DATA MINING SPATIAL UN PROBLEME DE DATA MINING MULTI-TABLES

CHELGHOUM Nadjim [*]***, ZEITOUNI*** [*]***Karine*** [*]

Institut Français de Recherche pour l'Exploitation de la Mer (IFREMER)
Laboratoire Environnement- Ressources du Languedoc Roussillon (LER/LR),
BP 171, Boulevard Jean Monnet
34203 Sète Cedex, France
Tel/Fax (0)4 99 57 32 83/ (0)4 99 57 32 96
nadjim.chelghoum@ifremer.fr

*** Laboratoire PRiSM, Université de Versailles*
45, avenue des Etats-Unis, 78035 Versailles Cedex, France
Tel / Fax: (0)1 39 25 40 46 / (0)1 39 25 40 57
karine.zeitouni@prism.uvsq.fr

1. Introduction

Le data mining spatial (DMS) consiste à appliquer le data mining aux données spatiales [23], [11], [20]. Contrairement aux données classiques, ces données sont de nature dépendantes [2], [15] car tout phénomène spatial est influencé par son voisinage. Par exemple, la contamination des moules dans une lagune est influencée par les champs d'agriculture autour. Cette notion de dépendance entre les données est justifiée par une loi qui stipule que « *ce qui se passe dans une localité particulière dépend de ce qui se passe dans d'autres localités et ces interactions sont d'autant plus fortes que les localités concernées sont plus proches* » [22]. De point de vue analyse de données, ceci revient à dire que l'analyse spatiale nécessite l'analyse des entités spatiales en fonction de leurs caractéristiques, les caractéristiques de leurs voisins et les relations spatiales[1]. Analyser ces données sans tenir compte de cette spécificité génère des résultats erronés et incorrects [1]. Prendre en compte cette spécificité nous pose deux problèmes. Le premier est lié à l'inadaptation des méthodes existantes de data mining à ce type d'analyse. Le deuxième est dû à la complexité de calcul des relations spatiales.

1.1 Dépendance entre les données spatiales

Aucune méthode d'analyse de données ne permet d'analyser les données dépendantes. Ni la statistique classique, ni les méthodes du data mining traditionnel ne permettent

[1] Liens de voisinage qui relient les objets spatiaux. Ils peuvent être métriques, topologique ou directionnelles [6]

d'analyser les données spatiales. Toutes supposent que les données sont indépendantes [13]. De plus, aucune de ces méthodes ne peut interpréter les relations spatiales implicites qui relient les objets spatiaux. Toutes traitent des données simples de type numérique ou chaîne de caractère. Certes, des travaux en SIG[2] [14] et en statistique dite spatiale [4], [19], [21] ont abordé ce problème, mais ils restent encore limités dans ce type d'analyse. En effet, les SIG, même s'ils permettent d'interroger les données spatiales et de répondre à un certain calcul (ex : trouver le nombre de maison dans une ville), ont des capacités d'analyse limitées et ne permettent pas de découvrir des modèles, ni des règles ou de nouvelles connaissances cachées dans les bases de données spatiales. La statistique spatiale, même si elle est largement répandue pour l'analyse spatiale et offre un grand nombre de techniques allant de la géostatistique à l'analyse globale et locale d'autocorrélation ou l'analyse de données multi-variées, elle reste le plus souvent confirmatoire, guidée par un expert, basée sur des données numériques et ne découvre pas de règles. De plus, l'analyse exploratoire de données multi-variées, sous contrainte de contiguïté, présente l'inconvénient de ne considérer que les relations entre objets d'une même table excluant les relations spatiales pouvant exister entre objets de tables différentes [23].

1.2 Inadaptation des méthodes existantes de data mining

L'analyse des interactions dans l'espace, dites relations spatiales [6], est la principale spécificité de l'analyse des données spatiales car elles mettent en évidence l'influence de voisinage. Le problème est double. Le premier est que ces relations spatiales sont souvent implicites. Elles ne sont pas stockées dans les bases de données spatiales et à chaque fois qu'on a besoin d'eux, on fait appel à des calculs géométriques complexes et coûteux. Le deuxième problème est que les relations spatiales sont nombreuses, voir infinies (distance, inclusion, adjacence, …). Par conséquent, le choix de la ″bonne″ relation spatiale est difficile à faire. Toutes les méthodes existantes en data mining spatial se limitent à une relation spatiale choisie par un utilisateur métier [7], [12]. Ce choix devient difficile lorsque ces relations sont multiples. Il faut donc l'automatiser.

La section 2 présente l'approche proposée. La section 3 décrit l'application de cette approche aux arbres de décision spatiaux. Les expérimentations et les résultats obtenus sur l'analyse du risque d'accident routiers dans la région parisienne seront présentés dans la section 4, suivis par une discussion et une conclusion.

[2] Système d'Information Géographique

2. Approche proposée

Le data mining spatial utilise d'une manière intensive les relations spatiales car ces dernières mettent en évidence l'influence du voisinage. Ces relations sont à l'origine implicites. Zeitouni et al. [24] proposent de les rendre explicites en utilisant l'index de jointure spatiale. Leur idée était de pré- calculer la relation spatiale exacte entre les localisations de deux collections d'objets spatiaux et de la stocker dans une table secondaire (cf. Figure 1). Nous proposons d'exploiter cette structure et de l'intégrer dans le data mining spatial. Ceci nous permet de bénéficier de toutes les optimisations offertes par l'index de jointure spatiale, à savoir : (i) le stockage des relations spatiales évite de les recalculer pour chaque application, (ii) Un simple parcours de l'index de jointure spatiale permet de connaître toutes les relations spatiales existantes entre deux collections d'objets spatiaux, (iii) le même index permet de déduire d'autres relations spatiales. Un autre grand avantage de l'utilisation de l'index de jointure spatiale est qu'il ramène le data mining spatial au data mining multi-tables [5].

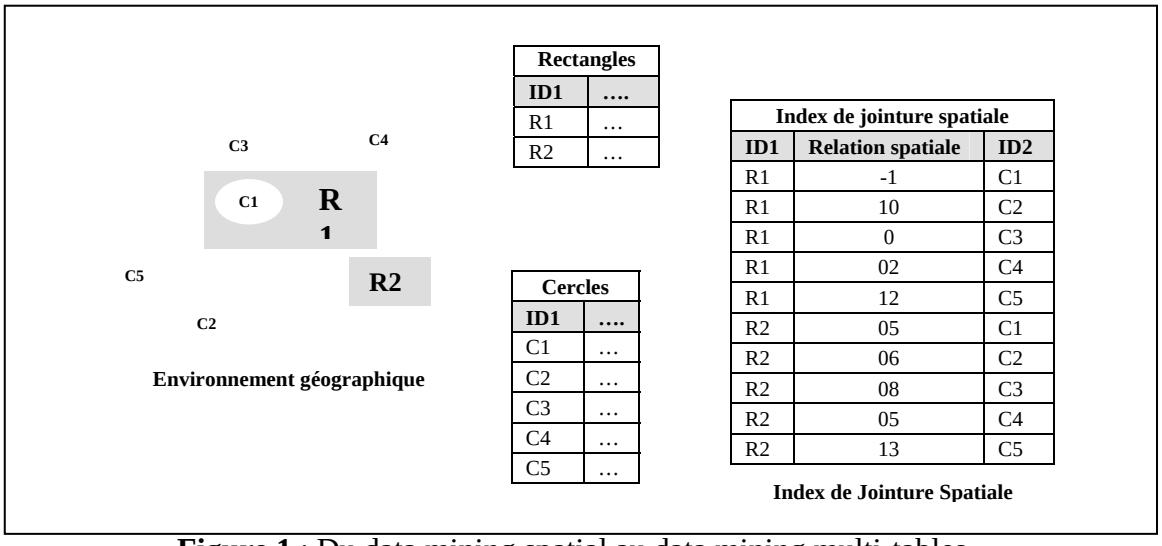
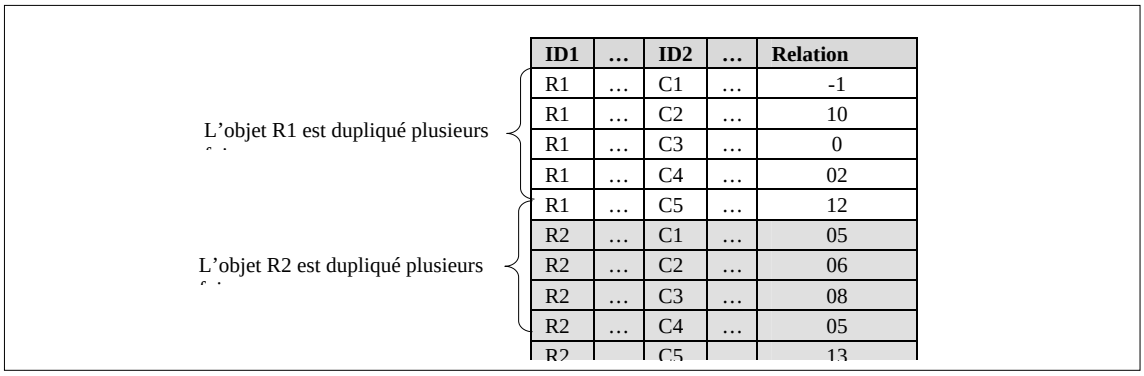


Figure 1 : Du data mining spatial au data mining multi-tables

Désormais, tout problème de data mining spatial peut être réduit à un problème de data mining multi-tables et l'utilisation des relations spatiales devient alors possible car elles seront vues par les méthodes d'analyse comme un attribut à analyser à l'instar des autres attributs. Ainsi, le choix de la bonne relation spatiale peut être fait automatiquement par les méthodes d'analyse répondant ainsi au deuxième problème cité précédemment. Or, cette organisation multi-tables des données ne peut pas être directement analysée par les

méthodes d'analyse de données car celles-ci considèrent que les données en entrée sont dans une table unique où chaque tuple constitue un individu à analyser et chaque colonne est une variable d'analyse. Il est possible de se ramener à une seule table en joignant les différentes tables initiales. Or, cette jointure peut dupliquer des tuples car les observations à analyser sont en liaison N-M avec les objets voisins (cf. Figure 2). Ceci fausse les résultats des méthodes de data mining en raison du multiple comptage de ces observations. Par exemple, l'objet rectangle R1 se retrouve dupliqué autant de fois qu'il existe d'objets voisins C_i. Le même objet sera compté plusieurs fois et risque d'être classé dans différentes classes si on applique un algorithme classique d'arbre décision et générera ainsi des règles non discriminantes.



ID1	…	ID2	…	Relation
R1	…	C1	…	-1
R1	…	C2	…	10
R1	…	C3	…	0
R1	…	C4	…	02
R1	…	C5	…	12
R2	…	C1	…	05
R2	…	C2	…	06
R2	…	C3	…	08
R2	…	C4	…	05
R2		C5		13

Figure 2 : Problème de la jointure entre les trois tables

Pour résoudre ce problème multi-tables, nous proposons de transformer, grâce à notre opérateur CROISEMENT, la structure multi-tables des données en une table unique de manière à ne pas dupliquer les observations et tout en gardant les informations sur le voisinage et les relations spatiales. Notre idée est de compléter, et non pas de joindre, la table d'analyse par des données présentes dans les autres tables. Cet opérateur est définit ci-dessous.

2.1 Définition de l'opérateur CROISEMENT

Le principe de notre opérateur «CROISEMENT» est de générer pour chaque valeur d'attribut de la table de voisinage un attribut dans la table résultat. Il est défini comme suit :

Soient R (ID1, A_1, …, A_n), V (ID2, B_1, …, B_m) et I (ID1, ID2, W) trois tables dont les clés sont soulignées, B_i sont des attributs qualitatifs et b_{ij} (j = 1, …K_i) sont les valeurs

distinctes de B_i et W désigne le poids (ou la relation spatiale dans la cas de DMS). Soit $F = \{F_1, F_2, \ldots, F_m\}$ un ensemble de fonctions d'agrégats.

0CROISEMENT (R, V, I, F) est une table T ayant le schéma suivant:

T (ID1, A_1, …, A_n, W_b_{11}, …, W_b_{1K1}, **……,** W_b_{m1}, …, W_b_{mKm}) de clé ID1 et où :

t = (id1, a_1, a_2, …, a_n, W_b_{11}, …, W_b_{1K1}, …… , W_b_{m1}, W_b_{m2}, …, W_b_{mKm}) ∈ T avec

- (id1, a_1, a_2, …, a_n) = $\sigma_{(ID1 = Id1)}$ (R)
- W_b_{ij} = F_i ($\sigma_{(ID1 = Id1)}$ (I) ∞ $\sigma_{(Bi = bij)}$ (V) ; W) si $\sigma_{(ID1 = Id1)}$ (I) est non vide, la valeur NULL sinon[3].

Exemple

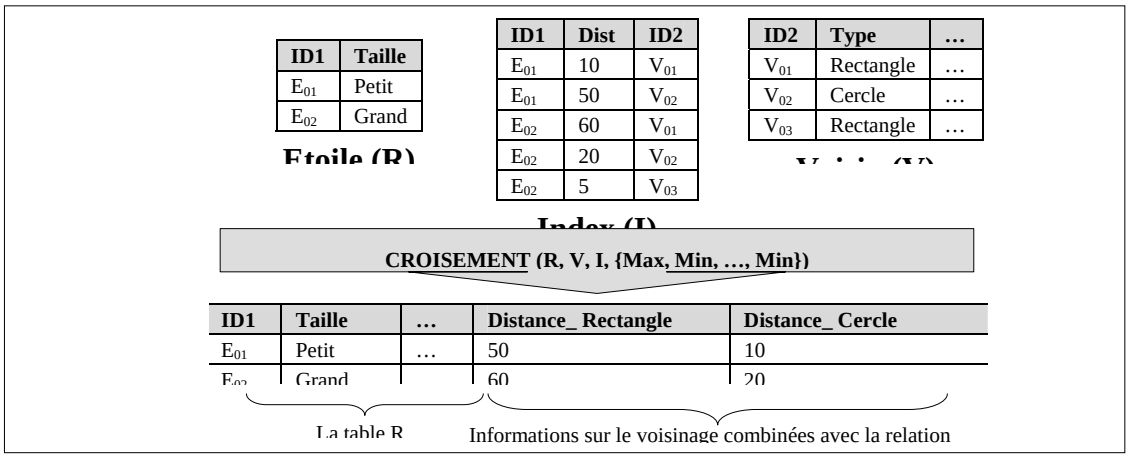


Figure 3: Exemple d'utilisation de l'opérateur CROISEMENT

En fait, I est une table de correspondances pondérées qui relie la table R de « faits » à analyser avec une table comprenant des « dimensions » à considérer. Cet opérateur n'est recommandé que lorsque les attributs B_i de V comportent assez peu de valeurs distinctes. La propriété de l'opérateur CROISEMENT est que le résultat comporte toujours comme partie gauche la table R dans son intégralité sans duplication de ses tuples et que celle-ci est complétée en partie droite par les poids des « dimensions » provenant de V et de I. Ce cas de figure arrive couramment dans les données relationnelles où la table de correspondance traduit les liens de cardinalité N-M. Cet opérateur peut être vu comme un moyen de préparation des données relationnelles pour le data mining.

[3] Les symboles σ et ∞ expriment respectivement la sélection et la jointure.

Faisons remarquer que pour un même tuple de R et une même valeur d'attribut b_{ij} de V, il peut y avoir plusieurs liens dans I avec des poids éventuellement différents. Comme la valeur W_b_{ij} du résultat T devait être unique, nous avons introduit des fonctions agrégats afin de calculer une seule valeur pour W_b_{ij}. En cas d'inexistence de ce lien, la valeur NULL remplace cette fonction. Les tuples, n'ayant pas de correspondant dans I, sont complétés par des valeurs NULL tout comme une jointure externe gauche.

2.2 Comparaison avec l'opérateur UNPIVOT d'Oracle

Récemment un opérateur UNPIVOT a été implémenté dans Oracle 9i dans le cadre du processus, dit ETL[4], d'un entrepôt de données [17]. Une autre forme d'opérateur UNPIVOT a été proposée dans [9].

Dans Oracle 9i, il s'agit de remplacer une colonne –dite colonne locatrice– par autant de colonnes que de valeurs dans la colonne locatrice. Typiquement, une table de ventes par magasin et par mois de schéma (magasin, mois, vente) peut être transformée en table de schéma (magasin, janvier, février, …, décembre) où les mois figurent en colonnes et où un tuple indique les montants de tous les mois pour le magasin. Cet opérateur peut être vu comme un outil d'adaptation de schéma en vu de l'intégration dans un entrepôt de données. Un opérateur inverse PIVOT est également proposé.

Dans [9], le but est d'optimiser les méthodes de classifications comme CART [3] dans le cas tout à fait classique. En effet, le recueil de statistiques nécessaires aux calculs d'entropies ou d'indices de gain, s'il est fait en SQL standard, entraîne plusieurs lectures de la table en entrée car les comptages considèrent un à un les attributs explicatifs. Grâce à l'utilisation d'un opérateur direct appelé UNPIVOT, ces comptages se font plus efficacement. En réalité, cet opérateur est curieusement équivalent à l'opérateur PIVOT d'Oracle car il transforme plusieurs colonnes en deux colonnes, une reportant le nom d'attribut et la seconde sa valeur. Après cette étape, une requête simple SQL donne les comptages utiles à la classification par groupage sur la combinaison (nom d'attribut, valeur d'attribut).

La différence entre l'opérateur CROISEMENT que nous avons proposé et l'opérateur UNPIVOT d'Oracle est que le premier, contrairement au second, prend en entrée trois tables et intègre les fonctions agrégats. Ce qui répond à notre problématique. L'expression de l'opérateur CROISEMENT par UNPIVOT d'Oracle est possible et

[4] Extraction, Transformation, Chargement

donnée par la formule ci-dessous[5]. Cependant, une implémentation directe de notre opérateur évite les multiples jointures dans le cas de plusieurs attributs ainsi que le stockage des résultats intermédiaires de UNPIVOT.

CROISEMENT (R,I,V,F) = R ∞ UNPIVOT (F_1 (I ∞ V, ID_1 , B_1; W)) ∞ UNPIVOT (F_2 (I ∞ V, ID_1 , B_2 ; W)) ∞ …∞ UNPIVOT (F_m (I ∞ V, ID_1 , B_m; W))

3. Application à la classification supervisée par arbre de décision spatial

Un arbre de décision est un modèle de data mining représentant une structure de connaissances composée d'une séquence de règles de décision. Il a pour but de trouver les attributs explicatifs et les critères précis sur ces attributs donnant le meilleur classement vis-à-vis d'un attribut à expliquer. L'arbre est construit par l'application successive de critères de subdivision sur une population d'apprentissage afin d'obtenir des sous-populations plus homogènes **Erreur ! Source du renvoi introuvable.**. Il existe diverses méthodes d'arbres de décision **Erreur ! Source du renvoi introuvable.**. Le critère de subdivision est déterminé au niveau de l'attribut comme dans ID3 [18] et au niveau d'une valeur d'attribut comme dans CART [3].

L'extension des arbres de décision au spatial se traduit par la prise en compte, non seulement des propriétés des objets à analyser, mais également des propriétés des objets voisins et des liens de voisinage. Les premiers travaux dans ce domaine ont été réalisés par [8]. Ils utilisent les arbres de décision pour classifier les images satellitaires afin de détecter les astres et les galaxies. Le défaut de cette classification est qu'elle n'utilise que des paramètres alphanumériques qui sont les attributs des images et n'exploite guère les données spatiales.

Une autre méthode a été proposée par Ester et al. [7]. Elle se base sur la méthode ID3. Elle utilise le concept de graphe[6] de voisinage pour représenter les relations de voisinage. Son avantage est qu'elle considère non seulement les propriétés des objets à classer, mais aussi les attributs des objets voisins et les relations spatiales. Son grand défaut est qu'elle ne garantit pas une segmentation correcte menant à des sous-populations non disjointes car les critères spatiaux ne sont pas discriminants. De plus,

[5] L'opérateur d'agrégat est noté F_i (table, attribut1_groupement, … ; attribut_calculé)

[6] Un graphe de voisinage est un graphe orienté G (N, E) dont N est l'ensemble des nœuds et E est l'ensemble des arcs. Chaque nœud « n_i » correspond à un objets spatial et chaque deux nœuds (n_i, n_k) sont reliés par un arc s'ils vérifient la relation de voisinage.

elle est limitée à une seule relation de voisinage et elle ne fait pas de distinction entre les thèmes, ce qui est essentiel dans les applications géographiques.

Koperski et al. [12] proposent une autre méthode de classification basée sur le fait que les objets spatiaux sont caractérisés par différents types d'informations. Hormis les attributs alphanumériques de l'objet, ils préconisent de considérer les prédicats et les fonctions spatiales ainsi que les valeurs non spatiales d'autres objets reliés par une relation spatiale à l'objet considéré. Leur idée est de généraliser les données et de transformer ensuite toute propriété "attribut = valeur" et toute relation spatiale en prédicat. Les avantages de cette méthode est qu'elle garantie une bonne classification et qu'elle classifie les objets spatiaux en fonction de leur caractéristiques, les caractéristiques de leurs voisins et les relations spatiales exprimées sous forme de prédicats. Les inconvénients sont d'une part, le coût du prétraitement induit par la transformation en prédicats et d'autre part, perte de l'information détaillée dûe à la généralisation des données et l'absence de choix dynamique de la distance dans les critères de discrimination car la relation spatiale est uniquement binaire.

En s'inspirant de la démarche proposée, nous proposons une méthode baptisée SCART[7] pour la construction d'arbre de décision spatial. C'est une extension de la méthode CART aux données spatiales. Elle est décrite ci-dessous et résumée dans la Figure **4**.

Description de l'algorithme SCART proposé

L'algorithme prend en entrée (i) la table cible qui contient les objets à classer, (ii) la table de voisinage, (iii) l'index de jointure spatial qui formalise les liens de voisinage, (iv) les attributs explicatifs qui peuvent provenir de la table cible ou de la table de voisinage, (v) l'attribut à expliquer qui peut provenir uniquement de la table cible et (vi) les conditions de saturation qui déterminent la poursuite ou non du développement de l'arbre. Afin d'éviter les multiples jointures et la duplication des objets d'analyse, l'algorithme propose de se ramener à une seule table (Etape 1. Figure **4**) en utilisant l'opérateur CROISEMENT défini précédemment. Ensuite, pour construire l'arbre, il applique la méthode CART (étape 2).

Dorénavant, la construction de l'arbre de décision spatial se base sur une seule table résultante de l'application de l'opérateur CROISEMENT sur la table cible, l'index et la table de voisinage. Chaque objet d'analyse est présenté par un seul tuple et chaque variable d'analyse est une colonne de cette table. Le principe de construction de l'arbre est typiquement le même que celui des algorithmes classiques. Initialement, tous les

[7] Spatial Classification And Regression Tree

objets à analyser appartiennent à la racine de l'arbre (étape 2.1). Au fur et à mesure du classement, les tuples de la table cible seront attribués aux feuilles courantes de l'arbre. Pour chaque attribut explicatif, on calcule le meilleur gain informationnel (étape 2.2) en appliquant la formule classique du gain informationnel sans la moindre modification. La valeur de l'attribut de segmentation est celle maximisant le gain informationnel de tout attribut confondu. Si la feuille courante n'est pas saturée (étape 2.3), on affecte les objets de la feuille courante aux fils gauche ou au fils droit selon qu'ils vérifient ou non la condition de segmentation. A noter que la saturation peut être déclarée par différents critères : seuil minimum d'occupation du nœud, une profondeur maximale de l'arbre ou une valeur plancher du gain informationnel. Si aucun de ces paramètres n'est donné, le découpage ne s'arrête que lorsque tous les objets du nœud sont affectés à la même classe. Le nœud est donc une feuille dite ″pure″. On réitère l'étape 2 pour le nœud suivant s'il existe. Sinon, l'algorithme s'arrête (étape 2.4). Les numéros des nœuds sont stockés dans une file. L'arbre est donc développé par niveau : on teste et on développe les nœuds du niveau N ensuite les nœuds du niveau N+1. L'arbre de décision généré est binaire. Les nœuds peuvent être déterminés par un code défini récursivement par les fonctions suivantes : n°_fils_gauche = 2 * n°_père et n°_fils_droit = 2 * n°_père +1. Le nœud racine a la valeur 1.

Les paramètres en entrée

- Table_Cible : table des objets à analyser
- Table_Voisin : table des objets de voisinage
- Index_Jointure_Spatial : contient les relations spatiales entre les objets
- Classe : la variable à expliquer. Elle appartient forcément à la table cible
- Variables_Explicatives : elles appartiennent soit à la table Table_Cible ou à la table Table_Voisin
- Conditions_Saturation : conditions qui arrêtent le développement de l'arbre ;

Paramètre en sortie

- Arbre de décision spatial binaire ;

Algorithme

Etape 1 : Appliquer l'opérateur CROISEMENT. *// On se ramène à une seule table*

Etape 2 : Appliquer la méthode CART

Etape 2.1 : Initialisation : nœud = 1 *//tous les objets d'analyse sont affectés à la racine*

Etape 2.2 : Pour chaque attribut explicatif *Att* faire

Pour chaque valeur *Att.Va* de l'attribut *Att* faire

Gain_Info_Valeur = gain informationnel en utilisant la formule de CART classique

FinPour

Gain_Info_Att = le meilleur gain informationnel pour l'attribut *Att*

FinPour

Meilleur_Gain_Info = le meilleur gain informationnel de tous les attribut explicatifs

Enregistrer le critère de segmentation correspondant aux Meilleur_Gain_Info *//Ce critère est celui qui donne le meilleur gain informationnel de tous les attributs explicatifs*

Etape 2.3 : Si le nœud courant n'est pas saturé alors

On divise le nœud courant en deux nœuds et on affecte les objets d'analyse aux fils gauche et droit selon qu'ils vérifient ou pas le critère de segmentation. Le fils gauche est le complément du fils droit. L'affectation est faite en respectant le codage suivant : n°_fils_gauche = 2 * n_père et n°_fils_droit = 2 * n_père +1.

FinSi

Etape 2.4 : Itérer les étapes 3 et 4 sur le nœud suivant *// Les numéros du nœud sont stockés dans une file*

L'algorithme s'arrête lorsque tous les nœuds sont saturés.

Fin de l'algorithme

Figure 4: L'algorithme SCART

4. Expérimentations et résultats

C'est dans le cas de l'analyse de l'accidentologie en sécurité routière que nous avons testé notre approche pour le data mining spatial. L'analyse part d'une base de données spatiales, fournie par LROP[8], comprenant des données sur les accidents de la route et de l'environnement géographique (bâtiments, voies, …). L'objectif est de construire un modèle prédictif en recherchant des correspondances entre les accidents et les autres couches comme le réseau ou le tissu urbain. Ceci revient à appliquer la classification par

[8] Laboratoire Régional de L'Ouest Parisien

arbre de décision en intégrant le caractère spatial des accidents et leur interaction avec l'environnement géographique.

Un exemple de résultat est donné dans la
Figure **5**. Il est obtenu en utilisant notre méthode SCART. Ici, on cherche à classer les segments de route en deux classes : « segment non dangereux» et « segment dangereux ». Les attributs explicatifs sont, soit liés aux sections de route où est localisé l'accident (ex : sens de circulation), soit liés à l'environnement urbain (ex : école, marché, administration, etc.) combinés avec la relation spatiale « distance ». Les critères de construction de l'arbre de décision spatial sont résumés dans le
Tableau **1**. L'arbre de décision spatial obtenu et sa matrice de confusion[9] sont présentés dans la
Figure **5**.

Paramètres en entrée			
Attributs explicatifs	- -P_Feux - Sens_Circulation - Distance_Administration - Distance_Centre_Commercial	- Distance_Ecole - Distance_Hôpital - Distance_Sport	
Classe	Dangerosité - Segment non dangereux (moins de 02 accidents) - Segment dangereux (plus de 02 accidents)		
Critères de saturation d'un noeud	Nombre min d'objet dans un nœud	Gain info min	Profondeur
	4	0.001	04

Tableau 1: Paramètres en entrée

Dans cet arbre, la première condition de segmentation est ″sens de circulation = double″. Cette variable provient de la table cible ″segment″. Le fils gauche de la racine correspond aux sections de route ayant un sens de circulation = ″double″ et le fils droit son complément. Le fils gauche est segmenté à son tour par une partie proche des écoles (distance_école ≤ 425m) et une partie loin des écoles (distance_école >425 m). A ce

[9] C'est une matrice à deux dimension dont l'élément (i,j) indique le nombre d'objets classés réellement dans la base de données dans la classe i et observés dans la classe j. En d'autres termes, elle donne le taux des objets bien classés par l'arbre par rapport aux classes réellement observées.

niveau, la condition de segmentation est une combinaison de la valeur ″école″ d'un attribut de la table voisin, de la relation spatiale ″distance″, le comparateur ″<=″ et la valeur de la relation spatiale ″425m″. Ceci répond à notre problématique.

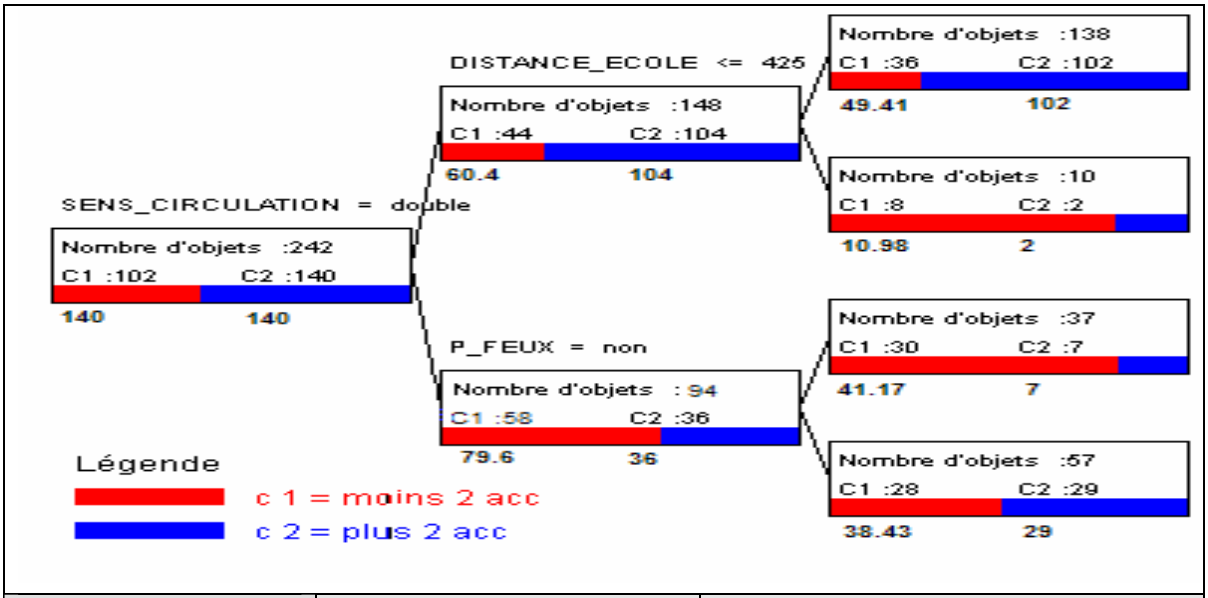
L'ensemble des règles de décision obtenues est :

(i) La première règle est issue de la feuille 1 à partir du haut. Elle stipule qu'il y a plus de segments de route dangereux près des écoles (distance ≤ 425 m) et où le sens de circulation est double (*coef = 67.37%*).

(ii) Lorsque le sens de circulation est double et qu'on est loin des écoles (distance > 425 m) alors il y a plus de segments de route non dangereux (*coef = 84.59%*). Cette règle correspond à la feuille 2.

(iii) La règle issue de la feuille 3 stipule que si le sens de circulation est unique et qu'il n'y a pas un feu rouge alors on a une section de route non dangereuses (*coef = 85.46%*).

(iv) On a plus de sections de route non dangereuses lorsque le sens de circulation est unique et qu'il y a un feu rouge (feuille 4) (*coef = 57%*).



	Segment non dangereux		Segment dangereux	
	Nombre	%	Nombre	%
Segment non dangereux	66	**64.7%**	36	35.3%
Segment dangereux	38	27.15%	102	**72.8%**

Figure 5: Arbre de décision spatial et sa matrice de confusion

On remarque, dans cet arbre de décision spatial, qu'on a plus de sections de route dangereuses lorsque le sens de circulation est double et qu'on est prés des écoles (feuille 1). Ceci nous laisse penser que les écoles sont des générateurs d'accidents lorsque le sens de circulation est double. On peut également remarquer que lorsque le sens de circulation est unique et qu'il n'y a pas de feu rouge, on a moins de sections de route dangereuses.

La matrice de confusion de cet arbre montre que 72.85% de segment de route dangereux et 64.7% de segment de route non dangereux sont bien classés par l'arbre. Ceci donne une crédibilité importante pour les règles extraites de cet arbre.

5. Conclusion

Le data mining spatial est une branche de data mining. Sa principale spécificité est qu'elle intègre, lors de l'analyse, les relations spatiales. Pour la mise en œuvre de ses méthodes, nous avons proposé une démarche en deux étapes. La première consiste à matérialiser les relations spatiales et de les stocker dans les index de jointure spatiale ramenant ainsi le data mining spatial au data mining multi-tables. La deuxième étape consiste à réorganiser, grâce à l'opérateur CROISEMENT proposé, les données multi-tables dans une table unique et appliquer ensuite un algorithme classique de data mining.

L'application de cette démarche à la méthode d'arbre de décision spatial a été décrite dans cet article. Une méthode, baptisée SCART, a été proposée. Contrairement aux méthodes classiques d'arbres de décision, qui prennent en entrée une seule table dont les tuples sont considérés comme des objets à classer, SCART dépasse cette limite et étend ces dernières à des données multi-tables. Le critère de division d'un nœud dans SCART se base sur le test d'existence d'un objet voisin vérifiant une telle condition et en relation spatiale R avec l'objet à classer. L'originalité de cette méthode est qu'elle nous permet de prendre en compte l'organisation en couches thématiques et les relations spatiales propres aux données spatiales. En effet, d'un coté, SCART classe les objets spatiaux selon à la fois leurs attributs, les attributs de leurs voisins et les relations spatiales. D'un autre côté, contrairement aux méthodes existantes, elle effectue un choix automatique de la "bonne" relation de voisinage. En outre, l'organisation en couches thématiques des données spatiales est tout à fait intégrée.

Cette méthode a été mise en œuvre et implémentée sous un environnement Orale 9i et Java. Des tests sur des données réelles relatives à l'analyse du risque d'accidents routiers ont été réalisés. Les résultats obtenus avec le prototype implémenté confirment l'efficacité de notre approche.

Des pistes pour faire évoluer les performances de cette méthode sont envisagées. Hormis les techniques d'optimisation déjà utilisées, des versions orientées disque des méthodes de construction d'arbres de décision telles que [9], [10], [16] et [17] peuvent améliorer le comportement de notre méthode face à de gros volumes de données.

References :

[1] Anselin L, Griffith D. A., Do spatial effects really matter in regression analysis? Papers, Regional Science Association 65, p11-34. (1988).

[2] Anselin, L. What is special about spatial data? Alternative perspectives on spatial data analysis. Technical paper 89-4. Santa Barbara, NCGIA 1989.

[3] Breiman L., Friedman J.H., Olshen R.A., Stone C.J. (1984), Classification and Regression Trees. Ed: Wadsworth & Brooks. Montery, California, 1984.

[4] Cressie N.A.C, Statistics for spatial data, Edition Wiley, New York, 1993.

[5] Dzeroski S., Lavrac N., (2001), "Relational Data Mining", Springer, 2001.

[6] Egenhofer M. J.: “Reasoning about Binary Topological Relations”, Proc. 2nd Int. Symp. on Large Spatial Databases, Zurich, Switzerland, 1991, pp. 143-160.

[7] Ester M., Kriegel H.P., Sander J. (1997) Spatial Data Mining: A Database Approach, In proceedings of 5th Symposium on Spatial Databases, Berlin, Germany, 1997.

[8] Fayyad Usama M., Djorgovski S. G., Weir N., Automating the analysis and cataloguing of sky surveys. In "Advances in Knowledge Discovery and Data Mining", AAAI Press / MIT Press, 1996 (Fayyad Usama M., Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy).

[9] Graefe G., Fayyad U., Chaudhuri S., On the efficient gathering of sufficient statistics for classification of large SQL databases, In Proceedings of the Fourth International Conference on Knowledge Discovery and Data-Mining (KDD 98), AAAI Press, New York City, August 27-31, 1998.

[10] Gherke J., Ramakrishnan R., Ganti V., "RainForest- A Framework for Fast Decision Tree Construction on Large Datasets", In proceedings of the 24 the Annual International Conference on Very Large Data Bases (VLDB), pp 416 - 427, New York, 1998.

[11] Han J., Kamber M., Data Mining. Concepts and Techniques, Academic Press Edition, 2001.

[12] Koperski K., Han J., Stefanovic N., An Efficient Two-Step Method for Classification of Spatial Data, In proceedings of International Symposium on Spatial Data Handling (SDH'98), p. 45-54, Vancouver, Canada, July 1998.

[13] Koperski K., Adhikary J., Han J., " Knowledge Discovery in Spatial Databases: Progress and Challenges", Proc. SIGMOD Workshop on Reasearch Issues in Data Mining and Knowledge Discovery Technical Report 96-08, University of British Columboa, Vancouver, Canada, 1996.

[14] Laurini R., Thompson D., Fundamentals of Spatial Information Systems, Academic Press, London, UK, 680 p, 3rd printing,1994.

[15] Longley P. A., Goodchild M. F., Maguire D. J., Rhind D. W., Geographical Information Systems - Principles and Technical Issues, John Wiley & Sons, Inc., Second Edition, 1999.

[16] Mehta M., Agrawal R., Rissanen J. "SLIQ: A Fast Scalable Classifier for Data Mining", In Proc. of Int. Conf. On Extending Database Technology (EDBT'96), Avignon, France, March 25-29, pp 18-32, 1996.

[17] Oracle9i Warehouse Builder Transformation Guide, Release 2 (9.0.4) for Windows and UNIX, Part No. B10658-01, February 2003, Oracle Corporation.

[18] Quinlan J.R. Induction of Decision Trees, Machine Learning (1), pp 82 - 106,1986.

[19] Sanders L., L'analyse statistique des données en géographie, GIP Reclus, 1989.

[20] Shashi S., Sanjay C. Spatial Databases: A Tour, Prentice Hall, 2003.

[21] Shaw G., Wheeler D., Statistical Techniques in Geographical Analysis, Edition David Fulton, London, 1994.

[22] Tobler W. R., Cellular geography, In Gale S. Olsson G. (eds) Phylosophy in Geography, Dortrecht, Reidel, p.379-86, 1979.

[23] Zeitouni K., Yeh L., Le data mining spatial et les bases de données spatiales, Revue internationale de géomatique, Numéro spécial sur le Data mining spatial, Vol 9, N° 4, 2000.

[24] Zeitouni K., Yeh L., Aufaure M-A., "Join indices as a tool for spatial data mining", Int. Workshop on Temporal, Spatial and Spatio-Temporal Data Mining, Lecture Notes in Artificial Intelligence n° 2007, Springer, pp 102-114, Lyon, France, September 12-16, 2000.

Zighed A., Ricco R., "Graphes d'induction - Apprentissage et Data Mining", Edition Hermès Sciences, 2000