution for association rules laying on a split of data into profils is presented, before a clustering method leying on underdimensionnal clustering results.

Key Words: Data Mining, Distributed computings, Heuristic methods

Pertinence des métriques fractionnaires pour l'analyse des données de grande dimension (signatures génomiques)

Sylvain Lespinats*, Patrick Deschavanne*
Alain Giron*, Bernard Fertil*

* INSERM U494
91 boulevard de l'hôpital
75634 PARIS
lespinats@imed.jussieu.fr
deschavanne@imed.jussieu.fr
giron@imed.jussieu.fr
fertil@imed.jussieu.fr

http://genstyle.imed.jussieu.fr/

Résumé. De nombreuses propriétés des espaces vectoriels de dimension 1, 2 ou 3 ne sont pas généralisables aux espaces vectoriels de plus grande dimension. Comme c'est fréquent dans l'analyse de données, la comparaison entre les signatures génomiques se heurte à cette « malédiction de la dimension ». En se basant sur des travaux qui tentent de répondre à ce problème, nous étudions l'influence de la métrique sur la classification et la représentation de ces données. Les métriques fractionnaires ont des propriétés intéressantes dans ce contexte.

1. Introduction

Les données engendrées par l'analyse de la fréquence d'apparition des mots dans le génome (la signature génomique) sont, en général, des objets de grande dimension (≥256). Pour comparer les signatures génomiques, il convient donc de s'appuyer sur une métrique dont les caractéristiques sont adaptées à la représentation des objets dans de tels espaces. La métrique euclidienne est souvent utilisée dans ce contexte, bien que ses propriétés soient loin d'être intuitives dans les espaces de grande dimension. En particulier la notion d'orthogonalité, les considérations de voisinage, le problème de l'espace vide, la concentration des distances sont autant de facteurs qui doivent êtres reconsidérés quand on souhaite développer une méthodologie fondée sur une métrique euclidienne (Gallinari 2003). Une alternative semble possible en intervenant directement sur la métrique associée à l'espace des données.

En nous basant sur les travaux de Charu, C. Aggarwal et al. (Aggarwal et al. 2001), nous nous proposons d'utiliser des métriques fractionnaires qui paraissent avoir des propriétés intéressantes dans ce contexte.

2. Données : les signatures génomiques

On appelle « signature génomique » d'une séquence d'ADN, l'ensemble des fréquences d'apparition des mots (un mot est une suite de nucléotides, par exemple, ACTG est un mot de 4 lettres). Il existe 4^n mots différents de n nucléotides. Nous avons choisi de représenter les signatures génomiques sous forme d'images où chaque mot est représenté par un pixel : plus le pixel est foncé, plus le mot est fréquent (Fig. 1).

Grâce à la structure fractale des images de signatures génomiques, la signature pour les mots de n-1 lettres peut immédiatement être déduite de la signature pour les mots de n lettres par réduction appropriée de la résolution.

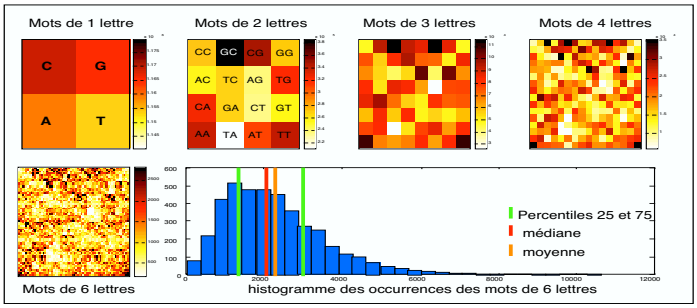


FIG. 1 : signature génomique de E. coli pour différente longueur de mots.

Il apparaît que la signature génomique est spécifique de l'espèce dont provient la séquence d'ADN. En effet, chaque espèce a sa signature propre (Fig. 2) [Deschavanne et al. 1999].

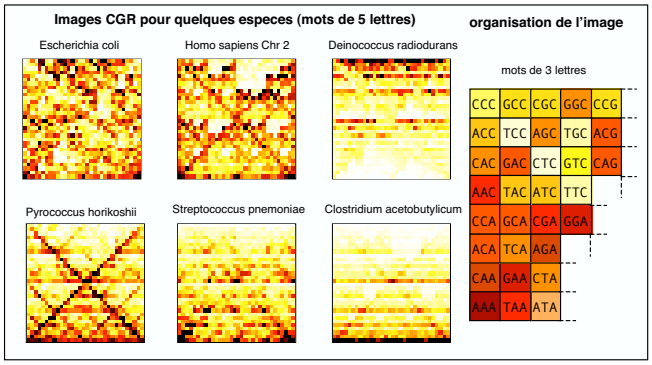


FIG. 2 : Ces 6 espèces montrent une grande diversité du point de vue de la signature génomique. La partie droite de la figure montre l'organisation des mots dans l'image.

La signature génomique d'une espèce est définie comme l'ensemble des fréquences d'apparition des mots d'une longueur donnée dans son génome. On appelle signature locale les fréquences d'apparition des mots dans un segment du génome. Au sein d'une espèce, la

plupart des signatures locales sont proches de la signature de l'espèce (Fig 3). Il existe un style d'écriture de l'ADN propre à chaque espèce qui se retrouve tout au long du génome (Deschavanne et al. 1999) (Deschavanne et al. 2000).

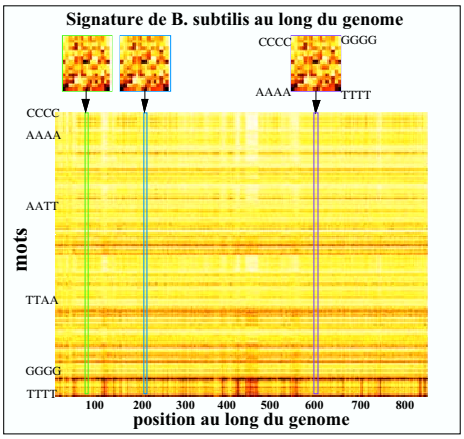


FIG. 3 : Signatures génomiques locales au long du génome de B. subtilis. Chaque colonne contient la signature locale d'un segment de 5000 nucléotides du génome de B. subtilis présenté sous forme d'un vecteur.

Plus les mots qui servent à calculer la signature génomique sont longs, plus la discrimination entre les espèces est bonne. Beaucoup d'espèces ont la même composition en bases, à peu près la même utilisation des mots de 2 lettres, mais plus on considère des mots longs, plus il apparaît des spécificités d'espèce (Deschavanne et al. 1999).

Bien sûr, le nombre de mots (et donc la dimension de l'espace des données) augmente avec la longueur des mots à considérer (64 dimensions pour les mots de 3 lettres, 256 pour les mots de 4 lettres, et 1024 dimensions pour les mots de 5 lettres).

3. Métriques fractionnaires dans les espaces de grande dimension

Soit x et y, deux points dans un espace à K dimensions, de coordonnées $x = (x_1, x_2, ..., x_k, ..., x_K)$ et $y = (y_1, y_2, ..., y_k, ..., y_K)$

Les métriques fractionnaires L_i sont définies par $d(x,y) = \left(\sum_{k=1}^{K} |x_k - y_k|^i \right)^{1/i}$.

L_2 correspond à la métrique euclidienne, et L_1 à la distance de Manhattan.

Pour des données réelles, la probabilité que l'une au moins de ses coordonnées ait une valeur extrême peut être élevée, dès lors que le nombre de coordonnées est grand. Dans le cadre de la métrique euclidienne, le point correspondant est alors rejeté aux frontières de

l'espace occupé par les données, et sa distance aux autres données est grande. Comme cette situation est partagée par un grand nombre de données, le centre de l'espace occupé par ces dernières est largement dépeuplé.

En général, on constate que la matrice des distances euclidiennes entre les données de grande dimension montre peu de contraste, c'est-à-dire que les distances sont toutes du même ordre. La distance euclidienne est donc peu apte à découvrir les ressemblances entre des points ayant beaucoup de caractéristiques en commun lorsque la dimension de l'espace est importante.

Les distances fractionnaires L_i où i est inférieur à 1 minimisent le poids des $|x_k - y_k|$ ayant des grandes contributions (Fig. 4). Des points ayant des valeurs proches sur la plupart des axes seront donc proches.

De fait, on observe une augmentation du contraste dans les matrices de distances correspondant à des métriques fractionnaires (pour i<1) (Aggarwal et al. 2001).

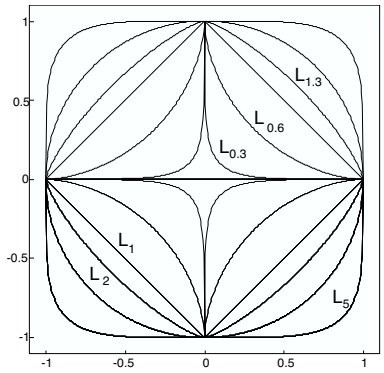


FIG. 4 : boules unité pour quelques métriques.

4. Evaluation des métriques pour la classification des signatures locales

Pour évaluer dans quelle mesure les signatures locales ressemblent à la signature de l'espèce dont elles proviennent, nous avons mis en place le protocole suivant :

Les signatures de segments de 3000 nucléotides non recouvrants ont été extraites de 43 génomes bactériens (193 segments par génome). On a considéré que la réponse était correcte lorsque le génome dont la signature était la plus proche (au sens d'une métrique choisie) était le génome d'origine de la signature locale (Lespinats et al. 2003).

On peut penser que plus la métrique est adaptée à la description de nos données, plus le pourcentage de bonne classification est élevé.

5. Evaluation des métriques pour la représentation des distances entre signatures locales

L'exploration des données de grande dimension pose un problème récurrent. Pour visualiser l'ensemble des données et leurs relations, il est nécessaire de les projeter dans un espace à 2 dimensions (voir 3 dimensions éventuellement). Dans ce but, la méthode la plus utilisée est l'analyse en composantes principales (ACP), qui cherche les axes expliquant l'inertie des données. Cette méthode n'est cependant pas complètement adaptée à notre objectif qui est d'étudier les distances entre les données. En effet, l'ACP est une projection orthogonale, donc deux points éloignés peuvent parfois se retrouver relativement proches.

C'est pour cette raison que nous proposons, pour notre problème, d'utiliser, en parallèle, l'analyse en composantes curvilignes (ACC) proposé par P. Demartines et J. Hérault (Demartines et Hérault 1997). Il s'agit d'une méthode qui a pour but de représenter les données dans un espace de plus petite dimension, en conservant au mieux les distances entre les points. Un avantage (paramétrable) est donné à la conservation des courtes distances, alors que les contraintes sur les plus grandes distances peuvent être assez relâchées [Lee et al. 2000].

L'ACC sera mis en place sur des matrices de distances calculées pour différentes métriques. La comparaison entre les distances entre signatures et les distances dans la représentation faite par l'ACC permettra d'apprécier la qualité de la représentation.

6. Résultats des classifications

Comme cela à été montré précédemment (Deschavanne et al. 1999), plus on considère des mots longs, meilleure est la qualité de classification des segments. Sur la base de nos résultats, nous pouvons ajouter que cela est vrai quelques soit la métrique considérée (à l'exception des métriques dont le paramètre prend une valeur très élevée).

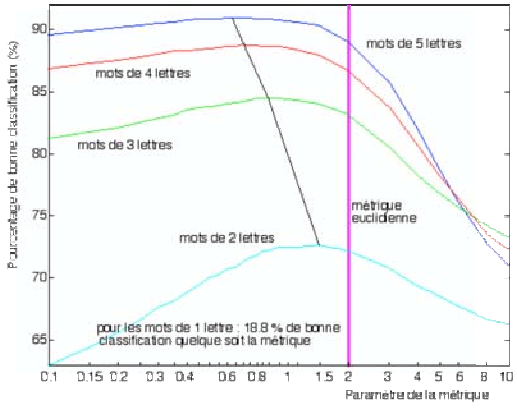


FIG. 5 : Pertinence de la métrique pour différente taille de mots. En abscisse : le paramètre de la métrique. En ordonnée, la qualité de la classification au plus proche voisin.

Il apparaît que les différentes métriques que nous avons évaluées (Fig. 5) ne sont pas équivalentes. Quelques soit la longueur des mots considérés, la métrique euclidienne ne donne pas d'excellents résultats en comparaison de certaines métriques fractionnaires. Pour une longueur de mots considérée, une existe une métrique L_i pour laquelle la qualité de classification est la meilleure.

Comme on s'y attendait, plus les mots sont longs (et donc plus la dimension de l'espace est grande), plus le paramètre de la métrique la mieux adaptée prend une petite valeur.

7. Résultats de la représentation

L'étude de la signature génomique du virus SARS (Severe Acute Respiratory Syndrome) révèle beaucoup de ressemblance avec celle des autres coronavirus, et plus particulièrement avec le « Porcine epidemic diarrhea virus » (PEDV) et le « Porcine transmissible gastroenteritis coronavirus » (PTGV). La comparaison entre segments de ces virus peut apporter des informations intéressantes sur ces organismes. Nous avons choisi d'étudier les signatures locales des virus pour des mots de 6 lettres (4096 variables, espace initial) en utilisant une fenêtre glissante de 5000 nucléotides. Le fait d'utiliser des fenêtres glissantes permet d'obtenir des signatures de segments ayant un grand nombre de mots en commun, et donc des signatures proches. Il existe donc une continuité entre les signatures locales successives d'un génome. Cette propriété est illustrée sur la figure 6 par une proximité spatiale matérialisée par un lien tracé entre les signatures consécutives.

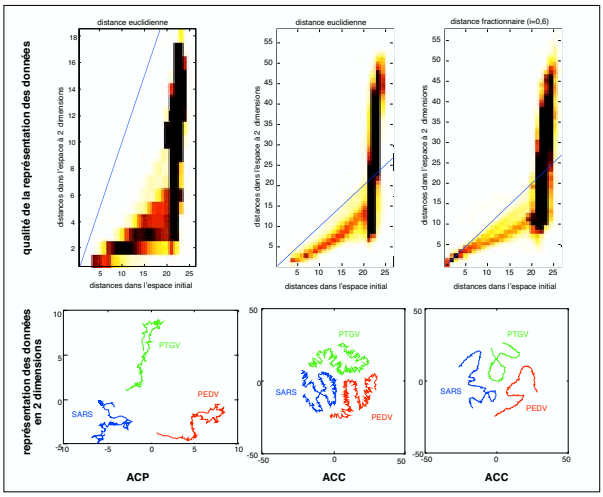


FIG. 6 : Comparaison des distances entre signatures locales dans l'espace d'origine des données et l'espace à 2 dimensions (partie supérieure de l'image), et visualisation des signatures locales dans l'espace de 2 dimensions (partie inférieure de l'image).
Colonne 1 : ACP, métrique euclidienne
Colonne 2 : ACC, métrique euclidienne
Colonne 3 : ACC, métrique fractionnaire (i=0,6)

Deux ACC ont été calculées pour ces données, l'une à partir d'une métrique euclidienne, l'autre à partir d'une métrique fractionnaire (paramètre : 0,6). Les mêmes données sont aussi représentées à l'aide de l'ACP (en utilisant la distance euclidienne).

La partie supérieure de la figure 6 exprime la conservation des distances dans la représentation en 2 dimensions. Il s'agit d'un graphique de densité où chaque point a pour abscisse la distance entre deux données dans l'espace des signatures, et en ordonnée la distance entre leurs représentants dans l'espace à 2 dimensions. Les zones sombres sont les zones où l'on trouve une forte densité de points.

L'ACP est une méthode qui fonctionne par projection orthogonale, c'est pourquoi, les distances sont plus petites dans l'espace de représentation que dans l'espace des signatures (Fig. 6, partie supérieure). Une autre conséquence de la projection orthogonale est qu'il arrive que des données loin l'une de l'autre soient projetées à proximité. Les ACCs représentent assez correctement les distances courtes, par contre, les grandes distances sont étendues pour permettre la représentation dans le plan (Fig. 6, partie supérieure). La matrice des distances pour les métriques fractionnaires a un plus grand contraste (résultats non présentés). La reconstruction par l'ACC dans un espace de petite dimension est donc facilitée ; cela explique que la qualité de la représentation soit meilleure (Fig. 6, partie supérieure).

La distance fractionnaire rapproche les données aillant beaucoup de variables communes, au contraire de la distance euclidienne. Cela explique pourquoi les séquences des virus apparaissent sous une forme relativement bruitée (ACC, métrique L_2), alors que leur représentation à partir des distances fractionnaires (ACC, métrique $L_{0,6}$) est beaucoup plus lisse (Fig. 6, partie inférieure).

Les distances courtes sont bien visualisées par ACC. On peut donc observer les ressemblances entre les signatures au sein de chacun des génomes. Malheureusement, il semble qu'un espace de deux dimensions ne permette pas de restituer fidèlement les longues distances. Il faudrait faire des représentations sur des espaces de plus grande dimension pour pouvoir entre autres observer les relations entre les signatures locales des différents génomes.

8. Conclusion

Nous ne pouvons maintenant plus ignorer la « malédiction de la dimension » dans l'étude des signatures génomiques. Le changement de métrique entraîne, à l'évidence, des répercutions importantes. Il semble que les distances fractionnaires ont des avantages non négligeables autant du point de vu de la pertinence face aux données que pour l'analyse des relations entre les objets dans les espaces de haute dimension.

References

Aggarwal, C. C., A. Hinneburg, et al. (2001) « On the Surprising Behavior of Distance Metrics in High Dimensional Space » *ICDT International Conference on Database Theory (2001)*. Article disponible sur le site http://citeseer.nj.nec.com/481093.html

Deschavanne, P. J., A. Giron, et al. (1999). "Genomic signature: characterization and classification of species assessed by chaos game representation of sequences." *Mol Biol Evol* **16**(10): 1391-9.

Deschavanne, P., A. Giron, et al. (2000). "Genomic signature is preserved in short DNA fragments." *BIBE2000 IEEE international Symposium on bio-informatics & biomedical engineering, Washington USA, 8-10 november 2000*: 161-167.

Demartines, P., J. Hérault (1997). « Curvilinear Component Analysis : A Self-Organizing Neural Network for Nonlinear mapping of Data Sets » *IEEE Transactions on neural networks, vol. 8, janury 1997* : 148-154

Gallinari, P. (2003) « Fouille de Données en Grande dimensions » *journée analyse de donnes, statistique et apprentissage pour la fouille d'images, Paris (France) 1er avril 2003*

Lee, J. A., A. Lendasse, et al (2000). « A Robust Nonlinear Projection Method » *ESANN 2000 proceeding – Eurpean Symposium on Artificial Neural Networks Bruges (Belgium), 26-28 April* 2000 : 13-20.

Lespinats, S., P. Deschavanne, et al. (2003). « L'ADN en tant que texte : style et syntaxe. » *Revue des Nouvelles Technologies de l'Information,* ***I*** :193-202.

Remerciement

Le travail présenté ici a été financé en partie par un contrat de l'action inter EPST Bio-informatique 2002-2003 (Ministère Français de la Recherche)

Summary

Genomic signatures are high dimensional objets: thousands of dimensions may be currently considered. The measure of proximity in such high dimensional spaces relies heavily on metrics. We specifically examine the properties of L_k norm (fractional distance metric) for the classification and graphic representation of genomic signatures as a function of dimensionality. The "Optimal" value for the norm parameter (k) is found dependent on data dimension.

Mots-clefs : grande dimension, métriques fractionnaires, signature génomique, classification, représentation, analyse en composantes curvilignes.

RNTI - E - 2

Visualisation avec les cartes topologiques catégorielles

Mustapha. LEBBAH[a], Fouad. BADRAN[b], Sylvie. THIRIA[a,b]
a- CEDERIC, Conservatoire National des Arts et Mtiers,
292 rue Saint Martin, 75003 Paris, France
b- Laboratoire LODYC, Université Paris 6, Tour 26-4[e]étage,
4 place Jussieu 75252 Paris cedex 05 France

Abstract

Ce papier introduit les cartes topologiques dédiée à la visualisation et la classification de données composées de variables catégorielles. Généralement, pour visualiser ces données par des cartes topologiques, les méthodes classiques utilisent une phase de codage de prétraitement de ces données en données numériques et appliquent l'algorithme classique des cartes topologiques. Dans ce papier nous proposons un modèle de cartes topologiques dédiées aux données catégorielles. Ce modèle est basé sur un formalisme probabiliste où chaque cellule est représenté par une table de probabilités. Un exemple réel est utilisé pour valider ce modèle. Les résultats obtenus montrent l'apport de ce modèle dans la visualisation et le traitement de données catégorielles.

1 Introduction

La visualisation des données est une étape importante pour la phase d'exploration d'une analyse de donnes. Cette tape est plus difficile pour les donnes qualitatives pour lesquelles il existe moins de mthodes standard. Les cartes topologiques sont de plus en plus utilisées comme outils de visualisation, puisqu'elles permettent de projeter sur des espaces discrets qui sont généralement de dimensions deux. Le modèle de base, proposé par Kohonen [8], est dédié uniquement aux données numériques. Il a été appliqué avec succès au traitement de donnés textuelles [7]. Ce modèle a été utilisé dans certaines applications afin de traiter des données catégorielles, La mise en œuvre de l'algorithme sur ces donnes suppose toujours phase prtraitement permettant d'extraire une information numrique des observations [6, 12]. Des extensions et reformulations du modèle de Kohonen ont été proposé dans la littérature : Cartes topologiques probabilistes [1] , Generative Topographic Mapping [5, 2]. En se basant sur le formalisme classique des cartes topologiques, nous avons déjà proposé un modèle de cartes topologiques dédiés aux données binaires [9]. Dans cet article, nous proposons le modèle CTM de cartes topologiques dédié aux données catégorielles. Ce modèle est basé sur le formalisme des cartes topologiques probabilistes, l'apprentissage consiste alors à estimer les paramètres du modèle en maximisant la fonction de vraisemblance des données de la base d'apprentissage. L'algorithme d'apprentissage

que nous proposons est une application de l'algorithme EM. Au paragraphe 2 nous présentons le modèle CTM et l'algorithme d'apprentissage associé. Nous présentons au 3ième paragraphe une application de CTM sur une enquête de semiologie et nous discuteons des résultats obtenus. Au dernier paragraphe nous concluons en donnant quelques perspectives.

2 Carte Topologique Catégorielle

2.1 Modèle CTM

Comme tout modèle de cartes topologiques nous supposons que l'on dispose d'une carte discrète $\mathcal{C}$ ayant N_{cell} cellules structurées par un graphe non orienté. Cette structure de graphe permet de définir une distance, $d(r,c)$ entre deux cellules r et c de $\mathcal{C}$, comme étant la longueur de la plus courte chaine permettant de relier les neurones r et c. Le modèle CTM est basé sur le formalisme probabiliste des cartes topologiques, qui associe à chaque cellule de c de $\mathcal{C}$ une probabilité $p_c(\mathbf{z})$ où $\mathbf{z}$ est un vecteur de l'espace des données. En suivant le formalisme bayésien, introduit dans Luttrel [13], la carte $\mathcal{C}$ est dupliquée en deux cartes identiques à $\mathcal{C}$: $\mathcal{C}_1$ et $\mathcal{C}_2$. Ce formalisme suppose que les données $\mathbf{z}$ sont générées de la manière suivante: on commence par choisir une cellule c_2 de $\mathcal{C}_2$ celle-ci est alors propagée à la seconde carte $\mathcal{C}_1$ en suivant la probabilité conditionnelle $p(c_1/c_2)$ et enfin $\mathbf{z}$ est générée suivant la probabilité $p(\mathbf{z}/c_1)$. Ce formalisme nous amène à définir le générateur des données $p(\mathbf{z})$ par un mélange de probabilités: $p(\mathbf{z}) = \sum_{c_2 \in \mathcal{C}_2} p(c_2) p_{c_2}(\mathbf{z})$, avec $p_{c_2}(\mathbf{z}) = \sum_{c_1 \in \mathcal{C}_1} p(c_1/c_2) p(\mathbf{z}/c_1)$, où la probabilité conditionnelle $p(c_1/c_2)$ est supposée connue. Dans la suite et afin d'introduire la notion de voisinage sur la carte, nous supposons que $p(c_1/c_2)$ est égale à $\frac{K^T(\delta(c_1,c_2))}{\sum_{r \in C_1} K^T(\delta(r,c_2))}$, où K^T est une fonction de voisinage dépendant du paramètre T et qui peut s'exprimer par $K^T(d) = K(d/T)$ où K est une fonction noyau particulière.

On suppose par la suite que les données $\mathbf{z}$ sont des vecteurs à n composantes catégorielles $\mathbf{z} = (z^1, z^2, ..., z^k, ..., z^n)$, où chaque composante $z^k \in M_k$ qui est l'ensemble fini formé par l'énumération des m_k modalités $\{x_1^k, x_2^k, ..., x_{m_k}^k\}$ de la k^{ieme} composante $\mathbf{z}$, dans ce cas $\mathbf{z} \in M_1 \times M_2 \times .... \times M_n$. Afin de simplifier ce modèle, nous supposons par la suite que les n composantes catégorielles de $\mathbf{z} = (z^1, z^2, ..., z^k, ..., z^n)$ sont indépendantes, ce qui nous permet d'écrire $p(\mathbf{z}/c_1) = \prod_{k=1}^{n} p(\mathbf{z}^k/c_1)$ où $p(\mathbf{z}^k/c_1)$ est une table de probabilité de dimension m_k. Nous associons dans la suite à chaque neurone c_1 de la carte, n tables unidimensionnelles de probabilité. Chaque table de probabilité $p(z^k/c_1)$ contient les probabilités des m_k modalités de la composante z^k. Cette table de probabilité sera notée par la suite θ^{k,c_1}: $\theta^{k,c_1} = \{\theta_j^{k,c_1}, j = 1 \ldots m_k\}$ avec $\theta_j^{k,c_1} = p(z^k = x_j^k/c_1)$. L'ensemble des paramètres permettant de définir la table de probabilité d'un neurone c_1 de la carte $\mathcal{C}_1$, est constitué de l'union de toutes les tables de probabilités des variables composantes: $\theta^{c_1} = \cup_{k=1}^{n} \theta^{k,c_1}$. Ainsi, l'ensemble des paramètres $\theta = \theta^{C_1} \cup \theta^{C_2}$ qui permettent de définir le modèle probabiliste de carte topologique est constitué par: l'ensemble des coefficients des tables: $\theta^{C_1} = \cup_{c=1}^{N_{cell}} \theta^{c_1}$, et l'ensemble des probabilités a priori: $\theta^{C_2} = \{\theta^{c_2}, c_2 = 1..N_{cell}\}$, où $\theta^{c_2} = p(c_2)$.

Le problème est donc maintenant de définir la fonction de coût et l'algorithme d'apprentissage qui permet d'estimer l'ensemble de ces paramètres.

2.2 Algorithme d'apprentissage CTM

On suppose que l'on dispose d'une base d'apprentissage $App = \{\mathbf{z}_i; i = 1..N\}$ où toute les observation $\mathbf{z}_i$ sont indépendantes. L'algogorithme d'apprentissage consiste à maximiser la vraisemblance des observations en appliquant l'algorithme EM. L'usage de l'algorithme EM s'explique par l'existence d'une variable cachée notée ξ, constituée par le couple de neurones c_1 et c_2, $\xi = (c_1, c_2)$, responsable de la génération d'une donnée observée $\mathbf{z}$. En effet, la variable cachée $\xi = (c_1, c_2)$ apparaît lorsqu'on écrit:

$$p(\mathbf{z}) = \sum_{\xi \in \Xi} p(\mathbf{z}, \xi) = \sum_{c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2} p(\mathbf{z}/c_1) p(c_1/c_2) p(c_2)$$

selon le formalisme probabiliste $p(\mathbf{z}, c_1, c_2) = p(c_2)p(c_1/c_2)p(\mathbf{z}/c_1)$

A chaque donnée réellement observée $\mathbf{z}$, il correspond une donnée catégorielle disjonctive non observée ξ qui appartient à $\mathcal{C}_1 \times \mathcal{C}_2$. Si l'on code cette variable par le codage binaire disjonctif, on obtient un vecteur binaire $\mathbf{y}$ de dimension $N_{cell} \times N_{cell}$ dont les composantes $y_{(c_1,c_2)}$ sont définies par: $y_{(c_1,c_2)} = \begin{cases} 1 & \text{si } \xi = (c_1, c_2) \\ 0 & \text{sinon} \end{cases}$. Avec cette notation, $p(\mathbf{z}, \xi)$ s'écrit:

$$p(\mathbf{z}, \xi) = \prod_{c_2 \in \mathcal{C}_2} \prod_{c_1 \in \mathcal{C}_1} \left[p(c_2) \frac{K^T(\delta(c_2, c_1))}{T_{c_2}} p(\mathbf{z}/c_1) \right]^{y_{(c_1,c_2)}}$$

On suppose qu' chaque donnée réellement observée $\mathbf{z}_i$ de l'ensemble d'apprentissage $\mathcal{A}$, il correspond une donnée non observée $\xi_i \in \mathcal{C}_1 \times \mathcal{C}_2$ qui sera codée par le vecteur $y_i \in \{0,1\}^{p \times p}$ dont toutes les composantes sont nulles sauf la composante d'indice ξ_i qui vaut 1. La vraisemblance des données s'écrit:

$$V^T(\mathcal{A}, \Xi; \theta) = \prod_{i=1}^{N} \prod_{c_2 \in \mathcal{C}_2} \prod_{c_1 \in \mathcal{C}_1} \left[\theta^{c_2} \frac{K^T(\delta(c_2, c_1))}{T_{c_2}} p(\mathbf{z}_i/c_1) \right]^{y^i_{(c_1,c_2)}}$$

Le log-vraisemblance s'écrit:

$$\ln V^T(\mathcal{A}, \Xi; \theta) = \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_2 \in C_2} \sum_{c_1 \in C_1} y^i_{(c_1,c_2)} \left[\ln(\theta^{c_2})) + \ln(\frac{K^T(\delta(c_2, c_1))}{T_{c_2}}) + \ln(p(\mathbf{z}_i/c_1)) \right].$$

L'application de l'algorithme EM pour la maximisation de la vraisemblance des données observées fournit un algorithme itératif où les paramètres à l'itération $t+1$ sont calculés en fonctions des paramètres estimés à l'itération t. Les formules calculant les paramètres à l'itération $t+1$ sont définies par:

$$p(c_2) = \theta^{c_2} = \frac{\sum_{\mathbf{z}_i \in \mathcal{A}} p(c_2/\mathbf{z}_i, \theta^t)}{N} \qquad (1)$$

$$p(z^k = x_j^k / c_1) = \theta_j^{k,c_1} = \frac{\sum_{\mathbf{z}_i \in \tau_{k,j}} p(c_1/\mathbf{z}_i, \theta^t)}{\sum_{\mathbf{z}_i \in \mathcal{A}} p(c_1/\mathbf{z}_i, \theta^t)} \quad (2)$$

Avec $p(c_1/\mathbf{z}, \theta^t) = \frac{p(\mathbf{z}/c_1,\theta^t) \sum_{c_2 \in c_2} p(c_2/\theta^t) p(c_1/c_2)}{p(\mathbf{z}/\theta^t)}$, $p(c_2/\mathbf{z}, \theta^t) = \frac{\sum_{c_1 \in c_1} p(c_2/\theta^t) p(c_1/c_2) p(\mathbf{z}/c_1,\theta^t)}{p(\mathbf{z}/\theta^t)}$ et $\tau_{k,j} = \{\mathbf{z}_i \in \mathcal{A}; z_i^k = x_j^k\}$ représente l'ensemble des individus $\mathbf{z}_i$ qui ont répondu par la modalité j à la k^{ieme} composante.

L'algorithme CTM [10] pour une paramètre T fixé se présente de la manière suivante:

Initialisation $(t = 0)$
détermination de l'ensemble des paramètres initiaux (θ^0) et du nombre d'itérations N_{iter}.

Itération de base à T constant $(t \geq 1)$
Calcul l'ensemble des paramètres θ^{t+1} à partir de l'ensemble des paramètres précédent θ^t en appliquant les formules (2) et (1).

Répéter l'itération de base jusqu'à $t > N_{iter}$.

Dans ce paragraphe, nous avons présenté l'algorithme d'apprentissage CTM permettant d'estimer les paramètres maximisant la fonction log-vraisemblance pour un paramètre T fixé. Le paramètre T permet de contrôler la taille du voisinage d'influence d'une cellule sur la carte, celle-ci décroît avec le paramètre T. Par analogie avec l'algorithme des cartes topologiques, on peut faire décroître la valeur de T entre deux valeur T_{max} et T_{min}. Pour chaque valeur de T, on obtient une fonction de vraisemblance V^T et donc l'expression varie avec T. On peut observer deux étapes dans le fonctionnement de l'algorithme.

- La première étape correspond aux grandes valeurs de T. Dans ce cas, le voisinage d'influence d'une cellule c de la carte est grand et correspond aux valeurs "significatives" de $K^T(\delta(c, r))$. Les formules 1 et 2 ont tendance, dans ce cas, à faire participer un très grand nombre d'observations à l'estimation des paramètres du modèle CTM.

- La deuxième étape correspond aux petites valeurs de T. Le nombre d'observations intervenant dans les formules (1) et (2) est alors restreint. L'adaptation est très locale.

Si on utilise cette décroissance de T, l'algorithme d'apprentissage de CTM se présente de la manière suivante :

Phase d'initialisation $(t = 0)$
Choisir T_{max}, T_{min} et N_{iter}. Effectuer l'algorithme d'apprentissage CTM pour la valeur de T constante égale à T_{max}.

Etape itérative
L'ensemble des paramètres θ^t de l'étape précédente est connu. Calculer la nouvelle valeur de T en appliquant la formule suivante:

$$T = T_{max} \left(\frac{T_{min}}{T_{max}} \right)^{\frac{t}{N_{iter}-1}}$$

Pour cette valeur du paramètre T, calculer l'itération de base.

Répéter l'étape itérative tant que $t \leq N_{iter}$

3 Exemple: Construction du Corpus de Mots représentatifs

L'exemple que nous présentons porte sur une base extraite d'une enquêtede sémiologie portnt sur le sens des mots chez différentes personnes. Cette enquête est constituée par un ensemble de mots possédant un certain nombre de qualités définies par l'auteur [11, 15]: Univocité sémantique (chaque mot ne doit posséder qu'un seul sens premier), Stabilité sémantique (le sens de chaque mot est stable dans le temps, c'est-à-dire qu'un mot ne peut pas posséder un sens pour un groupe et un autre pour les autres), Non-consensualité (les mots doivent être capables de provoquer des réactions contradictoires) et intensité émotionnelle (les mots doivent posséder une charge affective suffisante afin de pouvoir jouer le rôle de stimuli).

L'enquête consiste à demander aux individus de noter les mots en fonction du plaisir que l'ensemble de leurs connotations fait naître. Le but de l'enquête est de définir des groupes homogènes par rapport aux réponses apportées à l'enquête et de déterminer par la suite à l'aide de variables additionelles, les caractéristiques de ces groupes.

Ces données nous ont été fournies par Monsieur L. Lebart; cette enquête est issue d'interrogations effectuées par la SOFRES auprès de son panel télématique. Nous avions accès à 1128 individus représentatifs de la population française. Cette enquête portait sur 286 mots, dans le cadre de cette étude nous n'avons eu accès qu'à 70 mots. Chaque individu interrogé devait donner une note comprise entre 1 et 7 permettant d'évaluer sa réaction à ce mot.

Afin de vérifier la cohérence locale de l'espace, l'auteur de la base introduit le chanps sémantique d'un mot en calculant les n mots les plus proches au sens d'une distance dérivée du coéfficient de corrélation. Pour le mot *Mystère*, le chanps sémantique est constitué par la liste suivante:*Inconnu, Orage, Secret, Sauvage, Aventurier, Emotion, Original, Magie, Nuit et Changement*, (ces mots sont écrits dans l'ordre des distances croisssantes).

L'algorithme CTM peut permettre de traiter ce problème: les notes affectées aux mots sont alors considérées comme des variables ordinales. Pour CTM, à chaque cellule est associée 70 table unidimentionnelles de dimension 7. La k^{ieme} table représente les probabilités de la note affectée au k^{ieme} mot: on a

donc pour chaque cellule c_1 et pour chaque k^{ieme} mot, la table de probabilités $\theta^{k,c_1} = \{\theta_j^{k,c_1}, j = 1..7\}$ (§2). L'ensemble de tous les paramètres estimés pour chaque cellule c_1 est égal à $\theta^{c_1} = \cup_{k=1}^{70} \theta^{k,c_1}$ auxquel il faut ajouter les coefficients du mélange correspondant aux cellule c_2 de la deuxième carte $\mathcal{C}_2$ noté θ^{c_2}. A l'aide de cet exemple, nous allons étudier les performances et la robustesse de CTM. En particulier, nous allons investiguer les informations qu'il est possible d'en tirer.

Nous avons effectué l'apprentissage d'une carte CTM de 7×7 cellules et estimé les paramètres des 70 mots dont nous disposons. L'apprentissage a été fait dans les conditions suivantes: à l'initialisation $\theta^{c_2} = \frac{1}{N_{cell}}$ et $\theta_j^{k,c_1} = \frac{\text{nombre d'élément de } \tau_{k,j}}{N_{cell}}$. $T_{max} = 1$ et $T_{min} = 0.1$ avec $N_{iter} = 1000$. A la fin de l'apprentissage si on affecte chaque observation $\mathbf{z}_i$ à la cellule $c = arg\max_{c_2 \in C_2} p_{c_2}(\mathbf{z}_i)$ on obtient la répartition des observations présentée par la figure 1. On observe que la partition obtenue a permis de bien distribuer les observations sur les 49 cellules de la carte.

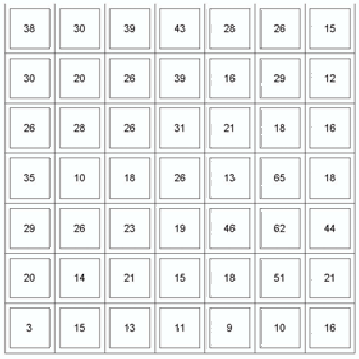


Figure 1: Cardinalité des sous-ensembles, carte CTM 7×7

A l'aide de cette carte 7×7 et des informations probabilistes qui lui sont associées, il est possible d'effectuer un certain nombre d'analyses de la base étudiée. Etant donné que le but des expériences qui suivent est de montrer le bon fonctionnement de CTM, nous nous sommes limités à étudier les effets dûs à quelques mots pour lesquels l'exactitude des propriétés sémantiques retrouvées peuvent être vérifiées.

La figure 2 illustre les distributions de probabilités des notes attribuées au mot *mort*, sous forme d'une carte en niveaux de gris. Chaque carte représente la distribution d'une note j ($j = 1..7$) correspondant à la probabilité estimée $\theta_j^{Mort,c_1} = p(z^{Mort} = j/c_1)$. On remarque que la probabilité est très forte dans la carte correspondant à la note "1" (la carte N1 de la figure 2); on constate que l'ensemble des individus pris dans la base n'apprécient pas la *Mort*. Un nombre restreint d'individus accepte facilement la *Mort* , ils apparaissent sur les cartes 2.N2, 2.N3 et 2.N4, mais avec une faible probabilité qui n'excède pas 0.3 ($\theta_2^{Mort,c_1} \leq 0.3, \theta_3^{Mort,c_1} \leq 0.3, \theta_4^{Mort,c_1} \leq 0.3$). Les notes 5, 6 et 7 sont

présentées par des probabilités très faibles. Le mot *Mort* présente globalement, une stabilité sémantique. Une même analyse peut être effectuée sur chacun des 70 mots. La carte résultat de l'algorithme d'apprentissage CTM permet d'obtenir une visualisation (et une information numérique) sur les distributions des différentes notes attachées à la partition.

La figure 3 représente la distribution de probabilités de la note 1 pour le mot *Guerre*. Cette carte montre une mauvaise appréciation de ce mot par la majorité des individus de la base. Le mot *Guerre* se distingue par une stabilité sémantique. La carte N1 de la figure 2 représente la distribution de probabilités de la note 1 du mot *Mort*, on constate une grande similarité entre les représentations pour le mot *Guerre* et *Mort*, ce qui peut laisser penser que l'ensemble des individus qui compose l'enquête réagit d'une manière similaire aux deux mots. Les groupes de personnes qui n'apprécient pas la *Mort*, n'apprécient pas non plus la *Guerre* (une cellule avec une plus forte probabilité dans la carte N1 de la figure 2 se retrouve avec une plus forte probabilité dans la figure 3). On peut visualiser d'une seconde manière les probabilités estimées par CTM en affichant pour chaque mot uniquement la note qui apparaît avec la probabilité maximale. La figure 4 présente pour le mot *Immobile* et en chaque neurone de la carte, la classe j la plus probable et la probabilité associée. L'étiquette affectée à chaque neurone de la carte représente la note j correspondant à la probabilité maximale, sur la figure , le niveau de gris représente la probabilité maximale. On constate une homogénéité spatial qui apparissent au niveau des notes attribuées à ce mot malgré les probabilités très variées. La probabilité est plus forte pour tous les sous groupes de donner la note 4, cependant plus la zone sur la carte est "claire" et plus d'autre notes sont possible.

La même étude est reprise pour le mot *Fleur* dans la figure 5.a, cette carte présente la distribution de la probabilité maximale du mot *Fleur*. On observe que la majorité des individus sont répartis entre ceux qui ont donné la note 6 et 7 à ce mot avec une forte probabilité pour la note 7. On constate qu'il existe une unicité sémantique du mot *Fleur*. La visualisation des probabilités est une proporiété importante pour les cartes CTM, elle permet d'extraire les différents champs sémantiques de chaque mot. La carte 5.b présente la distribution de probabilité maximale du mot *Séduire*; on voit que la répartition de probabilités et de notes sont proches de celles du mot *Fleur*. Les individus qui préfèrent offrir des fleurs, représentés par une plus forte probabilité (cellules noircies), aiment bien séduire et ils sont représentés aussi par des cellules ayant des notes élevées avec une plus grande probabilité (figure 5.a).

Nous pouvons aussi définir d'une manière plus globale la partition en employant plusieurs mots utilisés. Afficher toutes les distributions de probabilités de tous les mots est évidemment difficile mais réalisable; nous avons décidé d'afficher pour chaque cellule tous les mots qui contiennent dans leur table de probabilités une valeur supérieure à 0.8 ($k = \arg\{\theta_j^{k,c_1} \geq 0.8, j = 1..7\}$). Chaque cellule de la carte présentée par la figure 6 est caractérisé par un ensemble de mots. Ces mots sont notés différemment; ils constituent un ensemble de sens et de qualités représentatifs de chaque cellule. On remarque que les deux côtés (gauche et droite) de la carte opposent des groupes d'individus qui ont des réactions plus tranchées (beaucoup de probabilités supérieures à 0.8) à d'autres groupes moins

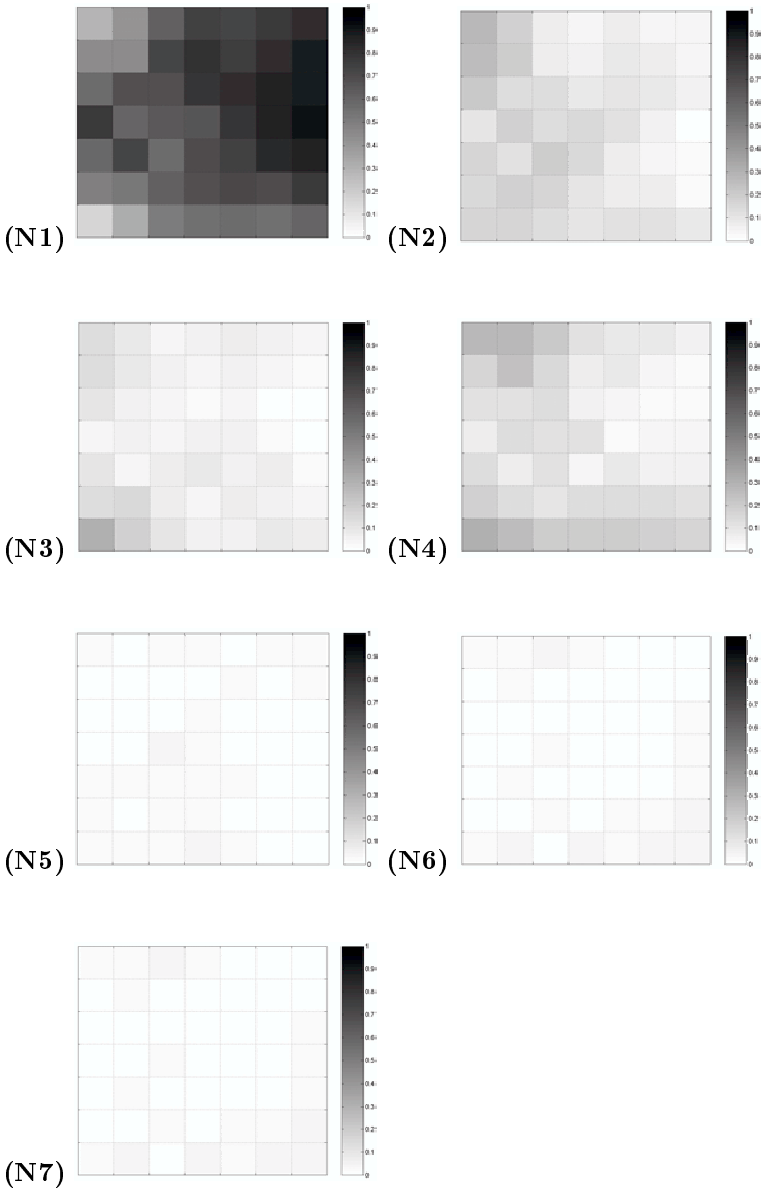


Figure 2: Cartes Topologiques décrivant les distributions de probabilités sur le mot Mort; (Nj): distribution de probabilités correspondant à la note j

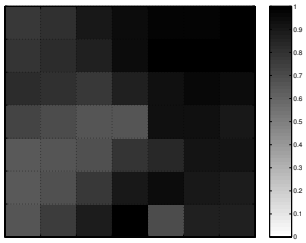


Figure 3: Cartes Topologiques décrivant la distribution de probabilité de la note 1 affectée au mot *Guerre*

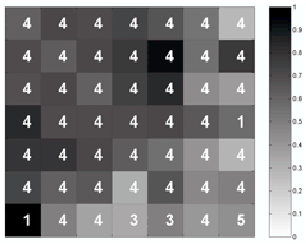


Figure 4: Cartes Topologiques décrivant la distribution de probabilité maximale du mot Immobile

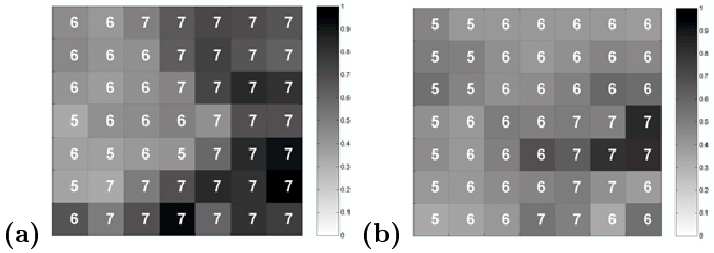


Figure 5: Cartes Topologiques décrivant la distribution de probabilité maximale. (a)du mot Fleur, (b) du mot séduire

attractifs. Avec ce procédé, plusieurs visualisations peuvent être effectuées. Chaque seuil de probabilité représente un niveau d'investigation; il est possible de faire apparaître les mots selon leurs caractéristiques sémantiques. En diminuant le seuil de probabilité de 0.8 à 0.6 et en ne tenant compte que des mots qui ont la même note "1", on obtient une nouvelle carte présentée dans la figure 7. On observe que les mots *Guerre* et *Mort* sont majoritairement mal appréciés par tous les individus tels qu'ils étaient dans la carte 6. On observe de plus le mot *Angoisse* qui apparaît sur la carte 7, d'autres mots tels que *Soldat*, *Punir* et *Immobile* apparaissent en diminuant le seuil de probabilités. Ceux-ci sont du même champ sémantique que *Guerre*.

4 Conclusion

Dans cet article, nous avons présenté un algorithme de carte topologique dédiée aux données catégorielles. Cet algorithme CTM se base sur le formalisme probabiliste des cartes topologiques et utilise l'algorithme EM pour maximiser la vraisemblance. L'expérience présentée montre la robustesse de celui-ci à traiter des données dont les variables peuvent avoir plus de deux modalités. D'autre part, nous avons vu qu'à traversles divers aspects de visualisations, l'algorithme CTM fourni une quantités d'informations exploitables dans des applications réelles.Toutes les visualisation qui ont été effectuées montrent que l'ordre topologique permet d'obtenir une interprétation des groupement obtenus.

References

[1] Anouar, F. Badran, F. Thiria, S. (1998): Probabilistic self-organizing map and radial basis function networks. Neurocomputing 20, 83-96.

[2] Bishop, C. M., Svensen, M., and Williams, C. K. I. (1998): GTM: The Generative Topographic Mapping. Neural Computation, 10(1), 215-234.

[3] Dempster,A. P. Laird, N. M. Rubin, D. B. Maximum liklihoode from incomplete data via the EM algorithm. Journal of royal Statistic Society, Series B, 39, 1-38

[4] Dolinicar, Weingessel , A. Buchta, C. (1998): Dimitriadou, E. A Comparaison of several cluster algorithms on artificial binary data, scenarios from travel market segmentation. Working paper series 19, SFB (adaptive information systems and modelling in economics and management science).

[5] Kaban, A and Girolami, M. (2001): A Combined Latent Class and Trait Model for the Analysis and Visualisation of Discrete Data. I.E.E.E Transactions on Pattern Analysis and Machine Intelligence.23(8), pp859 -.872.

[6] Ibbou, S. Cottrell, M. Multiple correspondance Analysis crosstabulation matrix using the Kohonen algorithm. In verlaeysen, M. Editor proc of ESANN'95, pages 27-32. Dfacto Bruxelles 1995.

	Guerre	Guerre	Guerre	Honnête, Guerre	Honnête, Guerre	Honnête, Guerre
	Guerre	Guerre	Guerre	Etranger Souverain Immobile Guerre	Guerre Honnête	Cadeau, Courage Guerre, Honnête Infini, Moëlleux Mort, Politesse
Guerre	Guerre		Guerre Mort	Guerre Immobile	Fleur Guerre Maison	Cadeau, Courage Dynamique, Fleur Guerre, Honnête Mort, Respect Victoire
Immobile				Angoisse Guerre Politesse	Cadeau Guerre Mort	Humour, Cadeau Courage, Gloire Séduire Maison, Mort Océan, Parfum Respect, Richesse Guerre, Angoisse
				Guerre	Cadeau Courage Fleur Séduire Honnête Maison Mort Guerre	Argent, Bijou Cadeau, Courage Dynamique, Fleur Guerre, Honnête Humour, intime Maison, Mort Politesse, Protéger Respect, Richesse Séduire, Victoire
			Guerre	Fleur Guerre	Fleur Guerre	Charitable, Courage Discipline, Dynamique Fleur, Guerre Honnête, Loi Maison, Politesse Protéger, Respect Travail,Victoire
Dynamique, Humour Peau, Intime Immobile		Guerre, Humour	Fleur, Humour Intime, Guerre		Fleur Guerre	Gloire Guerre

Figure 6: Carte Topologique 7×7, chaque cellule représente un neurone; chacun contient les mots, noté différemment, ayant une probabilité supérieure à 0.8

Guerre	Guerre	Guerre	Guerre, Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort
Guerre	Guerre	Guerre, Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort
Guerre	Guerre, Mort	Guerre, Mort	Guerre, Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort
Guerre, Mort	Guerre, Mort	Guerre, Mort	Guerre	Angoisse, Guerre Mort	Angoisse, Guerre Mort	Angoisse, Critiquer Guerre, Mort Punir
Guerre	Guerre, Mort	Guerre	Guerre	Guerre, Mort	Angoisse, Guerre Mort	Angoisse, Critiquer Mort
Guerre	Guerre	Guerre, Mort	Guerre, Mort	Angoisse, Guerre Mort	Guerre	Guerre, Mort
Angoisse, Guerre Immobile, Soldat	Guerre	Angoisse, Guerre Soldat	Angoisse, Guerre Soldat	Guerre	Guerre	Guerre

Figure 7: Carte Topologique 7×7, chaque cellule représente un neurone; chacun contient les mots ayant une probabilité supérieure à 0.6 avec la même connotation 1

[7] Kaski. S, Honkela. T, Lagus. K, and Kohonen. T. (1998). WEBSOM–self-organizing maps of document collections. Neurocomputing, volume 21, pages 101-117.

[8] Kohonen, T. (1994): Self-Organizing Map. Springer, Berlin.

[9] Lebbah. M, Thiria. S, Badran. ESANN, Topological Map for Binary Data, ESANN 2000, Bruges, April 26-27-28, 2000, Proceedings.

[10] Lebbah. M , Thiria. S, Badran. F, Chabanon. C. ICANN 2002, Categorical Topological map, Madrid 2002.

[11] Lebart, L. Piron, M. Steiner J.-F. La sémiométrie. Dunod, Paris. 2003.

[12] Leich, F. Weingessel, A. Dimitriadou, E. (1998): Competitive Learning for Binary Data. Proc of ICANN'98, septembre 2-4. Springer Verlag.

[13] Luttrel S. P. (1994). A Bayesian Ananlysis of Self-Organizing Maps, Neural Computing vol 6

[14] McLachlan, G. Krishman, T. The EM algorithm and Extensions. Wiley, New York, 1997.

[15] Steiner J.-F., Auliard, O. La sémiometrie: un outil de validation des réponses, In : La Qualité de l'Information dans les Enquêtes / Quality of Information in Sample Surveys, ASU, (L. Lebart ed.), Dunod, Paris, p 241-274, 1992.

Détection de motifs dans des images médicales par apprentissage supervisé

Arthur Tenenhaus, Maxime Saporta,
Alain Giron, Bernard Fertil

INSERM unité 494 équipe 6
91 boulevard de l'hôpital
Faculté de médecine de la Pitié Salpetrière
arthur.tenenhaus@imed.jussieu.fr
msaporta@free.fr
alain.giron@imed.jussieu.fr
bernard.fertil@imed.jussieu.fr

Résumé. Nous développons actuellement une méthode générale de détection automatique de motifs dans des images médicales. Fondée sur des méthodes d'apprentissage supervisé, notre approche est suffisamment générique pour permettre de traiter trois problèmes relativement différents : la détection des vaisseaux sanguins et du nerf optique dans des images de fond d'œil et la segmentation du réseau pigmenté dans les tumeurs noires de la peau.

1. Introduction

Un des projets du laboratoire d'imagerie médicale quantitative de l'unité INSERM U494 est la mise au point d'un système d'identification automatique de zones pathologiques de la rétine. La détection automatique des vaisseaux sanguins et du nerf optique (structures invariantes de l'œil) est une étape préliminaire qui permettra d'une part le recalage des images nécessaire au suivi médical des patients, d'autre part d'affiner la qualité de détection des maladies de la rétine en tenant compte de ces zones typiques.

Un second projet est la caractérisation automatique de tumeurs noires de la peau. Selon les dermatologues, un des critères de malignité d'une lésion est la présence d'une texture particulière nommé le réseau pigmenté. Nous souhaitons donc construire un système de détection capable de localiser des zones de réseau pigmenté de manière automatique.

Il convient donc de développer une méthodologie générale de détection de motifs. Exploitant l'expertise médicale, notre approche est basée sur des méthodes d'apprentissage supervisé choisies pour gérer la grande variabilité des images.

2. Matériels et méthodes

2.1. Chaîne de traitement

Le processus de création d'un classifieur illustré par la Figure 1 se décompose en trois étapes. La première étape consiste à construire une collection d'images reflétant au mieux la variabilité des régions d'intérêts. Partant du constat qu'un motif s'identifie visuellement sur une petite parcelle d'image (imagette), notre collection sera constituée d'imagettes. Cette collection, servant de base d'apprentissage, doit rendre compte des différentes formes du signe observé dans des régions d'intérêt (ROI : Region Of Interest), et doit être accompagnée d'un ensemble représentatif de contre-exemples. Le système de détection se réduit ici à un problème de classification à deux classes.

Une fois la collection d'imagettes constituée, plusieurs prétraitements sont appliqués sur les données pour rehausser les différences entre les deux classes, réduire la dimension des données et / ou extraire des attributs caractérisant le motif.

Une analyse statistique des attributs extraits de chacune des imagettes nous permettra de détecter des informations pertinentes pour la classification créant ainsi un système de discrimination de motifs. Ce système pourra alors être utilisé dans l'étape finale d'interprétation d'une nouvelle image. Nous cherchons donc à différencier les imagettes en présence d'un signe de leurs contre-exemples.

La chaîne de traitement (Figure 1) est illustrée par un exemple de détection de vaisseaux sanguins. Les principes sont les mêmes pour la détection de nerf optique ou de réseau pigmenté.

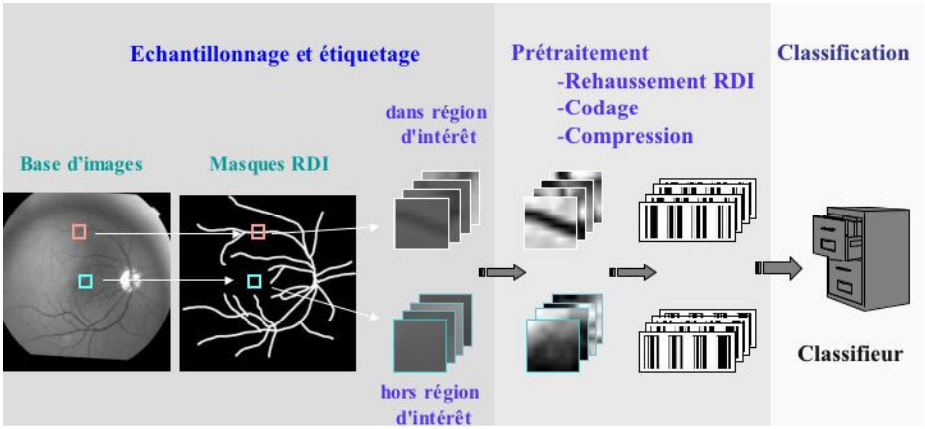


Figure 1. Création d'un classifieur

2.2. Constitution de la base d'imagettes

La performance d'un système de détection est mesurée sur sa capacité de généralisation, c'est-à-dire sur son aptitude à classer correctement de nouvelles images, différentes de celles utilisées dans l'ensemble d'apprentissage. La constitution d'un ensemble représentatif de données est donc capitale.

2.2.1. Le nerf optique et les vaisseaux sanguins

L'expertise des médecins nous a permis de définir les zones de vaisseaux sanguins (ROI) dans 23 images en niveau de gris. De chacune de ces zones furent extraites 25 imagettes carrées de 8 pixels. Nous disposions ainsi de 575 imagettes de vaisseaux sanguins auxquelles ont été ajoutées 575 imagettes carrées de 8 pixels sélectionnées en dehors des zones de vaisseaux sanguins. Dans le but d'améliorer la détection une standardisation des données ainsi qu'une égalisation d'histogramme ont été appliquées aux imagettes de vaisseaux sanguins.

Similairement, l'expertise des médecins nous a permis de définir les zones de nerf optique (ROI) de 47 images en niveau de gris. De chacune de ces zones furent extraites 25 imagettes carrées de 12 pixels. Nous disposons ainsi de 1175 imagettes de nerf optique auxquelles ont été ajoutées 1175 imagettes carrées de 12 pixels sélectionnées en dehors des zones du nerf otique.

2.2.2. Le réseau pigmenté

L'expertise des dermatologues nous a permis de définir 30 zones d'images couleurs en présence de réseau pigmenté (ROI) ainsi que 30 autres en l'absence de réseau pigmenté. De chacune de ces zones furent extraites 256 imagettes carrées de 32 pixels. Nous disposons ainsi de 7680 imagettes en présence de réseau auxquelles furent ajoutées 7680 imagettes en l'absence de réseau. Nous considérons, dans la suite du document, le niveau de gris des imagettes de réseau pigmenté, transformation conservant la structure de cette texture.

2.3. Extraction d'informations

La conception du module d'extraction d'informations doit faire l'objet d'une attention particulière : la performance générale du système est directement liée à la stratégie qui aura été mise en oeuvre pour extraire des attributs "pertinents" à partir des imagettes.

2.3.1. Nerf optique et vaisseaux sanguins

Compte tenu de la dimension des imagettes de vaisseaux sanguins et de nerf optique, l'utilisation du pixel comme variable d'intérêt semble un objectif raisonnable. Ainsi à chacune des imagettes est associée la valeur des pixels qui la compose ainsi que sa provenance (présence - absence du motif d'intérêt).

2.3.2. Le réseau pigmenté

Contrairement aux vaisseaux sanguins et au nerf optique, l'utilisation du pixel comme connaissance pose des problèmes de dimension. En effet, des imagettes de petites dimensions ne capturent pas la structure du réseau pigmenté. Il devient alors indispensable de considérer des imagettes plus grandes. Pour apprécier visuellement la structure du réseau pigmenté, nous considérons des imagettes carrées de 32 pixels. Il en résulte que le nombre de variables caractérisant une imagette est de 1024. Une des solutions permettant de réduire la

dimension du problème est de construire, pour chacune des imagettes, un vecteur de caractéristiques capturant l'information de texture.

Le réseau pigmenté (Figure 2) est caractérisé par la présence d'amas d'alvéoles approximativement identiques et réguliers. Il peut être observé sur des imagettes en niveau de gris.

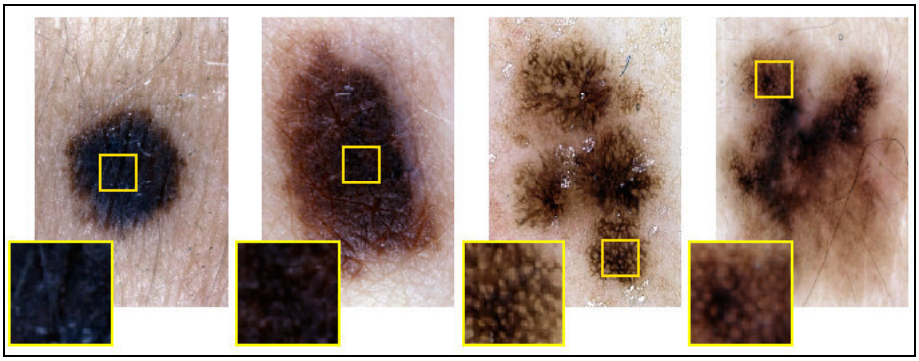


Figure 2. Tumeurs noires de la peau (à gauche 2 tumeurs sans réseau à droite 2 tumeurs avec réseau)

Compte tenu de la structure de ces textures, nous pouvons considérer une imagette en présence de réseau pigmenté comme un signal bidimensionnel approximativement invariant par translation présentant de fréquentes zones d'irrégularité. Similairement, une imagette en l'absence de réseau pigmenté peut être considérée comme un signal bidimensionnel approximativement invariant par translation mais ne présentant que peu de zones d'irrégularité. La transformation en ondelettes, qui permet une analyse locale des signaux ainsi que la mise en évidence des zones d'irrégularités, est une approche appropriée pour discriminer ces deux textures. Nous utilisons la transformation stationnaire en ondelettes (SWT) qui est une variante de la transformation discrète en ondelettes orthogonales (DWT) de Mallat. Nous soulignerons que la SWT est une décomposition invariante par translation. Cela signifie que la *SWT* d'une version translatée du signal est la version translatée de la *SWT* du signal. Cette propriété est intéressante dans la détection de motif et donc dans la reconnaissance de texture [1], [2], [3], [4].

La Figure 3 illustre la décomposition d'une imagette résultant de la SWT avec les paramètres suivants :

Chacune des lignes L_i, présente une matrice de coefficients d'approximation de niveau i (A_i) suivis des trois matrices de coefficients de détails de même niveau (D_h^i, D_v^i et D_d^i).

Nous représentons la signification de ces coefficients par l'équation suivante :

$$\textit{Image originale} = A_1 + D_h^1 + D_v^1 + D_d^1$$

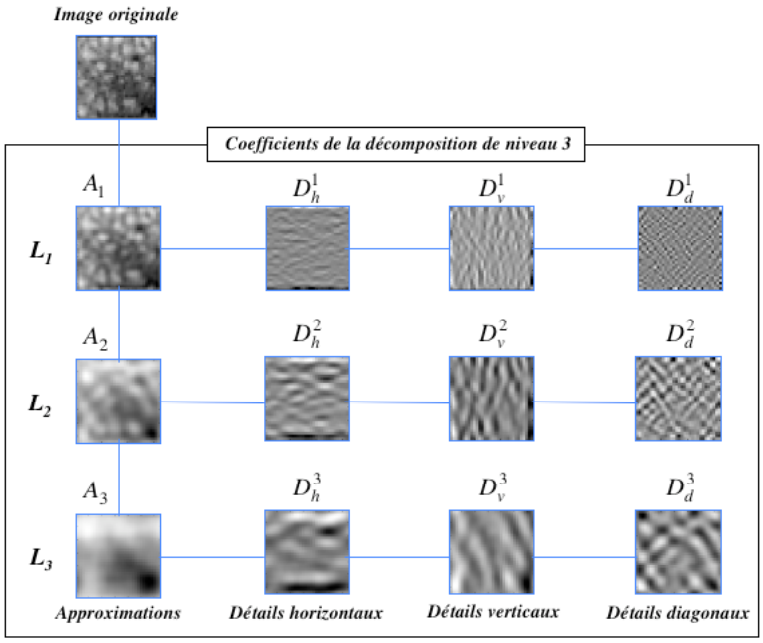


Figure 3. Décomposition de niveau 3 d'un signal bidimensionnel à l'aide de la SWT

En d'autres termes, la décomposition fournit une approximation A_1 de l'image originale. La perte d'information engendrée par cette approximation est encodée par les trois images de détails. L'image D_h^1 encode les détails horizontaux perdus dans l'approximation alors que D_v^1 (respectivement D_d^1) encode les détails verticaux (respectivement diagonaux) perdus dans l'approximation. Cette décomposition est un processus itératif. Elle s'intéresse à des niveaux de détails de plus en plus grossiers et dans trois directions. A chaque étape de la décomposition, on conserve les coefficients de détails et on décompose les coefficients de l'approximation. Chacune des lignes L_i, présente les coefficients de l'approximation de niveau i suivis des coefficients des trois détails de même niveau.

Calcul des coefficients : Stéphane Mallat [9] établit un lien entre la théorie générale des ondelettes et les bancs de filtres utilisés en traitement du signal. Toutes ces opérations de décomposition se réduisent à des produits de convolution avec des filtres spécifiques. En effet, pour passer du niveau de décomposition i au suivant, on filtre les lignes de l'image d'approximation de niveau i par un filtre passe-bas (respectivement un filtre passe-haut), puis on filtre les colonnes des résultats obtenus par un filtre passe-bas (respectivement un filtre passe-haut). L'image dont les colonnes auront été filtrées par un filtre passe-haut et les lignes par un filtre passe-bas représentera les hautes fréquences verticales, c'est-à-dire les coefficients de détails verticaux. Un procédé semblable fournit les coefficients de détails horizontaux et diagonaux. Une quatrième image, dont les colonnes et les lignes auront été filtrées par un filtre passe-bas représentera l'image de résolution moindre, c'est-à-dire l'image d'approximation depuis laquelle on relance l'algorithme jusqu'au niveau de décomposition désiré. Les coefficients de détails, appelés aussi coefficients d'ondelettes sont représentés par les trois dernières colonnes de la Figure 3. Qualitativement, les fortes

variations des coefficients d'ondelettes fournissent une information sur l'irrégularité locale du signal. La variance des coefficients d'ondelettes devrait donc capturer efficacement cette information.

Ainsi, pour un niveau de décomposition J, les vecteurs de caractéristiques sont donc composés des $3J$ variances de coefficients d'ondelettes de la décomposition ainsi que la variance des pixels de l'image d'approximation de dernier niveau. Par conséquent, Pour un niveau de décomposition J, une imagette est définie par $3J+1$ variances auxquelles nous associons la provenance du signal (présence - absence de réseaux pigmentés). Nous pouvons enfin souligner que les coefficients d'ondelettes résultant de la SWT sont de la même taille que l'image originale quelque soit le niveau de décomposition.

2.4. Classification

Une analyse statistique des paramètres extraits de chacune des imagettes permet de détecter des informations pertinentes pour la classification créant ainsi un système de discrimination des motifs.

Des méthodes linéaires et des méthodes non linéaires sont conjointement exploitées pour analyser les mêmes ensembles de données. Les méthodes linéaires sont souvent rapides, mais offrent généralement de moins bons résultats que les méthodes non linéaires. Leurs vitesses permettent d'avoir rapidement une idée sur la qualité de la base de données et d'obtenir des références pour les méthodes non linéaires.

2.4.1. Les méthodes linéaires :

2.4.1.1. La Régression logistique (RL)

La régression logistique est une méthode statistique adaptée à l'étude de la liaison entre une variable qualitative Y et un ensemble de variables quantitatives $X_1,...,X_p$. Elle modélise la probabilité d'appartenance à une classe.

Notons $\pi(x) = P(Y = 1/X = x)$. Le modèle qui décrit le lien entre $\pi(x)$ et x est :

$$\pi(x) = \frac{e^{g(x)}}{1+e^{g(x)}}, \text{ où } g(x) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$

La méthode d'estimation des paramètres $(\beta_j)_{1 \le j \le p}$ est basée sur le maximum de vraisemblance [9].

2.4.1.2. La Régression PLS

La régression PLS (Partial Least Squares) est une méthode d'analyse de données fournissant un modèle linéaire. Le modèle généré par la régression PLS à k composantes s'écrit :

$$Y = \sum_{h=1}^{k} c_h \underbrace{\sum_{j=1}^{p} a_{hj} x_j}_{t_h} + résidu$$

Nous noterons que les composantes PLS sont construites de la manière suivante : pour chaque $h = 1,...,k$, nous recherchons des composantes $t_h = Xa_h$ maximisant le critère $Cov(Xa_h, y)$ sous des contraintes de norme ($\|a_h\| = 1$) et d'orthogonalité entre t_h et les composantes précédentes $t_1,...,t_{h-1}$. On soulignera que les composantes PLS t_h sont des combinaisons linéaires des variables d'origine [8].

Nous appelons régression logistique PLS, la régression logistique de Y sur les composantes PLS construites à partir de X et Y.

2.4.2. Les méthodes non linéaires

2.4.2.1. Le perceptron multicouche (réseaux de neurones, NN)

Le perceptron multicouche est un algorithme itératif qui cherche à minimiser une fonction de coût de la forme : $\sum \left(c(x) - o(x)\right)^2$ où $c(x)$ désigne les sorties observées et $o(x)$ les sorties prédites par le réseau de neurones. Nous utilisons des techniques numériques d'optimisation dont la plus usuelle est l'algorithme de rétro-propagation du gradient pour résoudre ce problème de minimisation [5].

2.4.2.2. Les Support Vector Machines (SVM)

Supposons qu'il existe plusieurs hyperplans permettant de séparer les exemples d'une classe des exemples de l'autre classe, Les SVM ne se contentent pas d'en trouver un, mais cherchent parmi ceux-ci celui qui passe « au milieu » des points des deux classes d'exemples. Intuitivement, cela revient à chercher l'hyperplan le plus fiable. En effet, supposons qu'un exemple n'ait pas été décrit parfaitement. Si sa distance à l'hyperplan est grande, une petite variation ne modifiera pas sa classification. Formellement, cela revient à chercher un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale. Nous appellerons cette séparation l'hyperplan séparateur optimal.

Les SVM se nourrissent de ce principe, mais plutôt que de rechercher l'hyperplan séparateur « optimal » dans l'espace où les descripteurs sont définis, ils recherchent cet hyperplan dans un espace de plus grande dimension, fournissant ainsi une frontière non linéaire dans l'espace d'origine [5].

2.4.2.3. La Ridge Regression sous contraintes de complexité (RR+VC)

Cette méthode lie la Ridge Regression à la dimension de Vapnik-Chervonenkis (VC dimension). À l'aide de la VC dimension, on établit de manière dynamique au cours de l'apprentissage, un compromis entre la complexité du modèle et sa robustesse. Dans ce travail, nous utilisons une version développée par la société KXEN. Cette version fait appel à un prétraitement spécifique des données qui s'appuie sur les relations entre les variables d'entrée et la variable à prédire. On peut, par ailleurs, étendre l'ordre des variables du modèle (carré, cube).

2.4.2.4. La Kernel PLS (KPLS)

La kernel PLS s'appuie sur les mêmes principes que les SVM. Plutôt que de rechercher les composantes PLS dans l'espace d'origine, on se plonge dans un espace étendu (le « kernel trick »). On obtient ainsi un modèle non linéaire. On peut également exploiter le même principe pour la régression sur composantes principales (PCR) ou la Ridge Regression (RR). Ce principe donnant naissance à la kernel PCR (KPCR) ou la Kernel RR [6].

3. Conclusions et résultats

La démarche de travail fut la suivante : on sélectionne aléatoirement deux tiers des observations (base d'apprentissage) pour construire le classifieur et le tiers restant (base de test) permet d'évaluer sa robustesse et de le valider.

3.1. Le nerf optique et les vaisseaux sanguins

3.1.1. Le nerf optique

Les tests ont montré que les opérations d'égalisation d'histogramme et de standardisation n'apportent aucune amélioration. Ces prétraitements augmentent le contraste des imagettes, mais la présence de vaisseaux sanguins dans la zone du nerf optique induit le classifieur en erreur. Les pourcentages sont donc présentés pour les données brutes.

Modèle	Apprentissage (%)	Test (%)
RL	88	85
LOGISTIQUE-PLS	87	86
SVM	99	98
NN	99	96
RR + VC (ordre 1)	87	84
KPLS	99	96
KPCR	93	90

Les différents résultats nous permettent d'apprécier la performance des méthodes non linéaires face aux méthodes linéaires. Les méthodes non linéaires, et en particulier les SVM offrent de très bons résultats. Le temps de calculs n'étant pas un obstacle compte tenu de la taille de l'échantillon.

3.1.2. Les vaisseaux sanguins

Les tests ont montré que la standardisation des imagettes améliore le taux de bonne classification. Ceci peut s'expliquer par le fait que la standardisation des imagettes engendrent une augmentation du contraste donc une distinction plus fine des vaisseaux sanguins. Les pourcentages sont donc présentés sur des imagettes standardisées de 8 pixels.

Modèle	Apprentissage (%)	Test (%)
RL	80	77
LOGISTIQUE-PLS	80	79
SVM	97	89
NN	99	87
RR + VC (ordre 1)	88	83
KPLS	99	82
KPCR	90	86

On constate que les résultats sont du même ordre que ceux du nerf optique, notamment pour les SVM. De manière générale, nous disposons de plusieurs classifieurs permettant de faire des comparatifs de performance et de temps de calcul.

3.2. Les tumeurs noires

Les résultats sont présentés pour un niveau de décomposition d'ordre 3 sur des imagettes carrées de 32 pixels en niveau de gris.

Modèle	Apprentissage (%)	Test (%)
RL	68	67
PLS	66	65
NN	83	82
RR + VC (ordre 3)	84	82
SVM	85	84

KPLS et KPCR implique la gestion d'une matrice carrée de dimension n, où n correspond au nombre d'observations de la base d'apprentissage. Ceci rend impossible la gestion d'un grand nombre d'observations. Les SVM impliquent généralement la gestion de cette même matrice, mais une approche due à Edgar Osuna et Al [10] permet la gestion de jeu d'apprentissage de grande taille. Le principe revient à construire l'hyperplan progressivement à partir des sous-échantillons de la base d'apprentissage. Dans ces conditions, nous présenterons les résultats de KPLS et KPCR uniquement pour une partie de la base d'apprentissage (Figure 4).

Le temps d'apprentissage est un facteur à prendre en considération lorsque la taille de l'échantillon est importante. Les différences sont nettement marquées. En effet, pour un ensemble de 10240 observations, l'apprentissage dure 4 jours pour les SVM, une demi-journée pour les réseaux de neurones et 20 minutes pour RR + VC (ordre 3), le logiciel de KXEN.

D'une lésion à l'autre, le réseau pigmenté et plus généralement les tumeurs noires, peuvent présenter des aspects très variés. Cette grande diversité implique naturellement une importante variabilité de l'échantillon d'apprentissage. Cette variabilité perturbe la classification comme l'illustre la Figure 4.

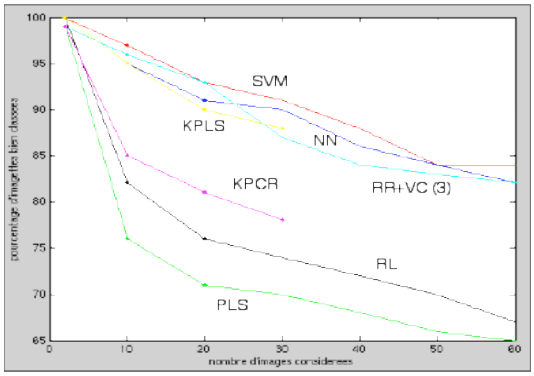


Figure 4. Classification en fonction du nombre d'images dans l'échantillon d'apprentissage

Nous distinguons deux niveaux de performance selon la méthode de classification. Les méthodes linéaires comme la régression PLS ou la régression logistique ne fournissent pas de résultats satisfaisants. Des méthodes plus puissantes comme les SVM, les réseaux de neurones ou la kernel PLS atteignent des pourcentages de classification intéressants, mais le coût de calcul dû principalement à la taille de l'échantillon reste dissuasif. RR+VC (ordre 3) se détache du lot offrant un niveau de classification très comparable aux méthodes statistiques les plus puissantes et est très peu pénalisé par le temps de calcul.

3.3. Interprétation des images

Il est indispensable d'évaluer la qualité de classification en analysant de nouvelles images à l'aide des modèles construits.

3.3.1. Le nerf optique et les vaisseaux sanguins

Les méthodes traditionnelles d'analyse d'image comme le seuillage offrent des résultats intéressants. Néanmoins, elles souffrent d'un sérieux inconvénient : d'une image à l'autre, le seuil de détection varie (Figure 5). Les méthodes d'apprentissage statistique gère cette difficulté en réalisant l'équivalent d'un seuillage adaptatif (Figure 6).

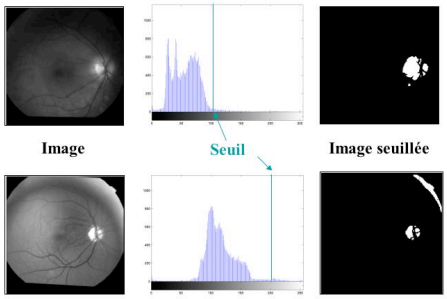


Figure 5. Détection du nerf optique par seuillage

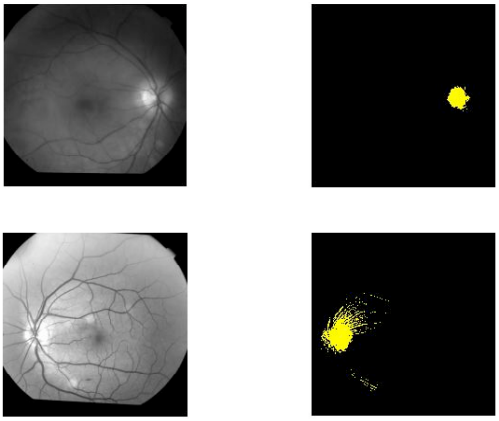


Figure 6. Influence de l'intensité lumineuse de l'image sur la détection du nerf optique Image originale (gauche) vs image interprétée (droite)

La détection de vaisseaux sanguins par masque de kirsch reste très sensible à la variabilité lumineuse des images, alors qu'une régression logistique permet d'atténuer l'influence de cette variabilité (Figure 7).

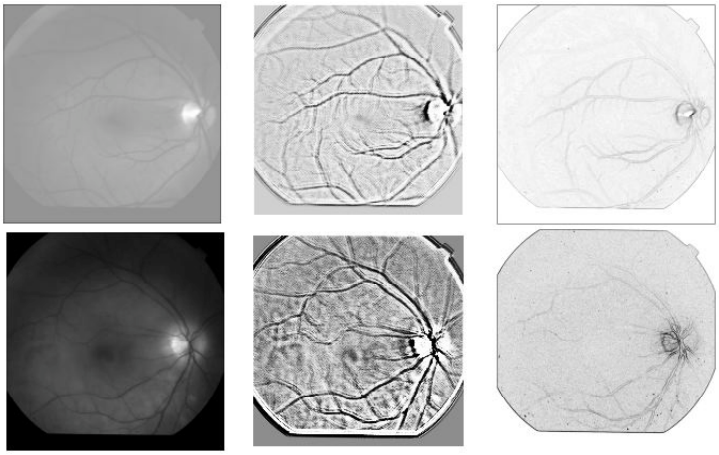


Figure 7. Détection des vaisseaux sanguins - Image originale (gauche) vs image interprétée (centre) vs image interprétée par Kirsch (droite)

Contrairement aux méthodes classiques d'analyse d'image, les méthodes par apprentissage offrent une robustesse d'analyse.

3.3.2. Tumeurs noires

Le réseau pigmenté symbolisé par du blanc sur l'image interprété par les SVM est, compte tenu de la difficulté de segmentation, repéré efficacement (Figure 8).

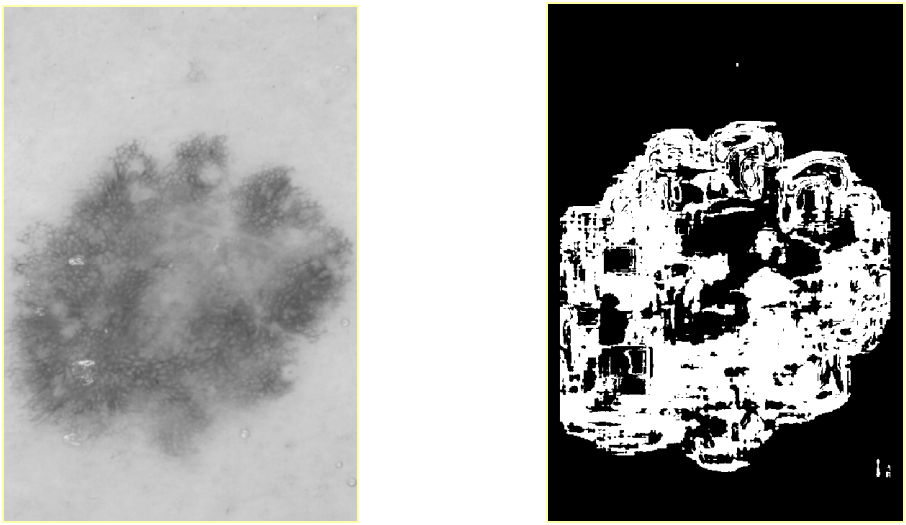


Figure 8. Image originale (gauche) vs image interprétée par les SVM (droite)

4. Conclusion

Les SVM sont un outil d'analyse puissant pour la détection de motifs sur des images médicales. Ils permettent la gestion d'un nombre très élevé de variables. On peut donc penser à une gestion du pixel quelque soit la taille de l'imagette. Des tests effectués sur les pixels des imagettes de réseau pigmenté offrent des résultats très prometteurs. Malheureusement les SVM souffrent d'un inconvénient majeur : le temps de calcul des SVM augmente rapidement en fonction de la taille de la base d'apprentissage $O(n^2)$.

Afin de résoudre les problèmes de temps de calcul pour les SVM et de dimension des données pour la KPCR et KPLS l'utilisation de prototypes créés à partir de méthodes comme les K-Means peuvent s'avérer judicieux.

Les différents résultats obtenus sont encourageants pour mener à bien les prochaines étapes : fusions des classifieurs détecteurs de pathologies avec ceux du nerf optique et des vaisseaux sanguins, recalages d'images, et intégration de nouveaux marqueurs pour améliorer le classifieur des tumeurs concernant la forme et la couleur des naevus.

Remerciements : Les auteurs remercient la société KXEN pour la mise à disposition de leurs modules d'analyse de données K2C et K2R.

Bibliographie

[1] M. Misiti, Y. Misiti, G. Oppenheim, J.M. Poggi, Les ondelettes et leurs applications, Lavoisier, 2003

[2] Michael Unser, Texture Classification and segmentation Using Wavelet frames, IEEE transactions of image processing, Volume 4, n°11, November 1995

[3] Gloria Menegaz, Advanced Image Processing.

[4] Stéphane Valaeys, Analyse de texture pour la détection de mélanomes malins, 2001
[5] T. Hastie, R. Tibshirani, J. Friedman, The element of statistical learning, Springer, 2001
[6] Roman Rosipal & Leonard J. Trejo, Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space, Journal of Machine Learning Research, Decembre 2001.
[7] Stéphane G. Mallat, A Theory for multiresolution Signal Decomposition : The Wavelet Representation, IEEE transactions on pattern analysis and machine intelligence, vol 11, n°7, juillet 1989.
[8] Michel Tenenhaus, La régression PLS, Technip, 1998
[9] Gilbert Saporta, Probabilités, analyse des données et statistique, Editions Technip, 1990
[10] Edgar Osuna, Robert Freund, Frederico Girosi, An Improved training algorithm for Support Vector Machines, IEEE NNSP'97, septembre 1997.

Summary

An automatic detection of patterns in medical images is under developpement in our laboratory. It is based on a supervised learning by sample approach which provides a versatile framework: three independent classifiers are developed in order to detect, on one hand, the blood vessels and the optic disc in retinophotos and to segment the pigmented network of skin lesions, on the other hand.

SOM-ART : Incorporation des propriétés de plasticité et de stabilité dans une carte auto-organisatrice.

Farida Zehraoui , Younès Bennani

LIPN-CNRS, UMR 7030, Université Paris 13
99, Av Jean Baptiste Clément 93430 Villetaneuse, France
{younes.bennani,farida.zehraoui}@lipn.univ-paris13.fr

Résumé. Dans cet article nous nous sommes focalisés sur l'utilisation des réseaux de neurones artificiels pour le traitement des données complexes évolutives. Les réseaux de neurones artificiels "statiques" comme la carte SOM (Self Organizing Map) ne permettent pas d'acquisition de nouvelles connaissances (plasticité) tout en maintenant les anciennes (stabilité). Plusieurs versions des cartes auto-organisatrices évolutives comme "Growing Cell Structure (GCS)" et "Growing Neural Gas (GNG)" ont été proposées pour acquérir en continu de nouvelles connaissances, mais ne tiennent pas compte des anciennes déjà apprises. Le compromis entre la plasticité et la stabilité est souvent appelé : "dilemme stabilité-plasticité". La théorie de la résonance adaptative (ART : Adaptive Resonance Theory) est spécialement conçue pour faire face à ce dilemme. Les modèles ART sont parmi les rares réseaux de neurones artificiels possédant les deux propriétés. Nous nous sommes intéressés à une classe assez récente de ces modèles qui est la classe "Simplified ART". Ce papier présente une nouvelle version des cartes auto-organisatrices baptisée : "SOM-ART" pour le classement et la classification de séquences. Cette nouvelle version possède les propriétés de stabilité et de plasticité. Le modèle SOM-ART est une intégration d'une carte SOM évolutive dans un paradigme basé sur la théorie de la résonance adaptative .

1 Introduction

Notre travail est réalisé dans le cadre d'un projet qui vise à étudier et à développer des méthodes d'apprentissage de comportements d'utilisateurs à partir de séquences de navigations sur le Web. Nous nous sommes intéressés à une application concrète qui consiste à étudier le comportement d'un internaute à partir de

traces de navigations. Nous disposons de données comportementales complexes contenues dans des fichiers de traces de navigations (fichiers log) issus d'un site de commerce électronique. Ces données sont caractérisées par leur volume très important et leur aspect temporel et évolutif. De plus, ces données contiennent du bruit et les connaissances à priori du domaine ne sont pas disponibles. Nous nous sommes intéressés aux approches connexionnistes qui sont connues pour leur efficacité dans le traitement de gros volumes de données ainsi que pour la possibilité de leur utilisation en l'absence des connaissances du domaine.
Dans cet article nous nous sommes focalisés sur la possibilité de l'utilisation de ces réseaux à long terme sachant que les données sont évolutives. Comment un réseau de neurones artificiels peut-il acquérir continûment de nouvelles connaissances (plasticité) tout en préservant les anciennes déjà acquises (stabilité) ?
Les modèles de réseaux de neurones artificiels évolutifs n'ont pas de structure prédéfinie. Ils sont générés à partir d'une succession d'ajouts (ou de suppressions) d'éléments. Ceux-ci s'adaptent aux changements des entrées. Certains sont capables d'acquérir en continu les nouvelles connaissances mais sans aucune garantie que les anciennes, déjà acquises, soient préservées. Comment un système d'apprentissage peut-il posséder la propriété de plasticité en réponse aux nouvelles entrées tout en restant stable ?
La plupart des réseaux de neurones artificiels sont soit stables mais incapables d'acquérir de nouvelles connaissances, soit possèdent la propriété de plasticité mais pas celle de la stabilité. La théorie de résonance adaptative (ART : Adaptive Resonance Theory) [CS88] est spécialement conçue pour faire face au dilemme plasticité-stabilité. Les réseaux de neurones de la famille ART possèdent des fondements théoriques solides, mais les premiers réseaux proposés sont compliqués et difficiles à implém- enter. Nous nous sommes intéressés particulièrement à la classe des réseaux "Simplified ART" [BA98]. Cette classe simplifie la structure d'un réseau ART et rend l'intégration d'une carte SOM possible. Nous proposons une nouvelle carte auto-organisatrice incrémentale construite suivant le paradigme Simplified ART.
La suite de l'article est organisé comme suit. Dans le paragraphe 2, nous décrivons certaines cartes auto-organisatrices évolutives ainsi que certains modèles ART. Dans le paragraphe 3, nous présentons notre approche. Dans le paragraphe 4, nous montrons les résultats de validation que nous avons obtenus. Dans le paragraphe 5, nous présentons les travaux en cours ainsi que les perspectives.

2 Etat de l'art

Les cartes topologiques auto-organisatrices "SOM" sont parmi les modèles connexionnistes les plus utilisés pour la classification et la visualisation des données [Koh95]. Elles permettent de projeter des données de dimension quelconque dans un espace discret de faible dimension. La carte SOM est composée de deux couches de neurones : une couche d'entrée et une couche compétitive de sortie (voire figure 1 (c)). La couche compétitive est celle qui s'occupe du traitement

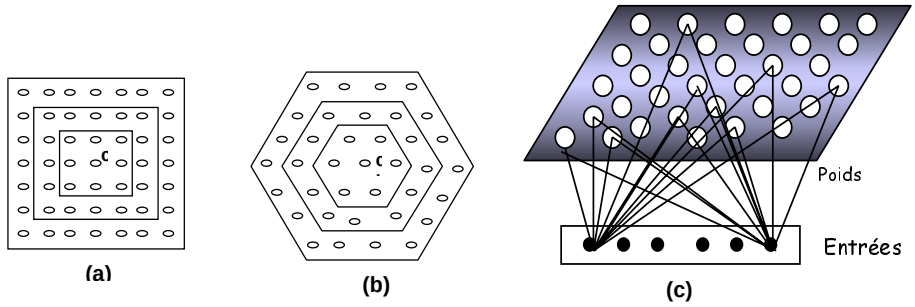


Figure 1: *Architecture à deux couches d'une carte topologique de dimension 2.*

des données d'entrée. Sa structure topologique est prédéfinie, la plus utilisée est celle sous forme d'une grille carrée (voire la figure 1 (a)). L'algorithme d'apprentissage de SOM est essentiellement effectué suivant trois phase :

- la phase d'initialisation où des valeurs aléatoires sont affectées aux poids des neurones de sortie de la carte,
- la phase de compétition dans laquelle, pour toute entrée ξ, un neurone $s(\xi)$ est sélectionné de la manière suivante :

  $$s(\xi) = \arg\min_{c \in A} dist(\xi, w_r)$$

  où A est l'ensemble des neurones de sortie et dist(,) est la distance euclidienne,
- la phase d'adaptation où les poids de chaque neurone de sortie r sont mis à jour de la manière suivante :

  $$\Delta w_r = \epsilon(t) h_{rs}[\xi - w_r]$$

  où $\epsilon(t)$ est le pas d'apprentissage et h_{rs} est une fonction de voisinage.

Plusieurs versions évolutives de SOM ont été proposées dans la littérature pour faire face à quelques problèmes liés à l'utilisation des cartes SOM pour le traitement de données évolutives comme le nombre de neurones qu'il faut déterminer au préalable, la nécessité de refaire l'apprentissage sur toutes les données quand de nouvelles données se présentent, ... etc.
Le modèle "Growing Gell Structure" (GCS) [Fri93] possède une structure constituée d'hypertétraèdres de dimensionnalité fixée. Le modèle est initialisé par un hypertétraèdre. A chaque neurone est associée une erreur cumulée qui représente la somme des erreurs commises lors de la classification des entrées. Après un nombre λ d'étapes d'adaptation, l'unité q ayant le maximum d'erreurs cumulées est déterminée et une nouvelle unité est insérée. De plus, d'autres connexions sont insérées afin de préserver la structure d'hypertétraèdres de même

dimension. Le modèle "Growing Neural Gas (GNG)" [Fri95] n'impose aucune contrainte sur son graphe. De plus, le graphe généré est continûment mis à jour par un apprentissage Hebbien compétitif. Celui-ci consiste, simplement, à créer une connexion entre le neurone gagnant et le second gagnant à chaque étape d'adaptation. Après un nombre fixé λ d'adaptations, l'unité q ayant l'erreur cumulée maximum est déterminée et une nouvelle unité est créée entre q et un de ses voisins dans le graphe. Les erreurs sont ensuite redistribuées et d'autres étapes d'adaptations sont effectuées. La topologie du réseau GNG reflète celle de la distribution des signaux d'entrée. Ces modèles ont la capacité d'apprendre en continu à partir de données mais n'assurent pas la propriété de stabilité.
Le modèle "Incremental Grid growing" (IGG) [BM93] est une amélioration du réseau SOM qui permet à la structure de la carte d'être dynamique en s'adaptant aux données (structure non fixée au préalable). En dimension 2, la grille est initialement constituée de quatre neurones connectés entre eux. Comme dans les versions évolutives précédentes, après un nombre λ d'étapes d'adaptation, des neurones sont ajoutés à la grille, mais cette fois au niveau du périmètre au voisinage du neurone ayant l'erreur cumulée maximale. Cette version ne nécessite pas un calcul préalable des dimensions de la carte, mais la même base est utilisée à chaque adaptation de la grille courante. L'aspect incrémental concerne seulement la topologie de la carte sans tenir compte de l'arrivée dynamique des données (pas de plasticité).
Certains des modèles que nous avons présentés permettent, certes, de prendre en compte l'arrivée dynamique des données mais ne garantissent pas la préservation des anciennes connaissances. Ceci peut poser des problèmes lors de leur utilisation à long terme pour traiter de gros volumes de données dans certaines applications où les connaissances déjà apprises peuvent être utiles pour traiter les nouvelles données. Pour réaliser un système qui permette d'acquérir en continu de nouvelles connaissances tout en continuant à mémoriser les anciennes, Carpenter et Grossberg [CG87] ont proposé la théorie de la résonance adaptative "ART : Adaptive Resonance Theory".
Un modèle ART est constitué de deux modules : le module d'attention "Attentional Subsystem" et le module d'orientation "Orienting Subsystem". Le module d'attention reçoit l'entrée et la compare aux poids des neurones existant et sélectionne le neurone gagnant. Le module d'orientation effectue le test de vigilance qui permet de décider si le neurone sélectionné est assez proche de l'entrée, de relancer une autre recherche dans le réseau si celui-ci n'est pas satisfait et de provoquer l'adaptation du neurone gagnant ou éventuellement la création de nouveaux neurones.
Le réseau ART1 est l'un des premiers réseaux ART conçu pour traiter des entrées binaires. Quand une nouvelle entrée ξ est présentée à la couche de reconnaissance, la valeur d'activation des neurones de la couche de comparaison est calculée en utilisant une fonction d'activation unidirectionnelle $\vec{F}(,)$, c'est une mesure de similarité non symétrique. Un apprentissage compétitif est ensuite effectué. Le neurone gagnant (éligible à être résonant) $s(\xi)$ est la solution, si elle existe, du problème de maximisation suivant :

$$s(\xi) = \arg\max_{c \in A} \vec{F}(\xi, w_c) \tag{1}$$

où A est l'ensemble des neurones et w_c le poids du neurone c.
soumis au test de vigilance décrit par :

$$\vec{G}(w_s, \xi) \geq \rho, \quad \rho \in [0, 1] \tag{2}$$

Où ρ est le seuil de vigilance défini par l'utilisateur et $\vec{G}(,)$ est une fonction de similarité unidirectionnelle (non symétrique).
Si le test de vigilance n'est pas satisfait (la résonance ne se produit pas), le neurone gagnant est inhibé et une recherche du second gagnant est effectuée, celui-ci est soumis à son tours au test de vigilance.
Si le test de vigilance est satisfait, l'activité du neurone gagnant est renforcée en ajustant son vecteur de poids pour le rapprocher de l'entrée. Si aucune solution n'existe pour le problème de maximisation décrit précédemment, un nouveau neurone est dynamiquement alloué pour la nouvelle entrée.
Dans ART1, le test de vigilance utilise la fonction $\vec{G}(,)$ tandis que le neurone éligible pour être résonant est recherché suivant la valeur de la fonction $\vec{F}(,)$. Donc, le vecteur de connexion du neurone gagnant en utilisant $\vec{F}(,)$ n'est pas nécessairement le gagnant en utilisant $\vec{G}(,)$, ceci justifie la recherche qui est répétée jusqu'à ce que le test de vigilance soit satisfait.

3 Notre approche : SOM-ART

Le modèle simplified ART "SART" [BA98] a été conçu pour faire face à certaines limites du modèle ART1 et permet de simplifier l'architecture des modèles ART. Ce modèle est capable de traiter des entrées multidimensionnelles à valeurs réelles. Étant donné un vecteur d'entrée à valeurs réelles $\xi \in R^n$ (aucun prétraitement n'est effectué), le neurone gagnant $s(\xi)$ est la solution, si elle existe, du problème de maximisation suivant :

$$s(\xi) = \arg\max_{c \in A} \overline{F}(\xi, w_c) \tag{3}$$

soumis à la contrainte de vigilance :

$$\overline{G}(w_s, \xi) \geq \rho, \quad \rho \in [0, 1] \tag{4}$$

où $\overline{F}(,)$ and $\overline{G}(,)$ sont des fonctions d'activation et de similarité (respectivement) symétriques et sont choisies de façon que :

$$\overline{G}(w_{c_1}, \xi) > \overline{G}(w_{c_2}, \xi) \Rightarrow \overline{F}(\xi, w_{c_1}) > \overline{F}(\xi, w_{c_2}) \tag{5}$$

et vice versa.
Le processus de recherche n'est pas nécessaire pour détecter le domaine de résonance : quand le vecteur de poids du neurone gagnant w_s, sélectionné selon

$\overline{F}(,)$, ne satisfait pas la contrainte de vigilance, un nouveau neurone peut être immédiatement alloué pour ξ. De plus, Comme les fonctions $\overline{F}(,)$ et $\overline{G}(,)$ sont symétriques, en prenant $\overline{F}(,) = \overline{G}(,)$ la propriété 5 est vérifiée.

Nous avons intégré une carte auto-organisatrice incrémentale dans un modèle Simplified ART. SOM-ART est une carte topologique de voisinage carré (même voisinage que dans IGG), mais les critères d'initialisation de la carte et d'insertion de neurones sont différents. Ceux-ci sont effectués suivant le paradigme ART. En plus de la notion d'auto-organisation qui existe dans les réseaux Simplified ART, SOM-ART permet de projeter des données de dimension quelconque dans un espace discret de façon que les données similaires aient des projections proches dans la carte. Ceci est l'une des particularité des cartes SOM qui facilite la visualisation des données et leur interprétation.

Le tableau 1 montre les propriétés de notre modèle SOM-ART en comparaison avec différentes cartes SOM ainsi que le réseau Simplified ART. le symbole '+' représente la présence de la propriété, '-' son absence et '~' sa présence dans certaines variantes du réseau.
La carte SOM-ART est d'abord initialisée à un neurone, ensuite des neurones

	SOM-ART	Simplified ART	SOM	SOM incrémentales en données
Stabilité	+	+	-	-
Plasticité	+	+	-	+
Classement	+	-	~	~
Classification	+	+	+	+
Visualisation	+	-	+	~

Table 1: *Comparaison des propriétés des différentes cartes SOM et du réseau Simplified ART.*

sont ajoutés suivant les critères du réseau Simplified ART en utilisant le test de vigilance pour sélectionner le neurone gagnant. Si le test de vigilance est satisfait, une mise à jour de la carte est effectuée. Sinon, un neurone est ajouté au périmètre de la carte comme dans IGG mais cette fois au voisinage du neurone le plus proche de l'entrée tout en respectant la structure de la carte (voisinage carré). Des connexions sont aussi ajoutées à la carte pour lier le neurone ajouté à ses voisins. Durant la dernière étape de l'apprentissage, les neurones de la carte sont étiquetés an utilisant un vote majoritaire sur l'ensemble des étiquettes des entrées qui les ont activés.
La distance euclidienne $dist(,)$ [BA98] est utilisée pour comparer les neurones. Les fonctions d'activation et de similarité, décrites précédemment, du réseau

SOM-ART sont obtenues à partir de la distance $dist(,)$ comme suit :

$$\overline{F}(,) = \overline{G}(,) = \frac{1}{1 + dist(,)} \tag{6}$$

Ces fonctions sont symétriques, donc vérifient la propriété (5) et peuvent être utilisées dans SOM-ART.
L'algorithme d'apprentissage de SOM-ART est décrit ci-dessous (voire algorithme 3.1).

Algorithm 3.1 Algorithme d'apprentissage de SOM-ART.

1. Initialiser l'ensemble A des neurones à une seule unité c_1.
   Initialiser le paramètre temps t : $t = 0$.
   Initialiser l'ensemble des connexions C, $C \subset A \times A$ à l'ensemble vide : $C = \emptyset$.

2. Présenter une entrée ξ
   Le vecteur référence w_{c1} de la première unité c_1 is initialisé a la première entrée.

3. Déterminer le gagnant $s(\xi) \in A$ par :

$$s(\xi) = \arg\min_{c \in A} dist(\xi, w_c)$$

4. Soumettre le neurone gagnant $s(\xi)$ au test de vigilance :
   - **Si** $(\frac{1}{1+dist(s(\xi),\xi)} \geq \rho)$, adapter chaque unité r selon :

$$\Delta w_r = \epsilon(t) h_{rs} [\xi - w_c]$$

   où $\epsilon(t) = \epsilon_i (\frac{\epsilon_f}{\epsilon_i})^{\frac{t}{t_{max}}}$, $h_{rs} = \exp(\frac{-d_1(r,s)^2}{2\sigma^2})$ et $\sigma(t) = \sigma_i (\frac{\sigma_f}{\sigma_i})^{\frac{t}{t_{max}}}$
   $\epsilon(t)$ est le pas d'apprentissage, ϵ_i et ϵ_f sont les valeurs initiale et finale de ϵ, σ_i et σ_f sont les valeurs initiale et finale σ, $d_1(,)$ est la distance de Manhattan et h_{rs} est la fonction de voisinage .
   - **Sinon** ajouter une nouvelle unité r dans le voisinage du neurone le plus proche de l'entrée dans le périmètre de la carte, son vecteur référence w_r est initialisé par : $w_r = \xi$.

5. Incrémenter le paramètre temps t : $t = t + 1$

6. Si $(t \leq t_{max})$, continuer à partir de l'étape 2.
   Si $(t = t_{max})$ étiqueter chaque unité en utilisant un vote majoritaire sur les étiquettes des entrées qui ont activé ces unités.

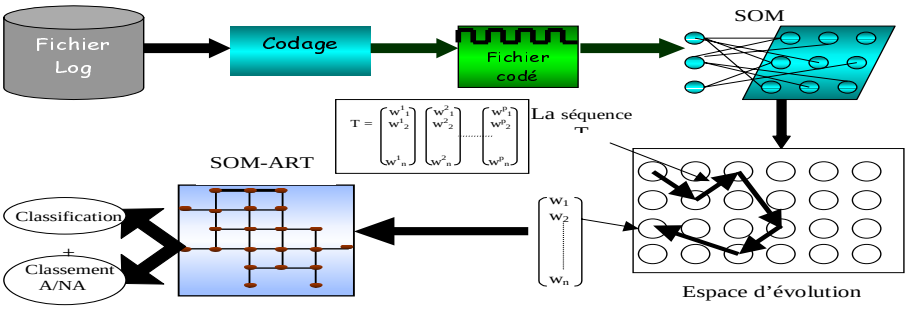


Figure 2: Les différents traitements effectuées sur les données.

4 Résultats de validation

Nous avons effectué plusieurs expérimentations sur des fichiers log d'un site de commerce électronique. Plus précisément, le comportement de chaque utilisateur est décrit par des informations sur la succession de pages qu'il a parcourues dans le temps. Cette succession de pages représente l'aspect temporel des données. De plus, le site de e-commerce est consulté chaque jour par environ 3000 personnes, par conséquent, nous avons un grand volume de données qui sont dynamiques puisqu'elles peuvent constamment changer. Comme notre travail est réalisé pour pouvoir effectuer des actions durant la navigation de l'internaute, le traitement des données doit être effectuées avant qu'il ne quitte le site, ce qui impose des contraintes temps réel. Dans notre application, nous ne possédons pas de connaissances à priori du domaine et les données contiennent du bruit.
Comme l'utilisation directe des données brutes n'est pas possible, plusieurs traitements ont étés effectués avant d'utiliser la carte SOM-ART (voire la figure 4). Les données sont d'abord codées en utilisant des matrices quasi-comportementales [ZBB03] après différents filtrages qui ont permis de supprimer une partie du bruit. Le principe de ce codage est de calculer pour chaque page sa fréquence de précédence et de succession à travers toutes les autres pages et de regrouper ces fréquences en une matrice. Ces données sont ensuite traitées par une carte SOM pour définir un espace d'évolution dans la carte. Les pages sont alors représentées par les vecteurs poids des neurones de la carte. Chaque vecteur possède 50 composantes. Les navigations représentent des successions de ces vecteurs (correspondant aux successions de pages) de longueurs variables. Une classification des navigations suivant les comportements des internautes ainsi qu'un classement suivant l'une des deux classes {acheteur, non acheteur } sont effectués en utilisant la carte SOM-ART. Dans les expérimentations, nous avons effectué un classement pour chaque état d'une navigation [1] (fenêtre temporelle

[1] Nous travaillons actuellement sur une extension de SOM-ART pour prendre en compte l'aspect temporel des données en représentant chaque navigation par une matrice de covariance dynamique.

de longueur 1).

Pour évaluer notre approche, nous avons utilisé les mesures suivantes :

- *Le résultat global :* représente le taux de classements corrects sur le nombre de tous les classements. Ce résultat est donné sous forme d'un intervalle de confiance à 95%.
- *La matrice de confusion*
- *La sensibilité :* représente le taux de bons classements de la classe "acheteurs".
- *La spécificité :* représente le taux de classements corrects de la classe "non-acheteurs".
- *L'espace ROC (Receiver Operating Characteristics) :* est une représentation graphique du compromis entre les taux de classements corrects des acheteurs et le taux de classements incorrects des non-acheteurs. L'axe des abscisses représente 1- spécificité et l'axe des ordonnées représente la stabilité. Chaque classifieur est représenté par un point de coordonnées (sensibilité, 1 - spécificité). Le meilleur classifieur est celui qui se trouve au nord ouest (0,1).
- *Le nombre de neurones de la couche de sortie :* Ce nombre a une influence directe le temps de traitement des données. Plus le nombre de neurones est petit, plus le temps de traitement est court.

Résultats

Nous avons, d'abord, calculé les résultats obtenus par SOM-ART en faisant varier le seuil de vigilance et ceux obtenus par SOM en utilisant une base d'apprentissage qui contient 7000 navigations et une base de test qui contient 3000 navigations. Les navigations sont de longueurs variables. Ceci permet de montrer l'incrémentalité de SOM-ART en structure.

	SOM-ART				SOM
Seuil de vigilance	***0.7***	***0.8***	***0.9***	***0.95***	--
Nombre de neurones	***155***	***200***	***302***	***357***	***1026***
Résultat global	[36.70%, 38.11%]	[56,10%, 57.55%]	[62.98%, 64.38%]	[84.51%, 85.55%]	[84.51%, 85.55%]
Matrice de confusion	3945 11068 226 2807	2788 6408 1383 7467	2447 4828 1724 9047	1733 261 2438 13614	1733 261 2438 13614
Sensibilité	***0.948***	***0.668***	***0.586***	***0.415***	***0.415***
Spécificité	***0.202***	***0.538***	***0.652***	***0.981***	***0.981***

Figure 3: *Comparaison de résultats obtenus pour le classement des états des séquences en utilisant SOM et SOM-ART.*

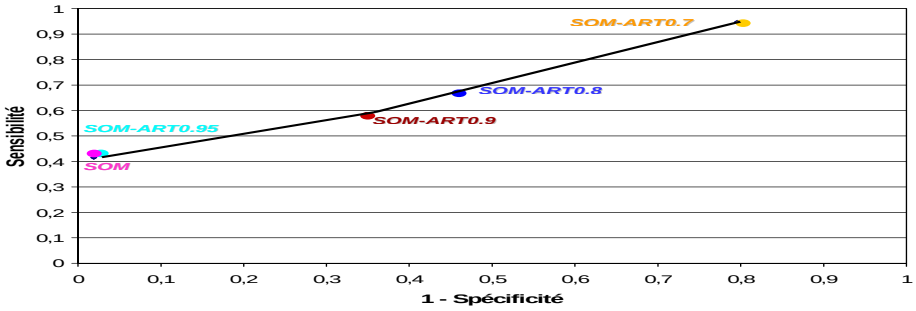


Figure 4: *Espace ROC : Comparaison de résultats obtenus pour le classement des états des séquences en utilisant SOM et SOM-ART.*

	Résultat global	Matrice de confusion	Sensibilité	Spécificité
SOM-ART	[84.51%,85.55%]	$\begin{pmatrix} 1733 & 2438 \\ 261 & 13614 \end{pmatrix}$	0.415	0.981
SOM	[82.85%,83.93%]	$\begin{pmatrix} 1176 & 00 \\ 2995 & 13875 \end{pmatrix}$	0.281	1

Table 2: *Comparaison des résultats obtenus par SOM et SOM-ART après la partition de la base initiale.*

Les tests montrent que le nombre de neurones augmente avec l'augmentation du seuil de vigilance dans le réseau SOM-ART ainsi que le résultat global. Celui-ci se stabilise et devient égal à celui de SOM pour un seuil de vigilance égal à 0.95 (voire le tableau de la figure 3).
La sensibilité diminue avec l'augmentation du seuil de vigilance tandis que la spécificité augmente. Dans l'espace ROC, nous remarquons que la carte SOM ainsi que SOM-ART pour un seuil de vigilance égal à 0.95 sont les plus proches du nord ouest (voire figure 4), donc fournissent les meilleurs résultats de classement. De plus, le nombre de neurones de SOM-ART est inférieur à celui de SOM, donc le temps mis par SOM-ART pour traiter les données est inférieur à celui mis par SOM.

Pour monter l'incrémentalité de SOM-ART en données (plasticité et stabilité), nous avons partitionné la base d'apprentissage en quatre parties et avons effectué des apprentissages successifs sur les différentes parties (voire tableau 2).

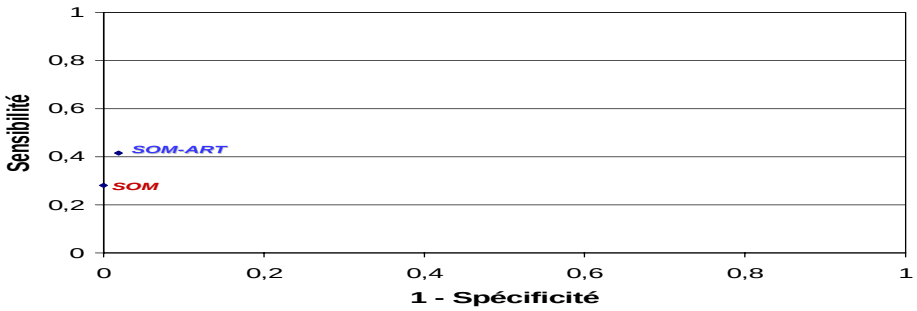


Figure 5: *Espace ROC : Comparaison des résultats obtenus par SOM et SOM-ART après la partition de la base initiale.*

Nous remarquons que les résultats de classement obtenus par la carte SOM se détériorent contrairement à ceux obtenus par SOM-ART pour un seuil de vigilance égal à 0.95. Ceux-ci ne changent pas comparés à ceux obtenus en utilisant la base d'apprentissage entière. Ceci montre, en plus de la possibilité d'apprentissage de nouvelles connaissances de SOM-ART, sa capacité à préserver les anciennes connaissances déjà acquises puisque les résultats obtenus sont les mêmes que ceux obtenus en faisant l'apprentissage avec l'intégralité de la base d'apprentissage.

La figure 5 montre la position des deux modèles dans l'espace ROC, et nous remarquons que SOM-ART est plus proche du nord ouest (c'est à dire meilleure que SOM).
Les tests que nous avons effectués montrent qu'avec SOM-ART, on a pu obtenir les mêmes résultats de classement tout en ayant moins de neurones, de plus notre approche possède les propriétés de stabilité et de plasticité.

5 Conclusion et perspectives

Nous avons présenté une nouvelle carte SOM évolutive qui est capable d'acquérir de façon continue les nouvelles connaissances tout en préservant les anciennes. Pour compléter la validation de notre approche, nous envisageons d'effectuer d'autres comparaisons de SOM-ART avec des cartes SOM évolutives en utilisant d'autres données et en comparant aussi les résultats de classification et ceux liés à la visualisation (erreur topologique). De plus, pour notre application réelle qui consiste à classer des séquences (les navigations), une approche qui traite des données temporelles s'impose. Pour ceci, nous nous intéressons, en ce moment, à construire une extension de SOM-ART qui prend compte l'aspect temporel des données. Cette extension consiste à modéliser chaque séquence (ou navigation) sous forme d'une matrice de covariance dynamique. De plus des modifications

de la carte SOM-ART seront effectuées pour traiter des matrices et non des vecteurs. Ces modifications concernent la distance matricielle utilisée qui doit être adéquate pour comparer deux matrices de covariances (comme par exemple, des distances basées sur des tests de sphéricité) ainsi que la formule d'adaptation.

References

[BA98] A. Baraldi and E. Alpaydin. Simplified ART: A new class of ART algorithms. Technical Report TR-98-004, Berkeley, CA, 1998.

[BM93] J. Blackmore and R. Miikkulainen. Incremen- tal grid growing: Encoding high-dimensional structure into a two-dimensional feature map. In *in Proceedings of the IEEE International Con- ference on Neural Networks (ICNN'93)*, volume 1, pages 450–455, San Francisco, CA, USA, 1993.

[CG87] G.A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing*, 37(1):54–115, janvier 1987.

[CS88] G.A. Carpenter and S.Grossberg. The art of adaptive pattern recognition by a self-organizing neural network. *IEEE Computer*, 21(3):77–88, march 1988.

[Fri93] B. Fritzke. Growing cell structures – a self-organizing network for unsupervised and supervised learning. TR-93-026, International Computer Science Institute, Berkeley, 1993.

[Fri95] B. Fritzke. A growing neural gas network learns topologies. In In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 625–632, Cambridge MA, 1995.

[Koh95] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[ZBB03] A. Zeboulon, Y. Bennani, and K. Benabdeslem. Hybrid connectionist approach for knowledge discovery from web navigation patterns. In *In the proceedings of ACS/IEEE International Conference on Computer Systems and Applications*, Tunisia, July 2003.