MAMADOU OUATTARA

FOUILLE DE DONNÉES : VERS UNE NOUVELLE APPROCHE INTÉGRANT DE FAÇON COHÉRENTE ET TRANSPARENTE LA COMPOSANTE SPATIALE

Mémoire présenté
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de Maîtrise en sciences géomatiques
pour l'obtention du grade de maître ès science (M.SC.)

DÉPARTEMENT DES SCIENCES GÉOMATIQUES
FACULTÉ DE FORESTERIE, DE GÉOGRAPHIE ET DE GÉOMATIQUE
UNIVERSITÉ LAVAL
QUÉBEC

2010

Résumé

Depuis quelques décennies, on assiste à une présence de plus en plus accrue de l'information géo-spatiale au sein des organisations. Cela a eu pour conséquence un stockage massif d'informations de ce type. Ce phénomène, combiné au potentiel d'informations que renferment ces données, on fait naître le besoin d'en apprendre davantage sur elles, de les utiliser à des fins d'extraction de connaissances qui puissent servir de support au processus de décision de l'entreprise.

Pour cela, plusieurs approches ont été envisagées dont premièrement la mise à contribution des outils de fouille de données « traditionnelle ». Mais face à la particularité de l'information géo-spatiale, cette approche s'est soldée par un échec. De cela, est apparue la nécessité d'ériger le processus d'extraction de connaissances à partir de données géographiques en un domaine à part entière : le Geographic Knowlegde Discovery (GKD).

La réponse à cette problématique, par le GKD, s'est traduite par la mise en œuvre d'approches qu'on peut catégoriser en deux grandes catégories: les approches dites de prétraitement et celles de traitement dynamique de l'information spatiale.

Pour faire face aux limites de ces méthodes et outils nous proposons une nouvelle approche intégrée qui exploite l'existant en matière de fouille de données « traditionnelle ». Cette approche, à cheval entre les deux précédentes vise comme objectif principal, le support du type géo-spatial à toutes les étapes du processus de fouille de données. Pour cela, cette approche s'attachera à exploiter les relations usuelles que les entités géo-spatiales entretiennent entre elles. Un cadre viendra par la suite décrire comment cette approche supporte la composante spatiale en mettant à contribution des bibliothèques de traitement de la donnée géo-spatiale et les outils de fouille « traditionnelle »

Abstract

In recent decades, geospatial data has been more and more present within our organization. This has resulted in massive storage of such information and this, combined with the learning potential of such information, gives birth to the need to learn from these data, to extract knowledge that can be useful in supporting decision-making process.

For this purpose, several approaches have been proposed. Among this, the first has been to deal with existing data mining tools in order to extract any knowledge of such data. But due to a specificity of geospatial information, this approach failed. From this arose the need to erect the process of extracting knowledge from geospatial data in its own right; this lead to Geographic Knowledge Discovery.

The answer to this problem, by GKD, is reflected in the implementation of approaches that can be categorized into two: the so-called pre-processing approaches and the dynamic treatment of spatial relationships. Given the limitations of these approaches we propose a new approach that exploits the existing data mining tools. This approach can be seen as a compromise of the two previous. It main objective is to support geospatial data type during all steps of data mining process.

To do this, the proposed approach will exploit the usual relationships that geo-spatial entities share each other. A framework will then describe how this approach supports the spatial component involving geo-spatial libraries and "traditional" data mining tools

Avant –propos

À ma princesse de Thyou

qui est mon inspiration et qui a toujours su m'encourager.

À mes parents

qui m'ont toujours encouragés et soutenus dans mes activités

Remerciements

Dans la présente, je tiens à remercier tous ceux qui ont contribués d'une manière ou d'une autre à la réalisation de ce travail de recherche.

En premier lieu, mes remerciements vont à l'Institut Canadien de Développement International (ACDI) par le biais du Programme Canadien de Bourses de la Francophonie qui m'a offert une bourse d'études afin que je vienne parfaire mes connaissances au sein d'une université canadienne, québécoise. Grand merci au PCBF et à travers lui à la gestionnaire du programme Jeanne Gallagher.

En second lieu, je tiens à remercier tout particulièrement mon directeur de recherche le Docteur Thierry Badard, professeur agrégé dans le département de sciences Géomatiques de la faculté de foresterie, de géographie et de géomatique. Je tiens à te dire M'sieur que c'était un honneur que d'avoir été « coaché » par toi. Malgré tes multiples occupations, tu as été d'une clairvoyance et d'une ouverture d'esprit qui m'ont été d'un grand secours dans l'avancement de mes travaux de recherche. Les discussion menées avec toi m'ont toujours été fructueuses et m'ont fait aller de l'avant. J'ai particulièrement apprécié cette collaboration et j'espère qu'elle ira au-delà de la présente étude de maîtrise.

Mes remerciements vont également à l'endroit des membres du groupe de recherche GEOSOA : À Belko Diallo mon compatriote et frère. On est dans le même labo, dans des bureaux côte à côte, mais on ne se voit presque jamais. À Jean Mathieu mon co-maitrisard. Merci pour ces moments passés ensemble et surtout pour m'avoir initié à la résolution du Rubik-Cube. À Etienne Dubé pour m'avoir éclairé à mes débuts sur le domaine géo-spatiale, les logiciels GeoKettle, GeoMondrian et autre.

J'aimerais adresser toute ma gratitude également aux professeurs Hervé Martin du Laboratoire d'Informatique de Grenoble (LIG), Frédéric Hubert du département des sciences géomatiques de l'Université Laval pour avoir accepté de corriger ce mémoire et surtout pour leurs remarques pertinentes qui m'ont aidé à parfaire ce document.

Enfin, je tiens à remercier mes collègues étudiants de l'association des gradués en géomatique pour les soirées ciné, partys à la RedHouse et autres activités. Un merci à toi : Vincent Thomas, Ève Grenier, Mojgan Jadidi et Kiarash, Elodie, Eric Jansens-Coron, Pom, Mathieu Plante, Tania, Eugenie, Albortz…et tous ceux que je n'ai pu citer.

Un merci spécial à Carmen Couture, Danielle Goulet et Marie-Claude toutes du département des sciences géomatiques.

Tables des matières

Liste des figures

Chapitre 1 – Mise en contexte

1.1. Introduction

Bien avant l'essor que connait actuellement le domaine des technologies de l'information et de la communication, la problématique d'apprendre de nos données, a été de tout temps posée. De part le passé, des méthodes plus « traditionnelles » étaient mises en œuvre afin d'extraire de la connaissance au sein des données stockées, à des fin de décision. L'évolution des technologies tant en matière de stockage que de traitement de l'information, est venue rendre cette tâche d'extraction de connaissances davantage plus ardue *(Fayyad, et al., 1996) (Koperski, 1999) (Kantardzic, 2003) (Klösgen, et al., 2002)*. En effet, on assiste non seulement à une croissance exponentielle du volume d'informations stockées au sein de nos organisations mais également à une complexification de ces données *(Frawley, et al., 1992) (Tan, et al., 2006)*. De ce contexte est apparue la nécessité de mettre en œuvre des outils (techniques et méthodes) qui permettront de guider le processus et d'arriver à une extraction efficace de connaissances au sein d'entrepôts de données *(Chen, et al., 1996)*. L'ensemble de ces outils et méthodes ont été regroupés sous l'appellation ECD pour Extraction de Connaissances à partir des Données ou KDD[1] – terme plus connu (cf. section 1.2.2).

Le KDD a fait ses preuves et continue d'en faire *(Agrawal, et al., 1993)*; en témoigne les nombreux logiciels de fouille de données et leur utilisation dans des domaines aussi variés (marketing, finance, médicale, etc.). Toutefois, l'émergence d'un nouveau type de donnée, en l'occurrence le type géo-spatial, modifie quelque peu la donne.

En effet, le développement de sciences telle la télédétection, la cartographie numérique, la géo-localisation, etc. combiné aux besoins du marché, ont suscité un intérêt de plus en plus grandissant pour l'information géo-spatiale. Dès lors, la distribution à grande échelle combinée à la facile accessibilité ont suscité le besoin d'apprendre de ces

[1] Knowledge Discovery in Databases est le processus global de fouille de données comportant plusieurs étapes dont la fouille de données. Toutefois, par abus de langage, fouille de données et KDD sont souvent confondu. Ainsi, sauf indication contraire, on emploiera le terme fouille de données pour faire référence au processus global le KDD.

données qui d'autant plus, offraient un potentiel énorme principalement comme support à la prise de décision *(Openshaw, et al., 1987) (Lu, et al., 1993) (Ester, et al., 1995) (Openshaw, 1999) (Buttenfield, et al., 2004)*.

Cependant, du fait des caractéristiques particulières de l'information géo-spatiale, les méthodes et outils du KDD ont montré leurs limites *(Shekhar, et al., 2001) (Zeitouni, 2002)*. Ainsi, il a fallu repenser à de nouvelles façon d'aborder cette problématique d'extraction de connaissances à partir de données géo-spatiales. De cela est née le GKD[2], une branche du KDD qui s'intéresse exclusivement à l'information géo-spatiale. La particularité de l'information géo-spatiale peut être résumée en trois(3) points (cf. section 1.2.3.2 pour plus de détail):

- Hétérogénéité
- Complexité
- Interdépendance

Plusieurs approches ont été développées afin de tenir compte de cette spécificité *(Han, et al., 1997) (Ester, et al., 1998) (Bogorny, et al., 2005) (Chelghoum, et al., 2002) (Malerba, et al., 2002)*. Malgré tout, bien des choses restent à faire en matière de fouille de données géo-spatiales.

1.2. État de l'art sur la fouille de données spatiales

1.2.1. Généralité

L'extraction de connaissances à partir des données est un domaine de connaissances qui prend de plus en plus de l'ampleur en raison principalement du besoin d'apprendre des grandes masses d'informations stockées dans les organisations et de la limite des techniques alors utilisées pour tirer parti du potentiel en terme de connaissances des données.

[2] Geographic Knowledge Discovery

Longtemps réduit à la tâche de fouille de données, l'ECD a été érigée en un processus regroupant diverses étapes (cf. section 1.2.2.1) couvrant la compréhension du domaine et des données, au déploiement des résultats de la fouille de données. La fouille en elle-même se compose d'un grand nombre d'algorithmes et de techniques que l'on peut catégoriser selon divers points de vue. Ainsi donc, vue sous l'angle de la finalité, ces algorithmes et techniques peuvent être regroupés en quatre(4) grandes catégories que sont la classification, l'estimation, la description, le clustering (cf. section 1.2.2.3). Tandis que d'un autre point de vue, ces algorithmes et techniques peuvent être regroupés en méthodes supervisées ou non selon que l'algorithme ait besoin d'une connaissance préalable des données manipulées (cf. section 1.2.2.2).

Dans cette section, nous avons choisi volontairement de parler du KDD (ECD) avant d'en venir à la particularité du champ de l'extraction de connaissances à partir de données géo-spatiales principalement parce que tout ce qui se dit pour le KDD, s'applique dans une large mesure au GKD. En effet, afin d'arriver à une extraction efficiente de connaissances utiles depuis des sources de données géo-spatiales, le GKD devrait, tout comme le KDD, être organisé sous forme de processus avec dans la théorie les mêmes étapes. La différence se fera au niveau de la pratique avec les différents moyens mis en œuvre pour traiter la composante géo-spatiale. Aussi, en considérant la fouille de données en tant qu'étape à part entière, les mêmes classifications peuvent se faire au niveau de la fouille de données géo-spatiales. L'objectif d'une fouille de données géo-spatiales ou non vise une estimation, une classification ou une description de ces dites données. Ceci dit, le KDD et le GKD ont beaucoup en commun même si ce dernier a ses spécificités (cf. section 1.2.3).

1.2.2. Extraction de Connaissances à partir des Données (ECD)

Le stockage massif d'informations au sein de nos organisations, combiné aux besoins d'apprendre de ces données ont contribué à la naissance du Knowledge Discovery in Databases (KDD) ou Extraction de Connaissances à partir de Données (ECD) *(Fayyad, 1996) (Frawley, et al., 1992)*.

L'ECD[3] (KDD) est un domaine de recherche à l'intersection de plusieurs disciplines parmi lesquelles les bases de données, l'intelligence artificielle, les statistiques, les reconnaissances de formes, l'apprentissage automatique, la visualisation des données *(Fayyad, et al., 1996) (Zytkow, et al., 2002)*, avec pour finalité l'extraction de connaissances au sein de données *(Fayyad, et al., 1996) (Ester, et al., 2001) (Kolatch, 2001) (Tan, et al., 2006)*.

De toutes les définitions possibles du KDD et de la fouille de données (plusieurs définitions ont été données et varient selon le domaine de compétence de leurs auteurs respectifs *(Friedman, 1997)*), on peut retenir que le KDD est un processus non trivial d'extraction de connaissances cachées et potentiellement utiles au sein d'entrepôts de données volumineux et à grande dimensionnalité *(Frawley, et al., 1992) (Han, et al., 1992)*.

La non-trivialité réfère au fait que, contrairement à la statistique qui est confirmatoire, la fouille de données est plutôt exploratoire *(Wijsen, 2001)*. En d'autres termes, avec le KDD, on ne sait pas a priori ce qu'on pourrait apprendre des données. Cette non connaissance a priori caractérise les résultats mis à nus qui sont plutôt cachés. La non-trivialité se justifie également par le fait que la découverte des connaissances passe par plusieurs étapes.

Les résultats d'une fouille de données devraient être non seulement utiles mais compréhensibles par les utilisateurs du domaine. En effet, les résultats devraient servir de support au processus de décision.

Une autre caractéristique essentielle du KDD est son utilisation sur de larges ensembles de données; large dans le sens que l'entrepôt contient un volume important de données, et que ces données sont décrites par plusieurs attributs.

Enfin, il est important de noter que le KDD est un processus; c'est-à-dire un ensemble d'étapes et d'actions dont la finalité est l'extraction de tendances et corrélations au sein des données. Contrairement aux idées reçues, le KDD ne se limite pas exclusivement à la fouille de données *(Miller, et al., 2001) (Ester, et al., 2001) (Kurgan, et al., 2006)* qui en constitue toutefois la partie visible. Ce dernier se compose en effet d'un ensemble d'étapes (cf. section 1.2.2.1) allant de la compréhension du domaine d'étude, à

[3] Dans la suite du document, nous utiliserons principalement l'acronyme anglais du terme Extraction de Connaissances à partir de Données à savoir KDD car celui-ci est largement utilisé dans la littérature.

l'exploitation des résultats de la fouille en passant par la fouille de données elle-même *(Fayyad, et al., 1996) (Piatestsky-Shapiro, 2002) (Hornick, et al., 2007)*. Il est important de noter que le processus d'extraction de connaissances est particulier en ce sens qu'il est interactif et itératif *(Piatetsky-Shapiro, et al., 1992) (Chen, et al., 1996) (Fayyad, et al., 1996) (Jambu, 1999)*.

- Par itératif, il faut entendre que le processus d'extraction de connaissances n'est pas un processus linéaire où chaque étape est appliquée une seule fois pour aboutir à la fin avec le modèle de connaissance recherché. Cela pourrait être le cas dans le meilleur des scénarii possibles mais cela arrive assez rarement dans la pratique.
- Le KDD est un processus « human-centric » c'est-à-dire que l'Homme est au cœur du processus d'où le qualificatif interactif. Il est important de relativiser le qualificatif ***automatique*** à tord attribué à la fouille. En effet, comme le note *(Brachman, et al., 1993) (Brachman, et al., 1994) (Fayyad, 1998)*, les outils de fouille ne sont pas des robots qui seuls doivent parcourir de larges ensembles de données afin d'y extraire quelques informations utiles à l'organisation. Bien au contraire, il s'agit d'un ensemble d'interactions entre l'utilisateur et les outils de fouille afin que les résultats obtenus au bout du processus soient non seulement compréhensibles mais utiles.

1.2.2.1. CRISP-DM ou processus d'extraction de la connaissance

Le CRISP-DM - Cross Industry Standard Process for Data Mining - est un processus normalisé qui décrit le cycle de vie d'un projet de fouille de données. L'objectif principal sous-tendu par le CRISP-DM est de standardiser le processus de fouille de données afin de le rendre indépendant du domaine d'application et des outils utilisés *(Larose, 2005) (Chapman, et al., 1999) (Reinartz, 2002)*. Le processus se compose de six (6) grandes étapes dont la fouille de données. Ces étapes vont de la compréhension du domaine au déploiement de la solution. Il est important de mentionner que les différentes phases de ce processus sont aussi importantes les unes que les autres *(Appice, et al., 2000)*.

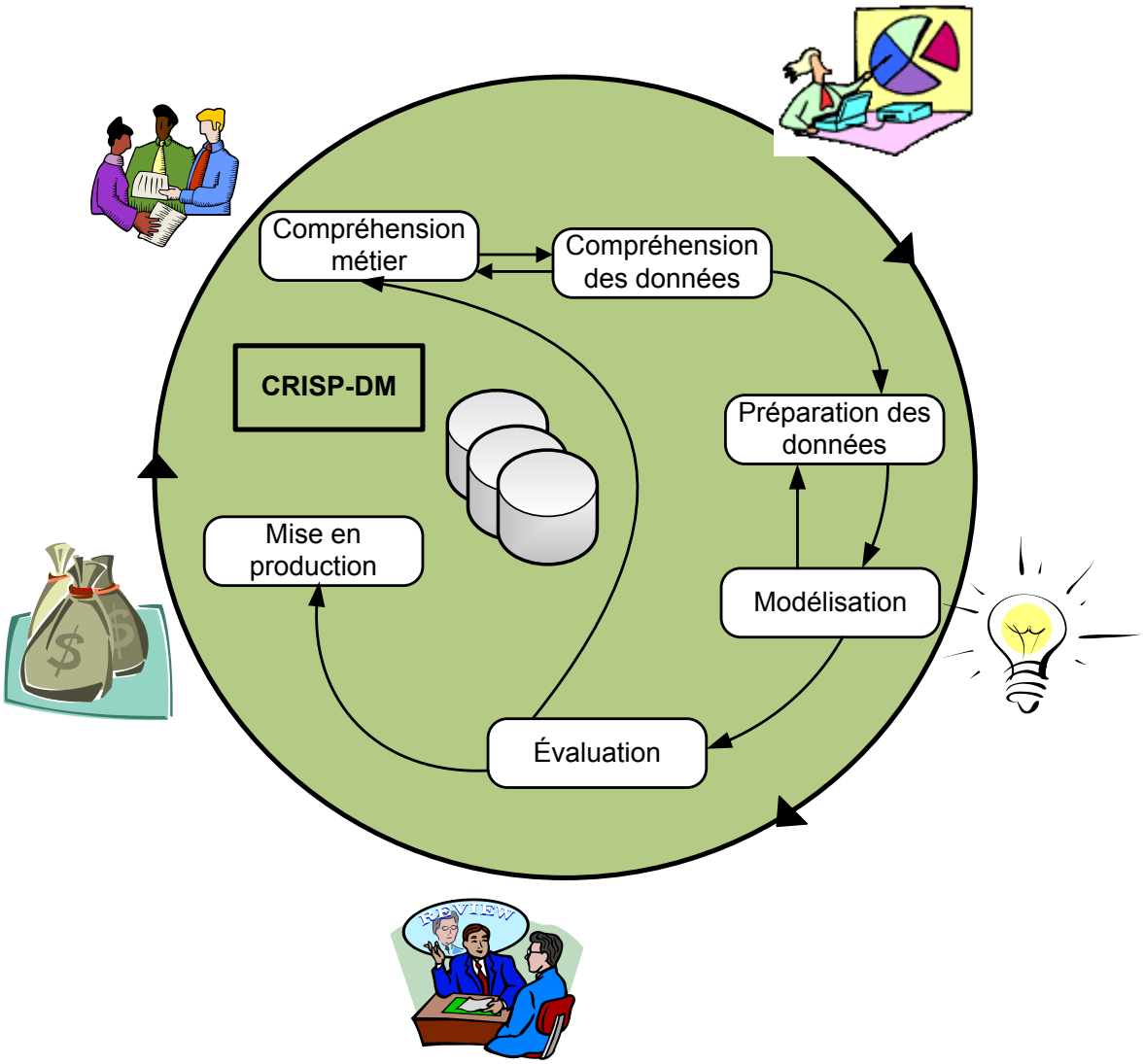


Figure 1.2-1: Les différentes étapes du CRISP-DM

1.2.2.1.1. Compréhension du métier

Cette étape est primordiale dans le sens où elle consiste en la compréhension du domaine dans lequel on désire effectuer de la fouille de données. Les tâches à effectuer au niveau de cette phase sont la:

- Formalisation en termes clairs des objectifs à atteindre ainsi que des exigences métiers en un problème de fouille de données,
- Traduction des objectifs en buts à atteindre,

- Mise en œuvre d'une stratégie pour atteindre ces objectifs.

1.2.2.1.2. Compréhension des données

Il s'agit de la phase de collecte de données. Tout au long de cette étape, on procèdera à une analyse exploratoire afin de se familiariser avec les données d'un point de vue qualité et disponibilité. Selon *(Kantardzic, 2003)* il est important de savoir comment la collecte des données affecte la distribution théorique de celles-ci. Cette information prend toute son importance quand vient le moment de modéliser et d'interpréter les résultats.

1.2.2.1.3. Préparation des données

La phase de préparation des données est celle qui requiert le plus d'effort, environ 60 à 70% de l'ensemble du processus *(Fayyad, 1998)*. Elle consiste essentiellement à :

- Nettoyer les données : il s'agit de nettoyer les données c'est-à-dire supprimer les données inconsistantes et développer entre autres des stratégies pour traiter les valeurs manquantes
- Transformer les données : il s'agit d'uniformiser les données ayant la même sémantique.
- Intégrer les données : cette étape revient à mettre les données à disposition au sein d'un référentiel de données afin d'en faciliter l'accès.

L'existence d'un entrepôt de données peut aider à diminuer sensiblement l'effort dépensé au niveau de cette phase. En effet, les données auront déjà passé par la phase d'Extraction-Transformation-Chargement avant d'être stockées dans l'entrepôt.

1.2.2.1.4. Modélisation

Cette étape est de loin la plus connue du processus global de découverte de connaissances et constitue le cœur de celui-ci *(Fayyad, et al., 1996)*. C'est à ce stade que sont appliquées les différentes techniques qui vont permettre l'extraction de modèles utiles à l'organisation. Pour cette phase, il s'agit de choisir la/les technique(s) de fouille qui sied au problème posé et de calibrer efficacement les paramètres.

Nous nous appesantirons davantage sur ce sujet au niveau de la section principalement dédiée à cet effet (cf. section 1.2.2.2).

1.2.2.1.5. Évaluation

Il s'agit de l'évaluation de la qualité des résultats issus de la phase de modélisation. Au cas, où les résultats obtenus ne sont pas satisfaisants, on pourrait retourner dans une étape précédente. Cette phase est d'une importance capitale dans le processus de fouille de données. Il ne s'agit pas seulement de produire des résultats ayant un certain niveau de précision mais il faut que ces résultats soient d'une interprétation facile pour les utilisateurs.

1.2.2.1.6. Mise en production

Il s'agit du déploiement « à grande échelle » des résultats obtenus à la phase de modélisation. Il pourra s'agir tout simplement de la génération d'un état ou bien de répéter diverses tâches de fouille sur les différentes données de l'entreprise.

1.2.2.2. La fouille de données ou datamining

La fouille de données constitue l'étape clé du processus de découverte de connaissances, étape sur laquelle bien des acteurs se focalisent d'emblé au détriment des autres étapes qui rappelons le sont toutes aussi importantes.

Depuis son apparition, la fouille de données a connu un essor fulgurant. A la faveur du développement technologique et des besoins du marché, elle s'est trouvée utilisée dans bien des domaines (Marketing, Télécommunication, détection de fraude, domaine manufacturier, etc.) *(Hornick, et al., 2007)*. La fouille de données en elle même consiste en l'utilisation des techniques et algorithmes afin d'extraire des connaissances implicites et potentiellement utiles, pouvant servir de support au processus de décision de l'entreprise. Son objectif principal est d'être soit prédictive, soit descriptive. Prédictive dans le sens de prédire la valeur future des variables étudiées et descriptive dans le sens de la production de modèle expliquant les données sous étude *(Kantardzic, 2003)*.

Elle est composée d'un grand nombre de techniques et algorithmes que l'on peut caractériser selon plusieurs points de vue : supervisé vs non-supervisé, prédictif vs descriptif, transparent vs opaques *(Tan, et al., 2006) (Hornick, et al., 2007)*.

Dans les lignes qui suivent, nous décrirons les différentes caractéristiques de la fouille de données. Notons toutefois que les différentes caractéristiques peuvent avoir des points communs dans la mesure où certaines techniques ou algorithmes possèdent plus d'une caractéristique.

1.2.2.2.1. Les méthodes supervisées/prédictives

Il s'agit de méthodes qui requièrent généralement de l'utilisateur la définition d'une variable cible dont on veut par exemple prédire la valeur. Il pourra s'agir par exemple, de prédire la valeur d'attributs dénommés variables dépendantes en fonction d'autres variables appelées variables explicatives ou indépendantes. Les algorithmes désignés comme supervisés fonctionnent généralement sur la base de trois(3) jeux de données *(Larose, 2005)*:

- jeu de données d'essai ou training set : contient l'ensemble des attributs y compris les valeurs de la variable à prédire. Ces valeurs aident à la supervision du processus en mettant à nus les erreurs quand l'algorithme utilise le modèle pour prédire les résultats *(Hornick, et al., 2007)*
- jeu de données test ou test set : contient les différentes variables exceptées les valeurs de l'attribut à prédire. Ce jeu de données contient après le lancement de l'algorithme, les valeurs prédites de la variable cible.
- jeu de données de validation ou validation set : est semblable au test set avec toutefois les vraies valeurs de l'attribut cible. Ce jeu de données sert à faire une confrontation avec les valeurs prédites de la variable cible afin d'estimer le pourcentage d'efficacité de l'algorithme utilisé.

On peut également subdiviser les méthodes supervisées/prédictives en deux(2) classes d'algorithmes: les algorithmes de classification et ceux de régression. La classification consiste en la prédiction de catégories de valeurs (valeurs discrètes) (exemple : quelle sera la réponse d'un client à une offre ? la réponse pouvant être « j'accepte sans réserve », « j'accepte avec réserve », « je refuse », etc.) Contrairement à la

régression qui prédit plutôt des valeurs continues (numériques) (exemple : quelle sera la valeur d'une maison ou le revenu d'une personne (25milles dollars par an, 10075.99 $, etc.). La classification et la régression diffèrent également quant à leur mode d'évaluation du résultat de la prédiction.

1.2.2.2.2. Les méthodes non supervisées/descriptives

Contrairement aux méthodes supervisées, celles non-supervisées n'utilisent pas de cible. Elles fonctionnent plutôt sur la base de recherche de structures intrinsèques, des relations, ou affinités dans le jeu de données fourni en entrée. En d'autres termes, il s'agit de trouver des tendances et corrélations qui résument les relations entre données *(Larose, 2005) (Tan, et al., 2006) (Hornick, et al., 2007)*. Ces méthodes peuvent servir par exemple dans des domaines tels la détection de fraude, la réduction de dimensionnalité. La plus connue des tâches d'apprentissage non supervisé est le clustering. Le caractère descriptif de ces méthodes réside dans le fait qu'elles décrivent de manière concise et résumée un jeu de données en présentant les propriétés intéressantes de ces données.

1.2.2.3. Classification des tâches de fouille de données

1.2.2.3.1. Classification - Estimation

Le terme classification pose souvent une ambigüité en raison de la confusion possible avec regroupement ou clustering. La classification est de loin l'une des tâches de fouille de données la plus utilisée car intervenant dans plusieurs domaines d'activité (Banque, Médecine,…) *(Larose, 2005)*. La finalité d'une tâche de classification est d'assurer la prédiction d'un attribut cible nominal (type chaine de caractère) sur la base d'une connaissance préalable des données qui leur sont fournies en entrée (training set).

Bien que largement utilisé, le problème majeur avec la classification demeure la nécessité de faire un compromis entre un modèle qui minimise et le taux d'erreur et la variance *(Larose, 2005)*. Comme algorithme accomplissant cette tâche, on note : le KNN[4] (cf. Annexe B), Les arbres de décision, les réseaux de neurones, les réseaux de Bayés.

---

[4] K-Nearest Neighbors

L'estimation ressemble beaucoup à la classification à la différence que la variable cible est numérique en lieu et place d'être nominale (type chaine de caractère) comme dans le cas de la classification. Comme algorithme, on note la régression linéaire simple ou multiple. Les réseaux de neurones peuvent être utilisés à cette fin.

Un exemple concret d'une tâche de classification est la prévision météo. En effet, pour prévoir le temps qu'il fera dans une zone géographique donnée, on prend en considération un certain nombre de paramètres nommés variables explicatives pouvant être l'humidité, la vitesse du vent, la température de la veille…et éventuellement la position de la zone géographique étudiée par rapport aux pôles. Ces paramètres ainsi que la variable à prédire (la température par exemple) pourraient être modélisés dans une fonction mathématique. Chaque fois qu'on aura alors besoin de prédire la température, il suffira de passer à cette fonction ou modèle les paramètres appropriés pour obtenir les résultats désirés.

1.2.2.3.2. Association

Encore connue sous le nom d'analyse d'affinité ou analyse du panier de consommation[5], la tâche d'association en fouille de données vise à voir quelles sont les variables qui vont ensemble *(Larose, 2005)*. Il s'agit de trouver des règles du type *si X alors Y* avec un certain niveau de probabilité *(Agrawal, et al., 1994) (Gardarin, 2006)*. Deux métriques sont utilisées pour caractériser généralement la qualité d'une règle d'association : le support et la confiance *(Hornick, et al., 2007) (Messaoud, et al., 2008)*. Le support décrit la probabilité d'existence de X et Y au sein du jeu de données. La confiance décrit quant à elle la probabilité d'existence de Y dans l'ensemble de données contenant X. Le problème majeur avec la tâche d'association est le nombre d'association possible entre attribut. En effet, avec k attributs prenant une valeur binaire, le nombre maximal d'associations est $K*2^{k-1}$. À titre d'exemple pour les algorithmes effectuant une tâche d'association, on note : GRI (General Rule Induction) et l'algorithme Apriori (pour plus de détails, cf. *(Agrawal, et al., 1993)*).

[5] Market Basket Analysis

À titre d'exemple, on peut mettre à profit les techniques d'association pour déceler les liens éventuels entre les différents produits vendus dans un supermarché. On peut ainsi noter que chaque fois que de la viande hachée est achetée, à 80% les pâtes sont également achetées. On note donc une certaine association entre les produits Viande et Pâtes avec 80% de taux de confiance. On pourra entreprendre comme action de disposer le rayon « Pâtes » à proximité de celui concernant la « Viande » afin d'amener le client à ne pas fournir d'effort pour aller dans le rayon « pâte » ou que celui-ci se rappelle qu'il doit acheter des pâtes au cas ou il aurait oublié.

1.2.2.3.3. Clustering

Selon *(Mirkin, 2005)*, l'idée de clustering renvoie tout simplement à l'utilisation de mesures de similarité (ou dissimilarité) entre les entités de sorte à regrouper ensemble celles similaires et celles dissimilaires dans un autre groupe. En d'autres termes, il s'agit d'une organisation des données en un ensemble de groupes homogènes *(Guha, et al., 1998)* : les clusters regroupés de telle sorte à minimiser la variance intra-classe et à maximiser celle interclasse.

Plusieurs algorithmes de clustering existent selon qu'ils utilisent différentes techniques d'évaluation de la similarité ou selon les objectifs visés (pour plus de détail cf. section 1.2.4):

- *Approche basée sur les partitions* : partitionne l'ensemble de données en un nombre K de clusters. Ces k partitions sont corrigées jusqu'à obtenir une similarité satisfaisante. On note comme algorithmes : k-Means, k-Médoïds, …
- *Approche basée sur la densité* : utilise un modèle probabiliste pour déterminer les entités denses au sein de l'ensemble de données. Les objets sont groupés selon que la valeur de la densité avoisine une certaine limite (voir les algorithmes DBSCAN, DENCLUE[6],…)
- *Approche hiérarchique* : qui récursivement construit les clusters sous la forme d'une structure hiérarchique selon une approche top-down ou bottom-up (agglomérative ou divisive). (voir les algorithmes CURE, STING, …)

[6] DENsity based CLUstering

- *Approche par grille* : l'espace est divisé en cellules pour former une grille multi niveaux. Par la suite, les cellules voisines sont groupées en fonction de la distance (CLIQUE,…)
- *Approche par modèle* : modélise les groupes et utilise le modèle pour classer les points. Exemple d'algorithmes : COBWEB, réseaux de neurones.

A titre d'exemple, on peut grâce au clustering, se pencher sur l'étude de la typologie des étudiants – toutes années confondues – d'une université; avec comme question de fond : est ce que la qualité de l'enseignement a baissée ou augmentée ? Ou est ce plutôt le niveau des étudiants qui a baissé ou non?

1.2.3. Extraction de connaissances à partir de données géo-spatiales ou GKD

Le GKD ou Extraction de Connaissances à partir de Données Géographiques (ECDG) est une sous branche du KDD qui a pour objet la découverte de connaissances implicites et potentiellement utiles au sein d'entrepôts de données géo-spatiales *(Miller, et al., 2001)* . Plusieurs facteurs ont concouru à la naissance de ce domaine de connaissances :

- la spatialisation[7] des systèmes d'information;
- le potentiel de connaissance que recèle la donnée géo-spatiale *(Openshaw, 1999)* ;
- l'inadéquation de méthodes et techniques de la fouille de données « traditionnelle ».

En effet, l'essor de technologies liées au traitement et stockage de l'information, au domaine spatial - la télédétection, la cartographie, à titre d'exemple – combiné au besoin du marché, a entrainé une accumulation de plus en plus importante de données géo-spatiales. Malheureusement, du fait de la complexité de ces données, de leur nature et des relations qu'elles entretiennent, l'utilisation des techniques « traditionnelles » de fouille de données

[7] Par spatialisation il faut entendre l'importance que prend de plus en plus l'information géo-spatiale au sein des organismes de sorte qu'il ait aujourd'hui peu de domaines qui ne désirent pas mettre à profit ce type d'information pour supporter le processus de décision.

s'est avérée impossible car produisant des résultats inconsistants *(Koperski, et al., 1996) (Ester, et al., 1997) (Popelinsky, 1998) (Shekhar, et al., 2003)*. L'inconsistance de résultats provient principalement des caractéristiques des entités géo-spatiales que *(Miller, et al., 2001)* résume comme suit (cf. section 1.2.3.2) :

- Dépendance spatiale et hétérogénéité,
- Diversité des types de données,
- Complexité des objets spatio-temporels.

Pour faire face à cette particularité des données géo-spatiales, diverses approches[8] ont été proposées *(Rinzivillo, et al., 2008)* :

- Une première approche consistant à développer des outils, algorithmes et méthodes capable de traiter dynamiquement les corrélations entre entités géo-spatiales.
- une seconde consistant à utiliser les outils de fouille « traditionnelle » mais en prenant soin d'extraire au préalable et sous forme de données de types classiques (chaine de caractère, numérique, booléen, etc.), les relations pouvant exister entre ces données.

Chacune des ces approches comporte sa part d'avantages et d'inconvénients. Nous y reviendrons beaucoup plus en détails dans les lignes à venir (cf. section 1.2.5).

Même si l'on note certaines dissemblances entre le GKD et le KDD, on peut cependant affirmer qu'ils ont beaucoup en commun, ne serait ce que dans la démarche générale (cf. section 1.2.2.1).

1.2.3.1. La fouille de données géo-spatiales

Comparée à la fouille de données « traditionnelle », la fouille de données géo-spatiales est un domaine relativement jeune qui à la faveur de l'évolution combinée en matière de structure de données et raisonnement spatial, des recherches de haut niveau sur

[8] La catégorisation des différentes approches de fouille de données géo-spatiales peut se faire selon divers point de vue. Ainsi on peut choisir d'aborder celle-ci sous l'angle des approches intégrées directement ou non à une base de données ou bien sous l'angle intra-thèmes ou inter-thème.

la fouille dans le monde relationnel, est en plein essor. Tout comme au niveau du KDD où la fouille constituait la principale phase, la fouille de données géo-spatiales reste une des étapes majeures du GKD mais n'en constitue pas la seule.

Elle (la fouille géo-spatiale) a fait l'objet de plusieurs définitions; définitions fonction du domaine d'application mais toutefois complémentaires. Ainsi pour *(Koperski, et al., 1996)* la fouille géo-spatiale vise l'extraction de connaissances, de corrélations spatiales non explicitement stockées dans un référentiel de données géo-spatiales; ou selon *(Han, 1997)* de tout autre modèle ou pattern non explicitement stockés. Pour *(Miller, et al., 2001)*, il s'agit de l'application de techniques et méthodes en vue de trouver des patterns intéressants entre objets et évènements géographiquement distribués dans l'espace et dans le temps.

Ces modèles ou patterns impliquent les propriétés spatiales de chacune des entités géo-spatiales ainsi que leurs relations. Ceci dénote comme le dit *(Miller, et al., 2001)* de l'importance que joue la dépendance et la proximité spatiale dans le processus de fouille de données géo-spatiales et constitue du même coup la particularité de ce dernier par rapport à la fouille de données classiques.

1.2.3.2. Caractéristiques de la fouille de données géo-spatiales

La particularité de la fouille de données géo-spatiales est intrinsèquement liée aux caractéristiques des données traitées. Cette différence peut être caractérisée selon cinq (5) points de vue *(Shekhar, et al., 2003)*:

1. les données,
2. les relations géo-spatiales,
3. l'hétérogénéité,
4. les fondements statistiques,
5. la consommation de ressources.

1 - Sous l'angle des données, on peut noter que comparées à la fouille de données « traditionnelle », les données traitées au niveau de la fouille de données géo-spatiales sont plus larges, complexes et volumineuses. En effet, l'information géo-spatiale dispose de deux(2) composantes :

- une composante descriptive qui est à l'image des données utilisées a niveau de la fouille classique, i.e. des données de type chaine de caractères, numérique, date, booléen, etc. Son principal intérêt est de décrire certaines caractéristiques de la composante géométrique. A titre d'exemple, lorsqu'on considère un ensemble de maisons spatialement référencées, l'information descriptive attachée pourra être le nombre de personnes vivant dans les dites maisons, le revenu du ménage y vivant, le nombre d'enfants, etc.
- une composante géométrique qui caractérise la position spatiale des objets et/ou leur localisation.

Si au niveau de la composante descriptive, le problème de relation entre les données ne se pose pas, en ce qui concerne la composante géométrique, le problème est tout autre. En effet, les données géométriques entretiennent des relations implicites entre elles. Cette constatation est d'autant plus vraie qu'elle a été érigée en première loi de la géographie ou loi de Tobler qui stipule en substance *(Miller, 2004) (Chawla, et al., 2001) (Bogorny, et al., 2005)* :

> ***« … les entités spatiales sont inter reliées. Celles plus proches s'influencent plus comparée à celles plus distantes»***.

Cette notion se rapporte à l'auto corrélation spatiale et se doit d'être prise en compte afin de produire de bons résultats à l'issue du processus de fouille.

2 - Les relations de voisinage constituent le cœur de la spécificité de la fouille de données géo-spatiales. En terme de nature de relation entres ces données, on en distingue trois(3) *(Miller, et al., 2001)*:

- Relation euclidienne ou métrique qui réfère généralement à la distance ou toute autre fonction utilisant cette distance d'une manière ou d'une autre (exemple : buffer, proche de, loin de,…)
- Relation topologique : il s'agit de relation décrivant l'interaction entre les frontières, intérieur et extérieur de deux entités géo-spatiales. Ce type de relation reste invariant quelques soient les transformations effectuées sur les données géographiques.

- Relation directionnelle : il s'agit du type de relation permettant de savoir la position géographique (Nord, Sud-est, etc.) d'une entité vis-à-vis d'une autre.

3 - L'hétérogénéité des données géo-spatiales. Cette notion d'hétérogénéité spatiale est fondée sur le principe que les variations spatiales des objets sont fonction de leur localisation et peut être évaluée par des mesures locales de l'auto corrélation spatiale *(Chawla, et al., 2001)*. En d'autres termes, il s'agit tout simplement de la difficulté de généralisation d'un phénomène géographique *(Miller, et al., 2001)*.

4 - Sous l'angle des fondements statistiques, l'application des principes de la fouille « traditionnelle » sur les données géo-spatiales produit des résultats biaisés *(Shekhar, et al., 2003)*. En effet, la statistique classique part sur une base d'indépendance entre les données alors que, de par les relations qu'elles entretiennent, les entités géo-spatiales s'influencent mutuellement. D'où la limite des techniques de fouille « traditionnelle » quant aux traitements des données géo-spatiales *(Koperski, et al., 1998)*. Dans une certaine mesure toutefois, on pourra adapter certaines techniques statistiques à la fouille de données géo-spatiales sous réserve d'avoir une démarche solide et méthodique. Ce qui sous-entend beaucoup d'effort et peut être même l'assistance d'un expert afin de mieux prendre en compte les possibles influences entre entités *(Koperski, et al., 1998).*

5 – La consommation de ressources lors du traitement est un autre point qui caractérise les données spatiales. Ce point est une conséquence directe de l'auto corrélations des données spatiales. En effet, extraire cette information au sein de gros volumes de données exige de la ressource *(Shekhar, et al., 2003)*. Afin de réduire cette charge computationnelle, diverses méthodes ont été mises en œuvre qu'il s'agisse de méthodes exploitant les index spatiaux à travers des graphes de voisinage *(Ester, et al., 1997)* ou de celles passant par un échantillonnage *(Ng, et al., 2002)*.

1.2.3.3. Principales tâches visées par la fouille de données géo-spatiales

Afin de pouvoir exploiter le potentiel de connaissances stockées dans les vastes entrepôts de données géo-spatiales, plusieurs méthodes on été mises en œuvre *(Koperski, et al., 1996) (Koperski, et al., 1998) (Ester, et al., 1999)*. Ces différentes tâches peuvent être catégorisées selon leur finalité. On note :

1.2.3.3.1. Généralisation spatiale

Il s'agit de découvrir des connaissances implicites stockées dans un référentiel de données en effectuant une généralisation depuis des données à fine granularité vers des données avec une granularité plus élevée sur la base de « concepts de hiérarchies » qui décrivent la hiérarchie au sein des données. Cette hiérarchie entre les données peut être fournit en entrée du processus de fouille par les experts ou être découverte grâce à une analyse préalable.

Au niveau des référentiels spatiaux, on distingue deux sortes de « concepts de hiérarchies » : spatial et non spatial. De ce fait, la généralisation peut s'effectuer avec soit la composante descriptive, soit avec celle spatiale. Cela donne naissance à deux types d'approches pour la généralisation spatiale :

- ***Spatial-data-dominant generalization*** : Elle consiste à effectuer d'abord une généralisation sur la composante spatiale des données en fusionnant les zones géographiques appartenant à la même hiérarchie. On réalise par la suite, sur les bases de la généralisation spatiale, une généralisation sur les attributs non spatiaux.
- ***Non Spatial-data-dominant generalization*** : Il s'agit de réaliser le contraire en effectuant tout d'abord une généralisation sur les attributs non spatiaux. Sur la base des résultats obtenus, on effectue ensuite une généralisation spatiale.

1.2.3.3.2. Exploration des associations spatiales

Il s'agit des trouver des règles d'associations spatiales dénommées SAR (Spatial Association Rules) entre entités géo-spatiales ou non. Une règle d'association spatiale est

un prédicat qui exprime une relation entre objets spatiaux ou non. Ce prédicat est de la forme *si on a toute occurrence d'un objet spatial donné, alors il correspond une ou plusieurs occurrences de tels ou tels objets*. On peut associer à ce prédicat, une probabilité qui caractérise le support de la règle *(Han, et al., 1997) (Rinzivillo, et al., 2008).* On peut catégoriser les algorithmes SAR en deux (2) approches *(Rinzivillo, et al., 2008)*:

- Une première approche basée sur un raisonnement quantitatif ; qui calcul la distance entre deux entités géo-spatiales sur la base des coordonnées. Elle a comme désavantage de ne traiter que les objets spatiaux de type point. En plus de ne pas tenir compte des attributs non spatiaux dans le traitement, elle ne considère les relations que d'un point de vue quantitatif.
- La seconde approche est basée sur un raisonnement qualitatif et permet d'extraire différents types de relations spatiales.

Quelque soit l'approche utilisée, il faut mettre en œuvre pour les algorithmes utilisant SAR, un moyen de filtrer les relations. En effet, l'extraction renvoie le plus souvent des prédicats trop évidents alors que l'objectif de la fouille est la découverte de connaissances implicites et non connues a priori.

1.2.3.3.3. Classification ou clustering spatiale

Les algorithmes appartenant à cette catégorie permettent d'organiser les entités géo-spatiales de telle sorte à obtenir des classes d'objets partageant les mêmes caractéristiques. Ces classes d'objets sont dénommées Clusters. Les objets d'un même cluster sont beaucoup plus homogènes entre eux comparés à ceux provenant d'un autre cluster.

Dans le monde de la fouille de données géo-spatiales, beaucoup d'auteurs se sont penchés sur le développement d'algorithmes de clustering spatial d'où l'existence d'une pléthore d'algorithmes et méthodes dédiés à cet effet. Ces algorithmes peuvent être scindés en cinq (5) grandes familles *(Kolatch, 2001) (Han, et al., 2001) (Ng, et al., 2002)*(cf. section 1.2.4).

1.2.4. Quelques algorithmes de clustering spatial

Avant de passer à la description des différentes familles d'algorithmes de clustering, *(Kolatch, 2001)* note que pour que ces algorithmes soient efficients, il leur faut avoir les caractéristiques suivantes :

- Gestion de la montée en charge et efficience dans le traitement de gros volume d'informations ;
- Insensible à la quantité de données aberrantes ;
- Insensibilité à l'ordre d'entrée des données ;
- Possibilité de traiter les données à grande dimension.

1.2.4.1. Méthodes hiérarchiques

Ces méthodes sont soit agglomératives (approche bottom-up), soit divisives (approche Top-down). A titre d'exemple d'algorithmes utilisant ces méthodes, on note DIANA et AGNES. Les méthodes agglomératives débutent avec n clusters et au fur et à mesure du processus, deux clusters sont fusionnés jusqu'à ce qu'on obtienne un cluster au finish.

Les méthodes divisives, commencent elles avec un cluster. Au fur et à mesure, un cluster est choisi et scindé en deux de sorte à obtenir n cluster en fin de traitement.

Ces deux méthodes ont comme désavantage de ne pouvoir annuler une fusion ou une scission effectuée dans une étape précédente. Pour palier à cette situation, d'autres algorithmes ont vue le jour : CURE, CHAMELEON, BIRCH.

1.2.4.1.1. BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

Il s'agit d'une méthode de clustering hiérarchique intégrée qui parcourt seulement une seule fois la base de données. Avec les données ainsi scannées, BIRCH met en œuvre des sous-clusters sur la base desquels il construit par la suite un CF-Tree (Clustering Feature Tree). Une CF-Tree est un arbre balancé stockant des CF (Clustering Feature) et construit dynamiquement au fur et à mesure que de nouvelles données sont insérées. Le CF

représente un triplet d'informations sur un cluster donné : le nombre de points dans le cluster N, la somme linéaire des éléments SL, et la somme des carrées des éléments SC.

*(Kolatch, 2001)*BIRCH est efficient en termes de minimisation des opérations d'entrée/sortie d'une part et d'autre part dans le traitement de larges ensembles de données. Pour atteindre cette efficience, l'opération de clustering est réalisée en quatre (4) principales phases dont deux optionnelles :

Une première phase de pré-clustering au cours de laquelle la base de données est scannée et un CF-Tree représentant les régions denses est construit. L'algorithme appliqué à ce stade est incrémental et approximatif.

La deuxième phase consiste à re-parcourir l'arbre afin de construire un arbre plus petit. Cette phase peut servir à l'élimination des données inconsistantes.

La troisième phase essaie de compenser la dépendance à l'ordre d'entrée des données ; c'est-à-dire la possibilité de produire les mêmes résultats quelque soit l'ordre dans lequel les données sont passées à l'algorithme. Pour cela, BIRCH utilise un algorithme existant de clustering basé sur les centroïdes pour la formation des différents clusters. Cet algorithme appliqué sur les sous-clusters prend en entrée soit un nombre désiré de clusters, soit un seuil qui représente la taille ou le diamètre d'un cluster.

La quatrième phase construit les clusters finaux sur la base des centroïdes déterminés au niveau de la phase précédente.

Le maillon faible de BIRCH est l'algorithme utilisé pour la création des centroïdes après le scan initial des données ; surtout lorsque les clusters crées ne sont pas de même taille ou lorsque les clusters sont circulaires. Le problème de clusters circulaires peut être éliminé en procédant à des scans répéter des données. Mais comme on peut s'y attendre, cela est cause d'une dégradation de performance.

1.2.4.1.2. CURE (Clustering Using Representatives)

Comme le note *(Kolatch, 2001)*, il s'agit d'une méthode agglomérative qui utilise un principe plus sophistiqué de fusion de clusters. En lien et place d'utiliser un centroïde pour la fusion, CURE utilise un nombre fixe d'objets bien dispersés pour chaque cluster. Par la suite, les objets représentatifs sont divisés à travers leur centre grâce à un coefficient variant

entre 0 et 1. Il faut noter que CURE est un algorithme assez robuste dans le traitement des valeurs extrêmes.

À la différence de BIRCH qui utilise une phase de préclustering pour le traitement de larges volumes de données, CURE effectue plutôt un échantillonnage. Cela permet d'une part de réduire les coûts d'entrée/sortie dans la mesure où les données échantillonnées peuvent résider dans la mémoire centrale. D'autre part, l'échantillonnage peut servir à l'élimination de données déviées. Par ailleurs, L'échantillonnage devrait être réalisé de sorte à éliminer la probabilité de ne pas trouver de clusters. Les avantages de CURE sont entre autre :

- La découverte de clusters de taille intéressante,
- La non sensitivité aux données déviées,
- Le partitionnement et l'échantillonnage réduisent la taille des données sans influer sur la qualité des clusters,
- Temps d'exécution réduit

1.2.4.1.3. CHAMELEON

CHAMELEON (cf. *(Karypis, 1999)* pour plus de détail) est une technique combinant le partitionnement et un clustering hiérarchique. Similaire à CURE, il améliore la qualité des clusters par l'application d'un critère élaboré dans la fusion de deux clusters. En effet, deux clusters seront fusionnés si l'inter-connectivité et la proximité du cluster fusionné est assez similaire à l'inter-connectivité et à la proximité des clusters à fusionner. L'inter connectivité et la proximité sont déterminées en prenant en compte les caractéristiques propres à chaque cluster.

1.2.4.2. Méthodes à partitionnement

Ces méthodes recherchent les k meilleurs clusters d'un ensemble n. Plusieurs méthodes permettant la découverte des meilleurs clusters existent. Parmi ces méthodes, on note :

- K-Means : il s'agit d'un algorithme relativement évolutif (scalable) et efficient dans le traitement de gros volume de données. Son inconvénient réside dans sa sensibilité aux données extrêmes (outliers)

- K-Medoïd : contrairement à K-Means, il est moins sensible aux données incohérentes et « outliers ». malheureusement, il exige un temps de traitement élevé parce qu'à la différence des deux précédents, il utilise comme centre du cluster, l'objet le plus centralement localisé : le Medoïd.
- EM (Expectation Maximisation) : cet algorithme évolue sur la base d'une probabilité gaussienne. Contrairement aux deux autres algorithmes, chaque objet peut appartenir à deux clusters mais avec une certaine probabilité.

1.2.4.2.1. PAM (Partitionning Arond Medoïds)

PAM est une technique de clustering utilisant les K-medoïds pour l'identification des clusters. Il permet de déterminer K-clusters en trouvant pour chacun d'eux l'objet le plus centralement localisé dans le cluster : « le Médoïd ».

La démarche de PAM est la suivante : après avoir sélectionné arbitrairement K Medoïd au départ, les objets non sélectionnés sont assignés à chaque cluster en fonction de la distance séparant chaque objet de chaque cluster. Un objet sera assigné à un cluster si la distance ou fonction de dissimilarité est minimale comparée aux autres. Les étapes suivantes consistent pour PAM à effectuer des itérations afin d'améliorer la qualité des clusters trouvés. Ainsi pour chaque cluster, PAM essaie de retrouver un médoïd de meilleure qualité. Si cela est le cas, le médoïd en cours est remplacé et les distances entre non-médoïds et le nouveau cluster sont évaluées afin de redéfinir l'appartenance des objets aux clusters. L'itération continue jusqu'à ce que les K-clusters trouvés soient de meilleure qualité.

On note que PAM est performant dans le traitement des ensembles de données moyennement volumineux. Plus les données sont larges, plus on note une dégradation des performances de PAM. Ce qui à d'ailleurs motivé la mise en œuvre d'un nouvel algorithme dénommé CLARA.

1.2.4.2.2. CLARA (Clustering LARge Applications)

(Kolatch, 2001) (Ng, et al., 2002) CLARA a été développé dans l'optique de remédier aux insuffisances de PAM quant au traitement de larges volumes de données. A la différence de PAM qui travaille sur l'entièreté des données, CLARA évolue sur la base

d'un échantillonnage. Ainsi donc, CLARA récupère un échantillon de l'ensemble des données et applique par la suite PAM sur l'échantillon ainsi sélectionné. Si le choix de l'échantillon est suffisamment aléatoire, de fortes chances existent que les médoïds de l'échantillon soient approximativement les mêmes que les données initiales. Afin d'obtenir des médoïds qui approchent le plus ceux des données initiales fournit en entrée, en lieu et place de travailler sur un seul échantillon, CLARA travaille sur de multiples échantillons et fournit en fin de traitement, le cluster de meilleure qualité, c'est-à-dire l'échantillon dont les médoïds sont plus proches de la réalité.

La qualité et la précision du clustering sont mesurées sur la base de la dissimilarité moyenne de tous les objets dans les données initiales et non d'un échantillon.

1.2.4.2.3. CLARANS (Clustering Algorithm based on Randomized Search)

(Ng, et al., 2002) CLARANS a été mise en œuvre pour suppléer aux insuffisances de PAM et CLARA. Tout comme CLARA et contrairement à PAM, CLARANS effectue également un échantillonnage. Mais à la différence de CLARA au niveau duquel l'échantillonnage est effectué au tout début, CLARANS réalise plutôt un échantillonnage dynamique.

Cet algorithme prend en entrée deux (2) paramètres: *maxneighbor* et *numlocal*. *Maxneighbor* représente le nombre maximum de voisins d'un nœud qui devront être examinés. *Numlocal* représente le maximum de minima locaux qui peuvent être collectés.

La difficulté majeure au niveau de CLARANS est la fixation de ces deux paramètres. De mauvais paramètres en entrée entraînent de sérieuses dégradations de performances. Par exemple, plus la valeur du paramètre *maxneighbors* est élevée, plus les performances de CLARANS sont proches de celles de PAM. D'un autre côté, moins la valeur de *maxneighbor* est élevée, moins les clusters produits sont de qualité médiocre.

(Kolatch, 2001) note que même si CLARANS a l'avantage de n'être pas sensible à l'ordre d'entrée des données, il traite tout de même difficilement les larges volumes de données - en termes de quantité et de dimensionnalité - dans la mesure où les données traitées résident dans la mémoire centrale. Également, il est influencé par les données inconsistantes du fait de la recherche aléatoire dynamique.

En vue de remédier à l'incapacité de CLARANS – mais aussi de la plupart des algorithmes de clustering - à traiter les types d'objets géométriques autre que ceux de types point, des améliorations ont étés apportées à cet algorithme en vue de la possibilité de traitement des polygones convexes. Si l'évaluation de la distance au niveau des objets de type point ne pose nullement de problème, la situation est tout autre au niveau des polygones. De ce fait, plusieurs techniques d'évaluation de la distance ont été proposées. On note :

- La technique centroïdes qui évalue la distance grâce aux centroïdes des polygones
- La technique MV-approximation qui évalue la distance minimale entre tous les sommets de chaque polygone
- La technique Separation Distance qui évalue la distance entre tous les points d'un polygone A par rapport à un polygone B
- La technique R-approximation qui évalue la distance entre les rectangles circonscrits aux polygones

A priori, on pourrait penser que la technique utilisant les centroïdes est beaucoup plus adaptée mais elle produit des résultats biaisés lorsque les polygones sont de formes diverses. *(Ng, et al., 2002)* ont démontré après moult expériences que la technique R-approximation est la mieux adaptée et on gagne en temps en mémorisant les différents calculs de distances afin de pas effectuer plusieurs fois les mêmes calculs.

1.2.4.2.4. SD CLARANS ET NSD CLARANS

(Ng, et al., 2002) notent qu'il existe deux façons d'effectuer de la fouille de données spatiales. D'une part traiter conjointement les données spatiales et non spatiales. D'autre part, il s'agit de traiter seulement les données spatiales et d'utiliser d'autres techniques pour le traitement des données non spatiales. C'est de ce contexte que sont nées SD (Spatial Dominant) CLARANS et NSD (Non SD) CLARANS. Il s'agit de deux extensions de CLARANS qui rappelons le découle lui-même de CLARA et PAM.

L'approche Spatial Dominante consiste à utiliser CLARANS pour effectuer la fouille de données sur les objets spatiaux. Avec les clusters ainsi obtenus, les données non spatiales sont récupérées et fournies entrée à des outils de fouille « traditionnelle » en l'occurrence DBLEARN dans l'expérience menée par *(Ng, et al., 2002)*

L'approche Non Spatial Dominante, consiste à effectuer en premier lieu une généralisation sur les attributs non spatiaux et par la suite utiliser CLARANS pour la formation des clusters spatiaux.

En termes de comparaison, *(Ng, et al., 2002)* notent que sous certaines conditions SD CLARANS est plus performant que NSD CLARANS. Mais de façon générale, les deux algorithmes produisent de bons résultats et le mérite revient à CLARANS qui est efficace dans la formation de clusters spatiaux.

1.2.4.3. Méthodes basées sur la densité

1.2.4.3.1. DBSCAN (Density Based Spatial Clustering Application with Noise)

À la différence de CLARANS, DBSCAN effectue un clustering basé sur la densité. Cette technique se base sur le principe selon lequel la densité est plus élevée à l'intérieur d'un cluster qu'à l'extérieur. L'algorithme prend deux(2) paramètres en entrée : Eps et MinPts. MinPts représente le nombre de points maximaux à l'intérieur d'un cluster. Eps quant à lui représente la distance maximale qui devrait séparer un point quelconque du centre du cluster *(Kolatch, 2001)*.

Ces deux paramètres doivent être déterminés avant chaque exécution de DBSCAN. Ce qui constitue un des inconvénients majeurs de cet algorithme en plus de l'incapacité de traiter les données à grande dimensionnalité. En termes d'avantages, on note que DBSCAN est performant dans le traitement des larges volumes de données ainsi que des données déviées et inconsistantes.

1.2.4.3.2. DENCLUE (DENsity based CLUstEring)

(Kolatch, 2001) DENCLUE est un algorithme qui hérite tout à la fois des méthodes basées sur le partitionnement, la densité et la hiérarchie. Cet algorithme s'appuie sur l'hypothèse selon laquelle l'influence d'un point sur ses voisins peut être modélisée

mathématiquement sous forme d'une fonction dénommée fonction d'impact. Cette fonction est alors appliquée à chaque point de l'espace de données pour ensuite dériver la densité qui n'est autre que la somme des fonctions d'impact.

On note que cet algorithme est efficace dans le traitement des données à large dimensionnalité ainsi que dans le traitement de clusters de formes différentes.

1.2.4.3.3. DBCLASD (Distribution Based Clustering of Large Spatial Databases)

Il s'agit d'un algorithme de clustering basé sur la densité qui contrairement à DBSCAN suppose que les points sont uniformément repartis à l'intérieur d'un cluster. En effet, on note que la distance d'un point quelconque à son voisin est plus petite à l'intérieur d'un cluster qu'à l'extérieur.

*(Kolatch, 2001) (Xu, 1998)*DBCLASD est un algorithme incrémental c'est-à-dire que les points sont traités en fonction des points précédemment traités. Cela fait de DBCLASD un algorithme dépendant de l'ordre d'entrée des données. Cependant, pour remédier à cette faiblesse, l'algorithme retraite les points qui n'ont pas été assignés avec succès à un cluster. Ce qui peut entrainer une réassignation des points à d'autres clusters.

En termes d'avantages, DBCLASD est capable de traiter des larges volumes de données ainsi que des données à haute dimensionnalité. En outre, à la différence d'autres algorithmes, il ne requiert pas de paramètres en entrée et traite efficacement les données inconsistantes car fondé sur une probabilité basée sur le facteur distance. A son désavantage, DBCLASD, comparé à d'autres algorithmes, consomme beaucoup de ressources. En plus, l'hypothèse selon laquelle les points sont uniformément répartis est un facteur limitatif quant à son efficacité.

1.2.4.4. Méthodes basées sur les GRID

1.2.4.4.1. STING (Statistical Information Grid Based Method)

Cette technique *(Kolatch, 2001)* exploite les propriétés de clustering des structures d'index. STING divise la zone spatiale en différentes cellules régulières suivant par exemple la latitude et la longitude. Les cellules sont structurées sous forme hiérarchiques en ce sens qu'une cellule donnée est subdivisée en un nombre défini de sous cellules. En

plus de stocker le nombre de points qu'elles contiennent, chaque cellule stocke également un certain nombre d'informations statistiques dont :

- M : correspond à la moyenne des valeurs contenues dans la cellule,
- S : correspond à l'écart type de la valeur des différents attributs de la cellule,
- Min : correspond à la valeur de l'attribut minimal dans la cellule,
- Max : correspond à la valeur de l'attribut maximal dans la cellule,
- Dist : correspond à la distribution dans la cellule. Cette valeur peut être uniforme, exponentielle, non déterminée, etc.

Le calcul et stockage de ces valeurs au niveau de chaque cellule est effectué selon une approche ascendante tandis que l'opération de clustering est effectuée depuis la racine de l'arbre jusqu'aux feuilles. L'opération de clustering au niveau de STING est d'ailleurs similaire à celle de DBSCAN qui en lieu et place d'utiliser des cellules, considère plutôt des points. Ainsi donc, les cellules sont regroupées pour former un même cluster au regard leur proximité ou si un certain seuil est atteint.

En termes d'avantages, cet algorithme n'est pas sensible à l'ordre des données ni aux données inconsistantes et déviées. Il s'en sort efficacement dans le traitement des larges volumes de données et fournit d'ailleurs des clusters de qualité tant que la granularité reste fine.

1.2.4.4.2. CLIQUE[9] (Clustering In QUEst)

CLIQUE est un algorithme de clustering basé sur la densité et les grilles qui a comme particularité la possibilité de traiter les données à grandes dimensions c'est-à-dire avec un nombre élevé d'attributs.

Afin d'obtenir une approximation de la densité, chaque dimension est partitionnée en intervalles de taille égales en utilisant une approche ascendante. Chaque partition ayant alors le même volume de données, les densités sont ensuite dérivées selon le nombre de points contenus dans chaque partition. Ces densités aident à l'identification automatique des sous-espaces dans lesquels les clusters sont déterminés en séparant les points selon une

[9]Cf. *(Agrawal, 1998)*pour plus de détails

fonction de densité et un regroupement des partitions à haute densité connectées dans le sous-espace.

L'identification des clusters au niveau de CLIQUE est réalisée en trois (3) grandes phases :

- Identification des sous espaces contenant les clusters : cette opération est réalisée en utilisant un algorithme ascendant de recherche d'unités denses.
- Identification des clusters : consiste à trouver les composants connectés en utilisant les sommets des unités denses. L'identification des clusters est bien entendue fonction du nombre d'unités denses.
- Génération d'une description minimale des clusters grâce aux composants déterminés dans la phase précédente.

CLIQUE est efficace dans le traitement de données contenant un certain taux de données déviées et inconsistantes. Il requiert de l'utilisateur, deux paramètres représentant le seuil de densité et le nombre d'intervalles d'égale longueur. On note toutefois que la qualité des clusters déterminés peut être mise en cause lorsque le seuil de densité est moins élevé.

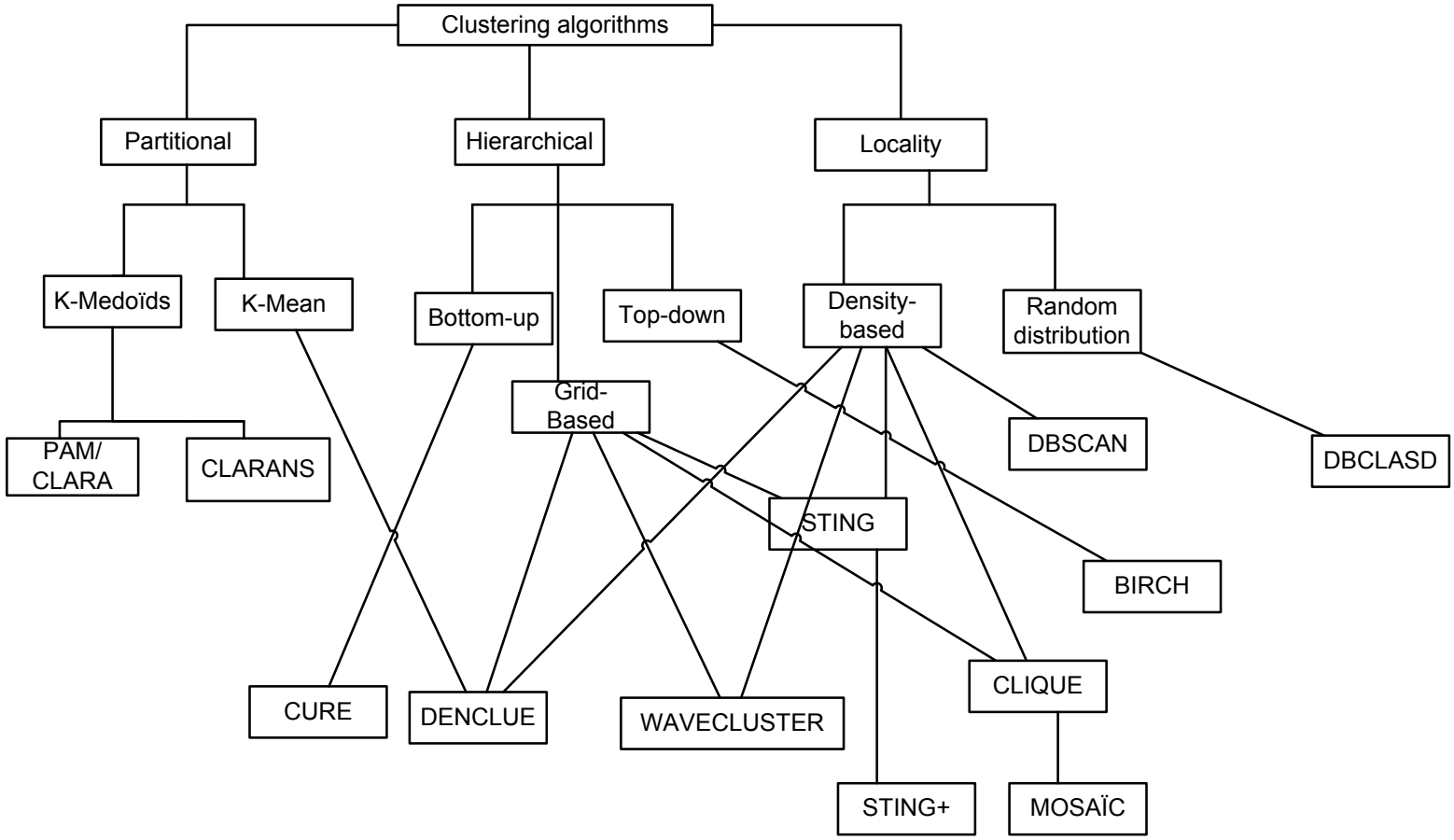


Figure 1.2-2: récapitulatif des différents algorithmes de clustering spatial

1.2.5. Approches de fouille de données géo-spatiales

1.2.5.1. Approche basée sur le prétraitement des données

Cette approche préconise la mise à profit des outils de fouille de données « traditionnelle» pour le traitement de la donnée géo-spatiale. Mais avant l'utilisation de ces outils, il est nécessaire de procéder à un prétraitement préalable (voir Figure 1.2-3). Ce prétraitement vise l'extraction et la représentation de façon explicite (sous la forme d'attributs de type classique) des relations existant entre les entités géo-spatiales *(Bogorny, et al., 2005) (Rinzivillo, et al., 2008)*.

Cette approche est assez pratique puisque simple à mettre en œuvre d'une part, d'autre part permet d'exploiter les outils existants qui ont déjà fait leurs preuves *(Klösgen, et al., 2002)*. Toutefois, elle comporte quelques inconvénients dont:

- la consommation énorme de la ressource temps: en effet, l'extraction peut nécessiter beaucoup de temps de calcul lorsque plusieurs relations doivent être extraites. Aussi, beaucoup de ces relations peuvent être non intéressantes à exploiter dans un contexte de fouille de données. Pour cela, certains auteurs *(Bogorny, et al., 2005)* s'attachent à filtrer ces types de relations.
- la redondance dans le stockage : les données prétraitées tout comme les données sources, sont stockées. On a de ce fait, un double stockage des mêmes données.
- la difficulté de mise à jour en cas d'ajout, modification ou suppression : cet inconvénient est lié à la redondance dans le stockage. Lorsque les données sources sont modifiées, il faut repasser par la phase de prétraitement dans la mesure où les données prétraitées sont stockées dans un autre schéma, table ou fichier.
- la difficulté d'utiliser les données pour la visualisation qui est aussi une résultante de la redondance de stockage. Les résultats de la fouille effectuée sur les données prétraitées ne se prêtent pas à une visualisation cartographique au regard de l'absence de composante géométrique. Il est

donc difficile de faire un couplage entre un outil de fouille géo-spatiale faisant du prétraitement et un visualisateur cartographique.

Plusieurs auteurs ont abordé cette approche (cf. sous sections 1.2.5.1.1 à 1.2.5.1.3)

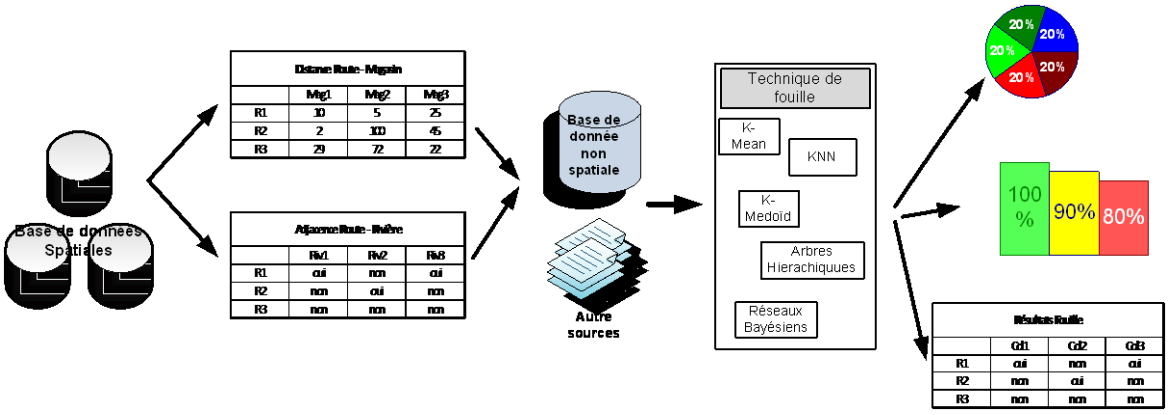


Figure 1.2-3: Principe général de l'approche de prétraitement

1.2.5.1.1. Les index de jointure spatiale de (Zeitouni et al)

L'approche proposée par *(Chelghoum, et al., 2002) (Zeitouni, 2006)* vise pour objectif la fouille de données géo-spatiales multi-thème c'est-à-dire l'extraction de tendances entre plusieurs couches spatiales. Pour cela, l'approche est structurée en deux (2) principales étapes (voir Figure 1.2-4) :

- ***Étape 1 - index de jointure spatiale*** : il s'agit de la construction d'une structure – une table dans le cas d'une base de données relationnelles - qui stocke les relations de proximité entre deux couches. Le calcul des indices se fonde sur les relations métriques et topologiques. Bien que le coût du calcul soit équivalent à celui de la réalisation d'une jointure spatiale sur critère de distance, l'index permet de gagner en temps lors des autres phases de la fouille de données.

- ***Étape 2 – formulation du problème en fouille multi-relations*** : après le calcul des index de jointure spatiale, la suite des opérations peut se résumer en une fouille multi-tables; ce qui pose problème pour les algorithmes traditionnels qui n'acceptent qu'une seule table en entrée. Pour remédier à cette situation, les auteurs proposent deux options. Soit transformer les algorithmes traditionnels

afin qu'ils puissent traiter ce type de données; ou bien ramener les données en une seule table et utiliser alors les algorithmes classiques.

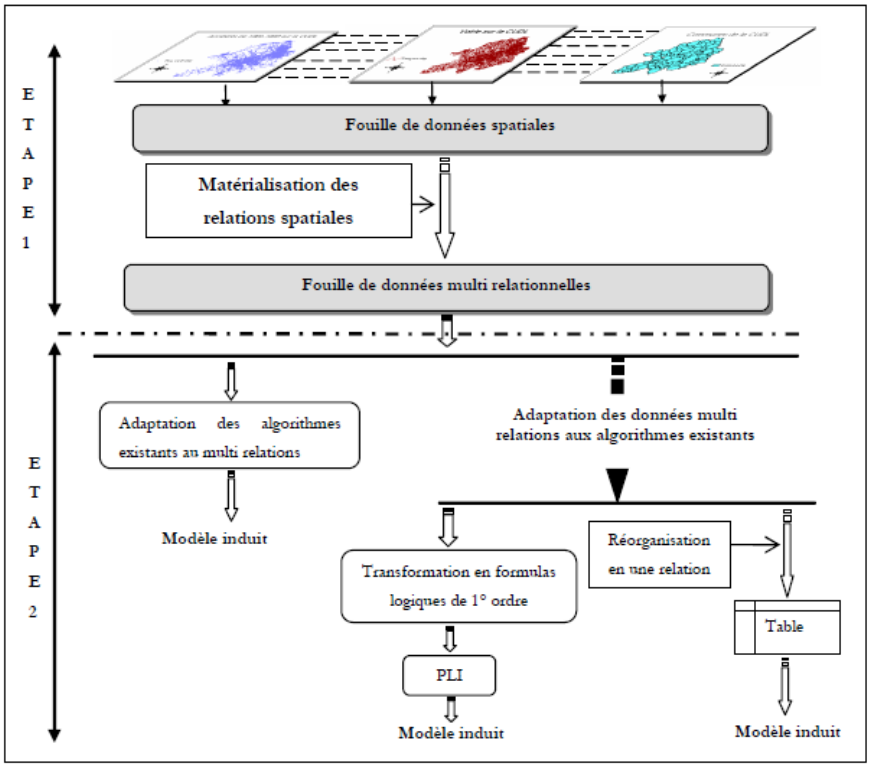


Figure 1.2-4 : Approche de fouille multi-tables tiré de *(Zeitouni, 2006)*

1.2.5.1.2. Le Framework géo-spatiales de (Bogorny et al)

Le Framework proposé par *(Bogorny, et al., 2005) (Bogorny, et al., 2006)* vise à fournir un module de prétraitement de données géo-spatiales afin d'enrichir WEKA (cf. Annexe A), un outil open-source de fouille «classique ». Ce Framework peut se subdiviser en trois (3) modules : le module d'accès aux données, celui de prétraitement et enfin celui de fouille de données (voir Figure 1.2-5).

Le module de prétraitement permet de réaliser des opérations sur les entités géo-spatiales en entrée sur la base des relations que l'utilisateur désire prendre en compte dans la fouille. L'ensemble des transformations effectuées est réalisé en mettant à profit les métadonnées des tables géométriques et le langage SQL spatial. Une particularité de cet outil est sa capacité à prétraiter les données selon leur niveau de granularité. Ainsi donc,

l'utilisateur choisit de faire un prétraitement avec un niveau de granularité de type « Instance » dans lequel les chaque tuples est considérés. Avec le niveau de granularité « Type », seul le type de géométrie est considéré.

Le résultat de ces opérations est ensuite transformé pour obtenir un format compatible avec l'algorithme de fouille.

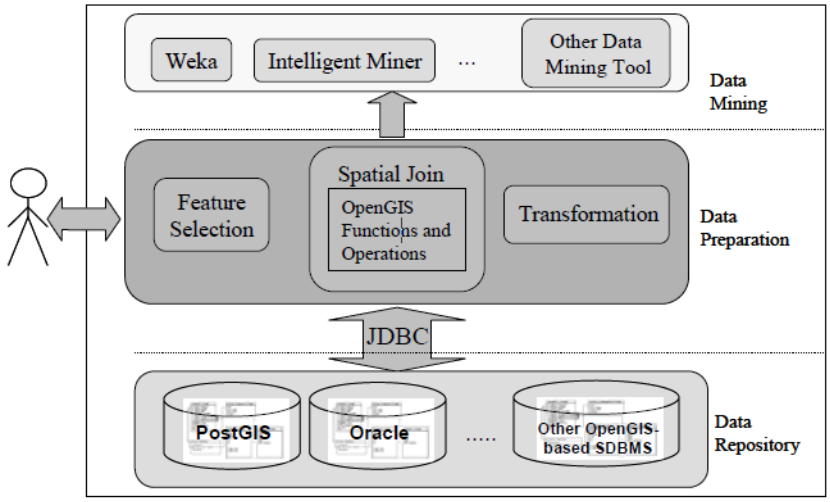


Figure 1.2-5 : Framework de prétraitement géo-spatiales tirée de *(Bogorny, et al., 2005)*.

Dans la suite de leurs travaux, *(Bogorny, et al., 2007)* proposent un prétraitement des données partant d'une connaissance a priori du domaine et des contraintes spatiales. Cela permet par exemple l'élimination de règles trop évidentes lors de l'extraction de règles d'association spatiale (cf. *(Bogorny, et al., 2005)* pour plus de détail).

1.2.5.1.3. Relations spatiales et logique de premier ordre de (Appice et al)

(Appice, et al., 2003) Propose une approche à la fouille de données géo-spatiales qui consiste à extraire les relations spatiales et les stocker dans une base de données déductive. Il s'agit pour les auteurs de mettre à profit la puissance offerte par la programmation

logique inductive[10]. Ainsi donc, en stockant en exploitant les bases de données déductives, non seulement l'utilisateur stocke les relations spatiales sous forme de prédicats mais également, il a la possibilité de définir les connaissances a priori du domaine sous forme de contraintes, de hiérarchie de concepts, etc.

(Appice, et al., 2000) ont mis en place un algorithme dénommé SPADA (Spatial Pattern Discovery Algorithme) a été développé afin de formaliser cette approche. L'algorithme a été par la suite couplé avec une base de données spatiales, en l'occurrence ORACLE Spatial, grâce à un module nommé FEATEX qui permet l'extraction des relations spatiales au travers du langage SQL.

1.2.5.2. Approché basée sur le traitement dynamique de l'information spatiale

Au regard des inconvénients de la première approche qui exige de la ressource temps et entraîne un stockage redondant. Mais aussi, afin de bien cerner la particularité de la fouille de données géo-spatiales, des auteurs se sont lancés sur une autre voie, celle de la mise en œuvre d'outils traitant dynamiquement les corrélations entre entités géo-spatiales.

L'objectif principal de cette approche est la prise en compte de la composante géo-spatiale sans passer par une phase de prétraitement ou de stockage préalable sur un support tiers (voir Figure 1.2-6). Autrement dit, il s'agit de mettre en œuvre des algorithmes ou des outils flexibles qui traitent dynamiquement la dite composante lors du processus de fouille *(Rinzivillo, et al., 2008)*.

En termes de comparaison, cette approche offre bien plus d'avantages comparativement à la première même si elle nécessite beaucoup de ressources en R&D. Aussi les outils issus de cette approche souffriront, pendant un certain temps, d'une certaine immaturité. Comme avantages *(Klösgen, et al., 2002)* note :

- la flexibilité
- la sélection dynamique des relations lors de la fouille
- la réduction des jointures spatiales et donc une diminution des ressources exigées pour la computation

[10] Traduction libre d'ILP (Inductive Logic Programming)

L'inconvénient majeur de cette approche est qu'elle est beaucoup focalisée sur la phase de modélisation (cf. section 1.2.2.1) alors que les autres étapes du processus de fouille sont toutes aussi importantes que l'étape ci-dessus mentionnée.

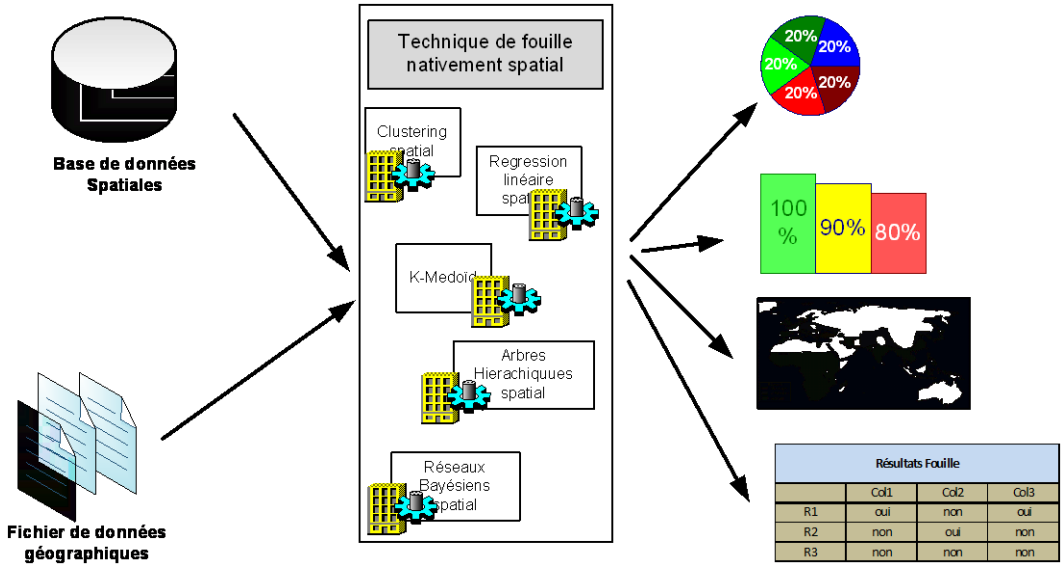


Figure 1.2-6: Principe générale de l'approche de traitement dynamique de l'information spatiale

1.2.5.2.1. GeoMiner

(Han, et al., 1997) Ont développé GeoMiner, l'un des premiers systèmes de fouille de données géo-spatiales. Il s'agit d'une extension de DBMiner qui est un outil de fouille classique. GeoMiner est un outil complet en ce sens qu'il contient divers modules qui interviennent dans un domaine bien précis de l'analyse décisionnelle (cf. Figure 1.2-7). En effet, on note les modules suivants :

- module pour la construction de cube
- module pour l'analyse en ligne OLAP
- module pour la fouille de données

GeoMiner intègre en plus, un langage d'interrogation géographique dénommé GMQL. Pour ce qui est du module assurant la fouille de données, il implémente plusieurs tâches dont : la géo-caractérisation, la géo-association et la géo-comparaison.

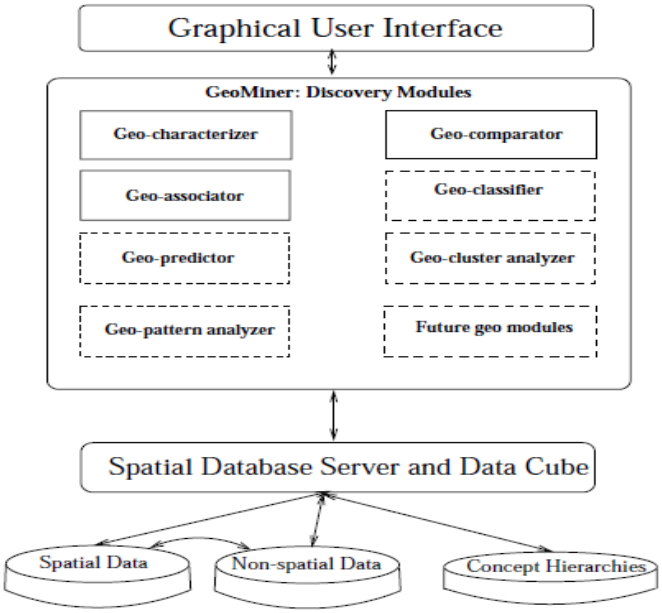


Figure 1.2-7: Architecture GeoMiner tiré de *(Han, et al., 1997)*

1.2.5.2.2. SubGroupMiner

La contribution de *(Klösgen, et al., 2002)* dans le domaine de la fouille de données géo-spatiales a consisté en la mise en place de SubGroupMiner qui est un outil permettant la découverte de sous groupes spatiaux (spatial Subgroup). Les sous groupes spatiaux réfèrent à des sous objets d'analyse décrits par des expressions d'un langage de requête; par exemple un sous ensemble des districts intersectés par un cours d'eau. L'efficience de cet outil réside dans le fait qu'il intervient à diverses étapes du processus de la fouille de données (l'accès aux données, l'analyse, l'évaluation, l'interprétation).

1.2.5.2.3. Neighborhood Graph

(Ester et al) proposent des primitives basées sur les relations spatiales que les entités entretiennent *(Ester, et al., 1997) (Ester, et al., 1999)*. L'idée sous-tendue par les auteurs est que de telles primitives réduiraient le temps de calcul qu'exige le traitement des données géo-spatiales et de ce fait aider au développement de nouveaux algorithmes spatiaux.

Le développement de ces primitives s'est basé sur la notion de graphe de voisinage qui est un ensemble de chemins et de nœuds traduisant les relations de voisinage entre entités géo-spatiales (métrique, topologique, directionnelle). Ainsi dans un graphe de

voisinage, plus la distance (le nombre de chemin à parcourir) entre deux nœuds est minime, plus ils s'influencent mutuellement.

Afin d'intégrer de façon efficiente ces primitives au sein de systèmes de gestion de base de données, les auteurs proposent l'utilisation d'indices de voisinage. L'idée est d'avoir une structure pré-calculée et de ce fait éviter chaque fois l'accès aux entités géo-spatiales elles-mêmes.

1.2.5.2.4. SPIN!

(May, et al., 2003) Abordent la fouille de données géo-spatiales sous l'angle de l'efficience, de la gestion de la montée en charge, de l'accès multi-utilisateurs, de la robustesse, de la sécurité. C'est dans ce sens qu'ils proposent un outil dénommé SPIN! qui est une plateforme extensible N-tiers de fouille de données basée sur J2EE. L'efficience de SPIN! réside dans sa capacité à effectuer de la fouille sur plusieurs serveurs. En effet, SPIN! est constitué de plusieurs clients organisés autour d'un serveur d'application et sur ces différents clients, plusieurs tâches de fouille peuvent être lancées en parallèle.

SPIN! Comme en témoigne son architecture (voir Figure 1.2-8) inclut plusieurs composants dont :

- Un serveur d'application dont la tâche est la gestion des clients, des tâches d'analyse, de l'accès à la base de données et de la persistance.
- Des clients qui sont des applications autonomes accédant à leur session de travail sur le serveur via java RMI ou Corba.
- Un visualisateur spatial des résultats de la fouille au niveau de chaque client.

En termes d'algorithmes spatiaux, SPIN intègre des algorithmes tels SubGroupMiner, d'autres algorithmes spatiaux éprouvés mais offre également des tâches de fouille de données « traditionnelles ».

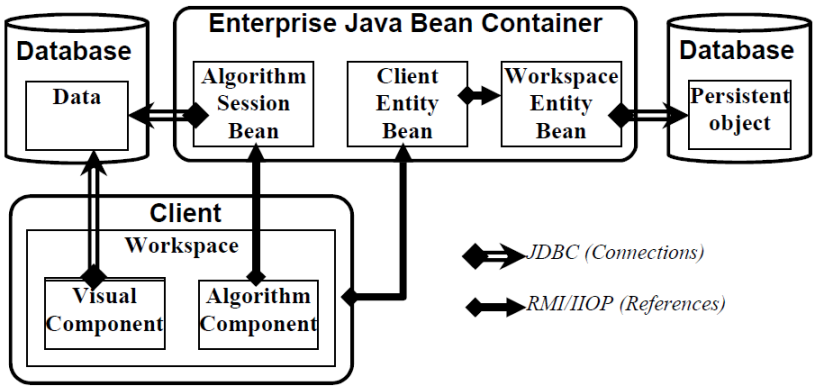


Figure 1.2-8: Architecture de la plateforme SPIN! Tiré de *(May, et al., 2003)*

1.2.5.2.5. INGENS

(Malerba, et al., 2000) ont développé INGENS (INductive GEographic iNformation System), un système inductif de découverte de connaissances spatiales basé sur la logique de premier ordre. Il s'agit en réalité de l'extension d'un outil SIG avec des algorithmes de fouille de données afin d'aider à l'extraction de tendances au sein de données raster ou vecteur. Le système est construit autour de diverses couches (voir Figure 1.2-9) dont les plus importantes sont :

- Map Storage subSystem : permet la mise à jour des données extraites du référentiel de carte ;
- Map convertor : permet l'acquisition des données géo-spatiales de type divers (vectoriel ou raster)
- Map editor : permet l'édition des données acquises ;
- Map descriptor : gère la génération automatique de prédicats de premier ordre à partir des données géo-spatiales ;
- Query interpreter : permet à l'utilisateur de formuler des interrogations sous forme de prédicats de premier ordre ;
- Learning server : permet l'induction des connaissances sur la base de celles disponibles dans le référentiel.

Le système permet également, un accès selon le profil de l'utilisateur. On note différents niveaux d'accès allant du niveau d'administrateur à celui d'utilisateur

occasionnel en passant par le profil d'utilisateurs privilégiés et celui de maintenancier du système.

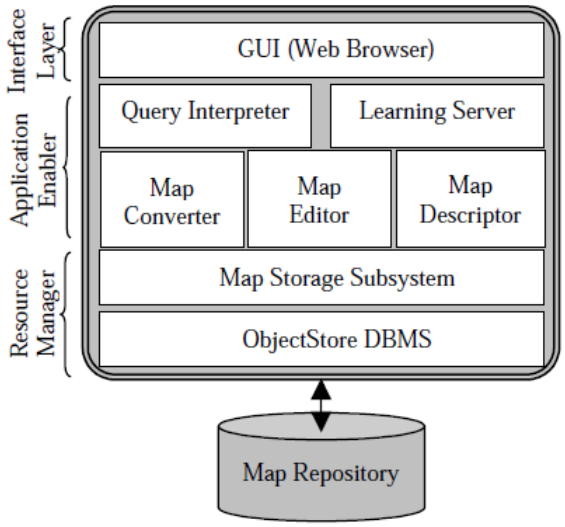


Figure 1.2-9: Architecture INGENS tiré de *(Malerba, et al., 2000)*

Le système INGENS a par la suite fait l'objet d'une amélioration par le développement de SDMOQL (Spatial Data Mining Object Query Language), un langage d'interrogation objet *(Malerba, et al., 2002)*. Ce langage qui s'appuie sur DMQL (Data Mining Query Language) offre des primitives de fouille, permettant ainsi aux utilisateurs du système de formuler des requêtes sans se soucier de la technologie sous-jacente et de l'ordre d'exécution des tâches.

1.3. Problématique

La volonté d'apprendre de nos données a entrainé l'émergence d'un domaine de connaissances, la fouille de données géo-spatiales qui entend tirer partie de la spécificité de l'information géographique à des fins de décision plus éclairées. Pour ce faire, plusieurs approches ont été mises en œuvre *(Zeitouni, 2002)* *(Zeitouni, 2006)*. Ces approches, regroupées en deux (2) grandes catégories notamment les approches fondées sur le prétraitement des données et celle basées sur le traitement dynamique de l'information spatiale *(Klösgen, et al., 2002)*, tentent à leur façon d'apporter une réponse efficace à la problématique de la fouille de données géo-spatiales mais reste toutefois limitées et ce pour de nombreuses raisons. On note principalement:

L'absence d'outils intégrés de fouille de données géo-spatiales : la plupart des outils existant tant sur le marché que dans le domaine académique se focalisent exclusivement sur la fouille de données du point de vue modélisation i.e. le développement d'algorithmes pour extraire les connaissances. Or la fouille de données ne se résume pas exclusivement à l'étape de modélisation. Bien d'autres étapes existent notamment celles de compréhension et de préparation des données ainsi que l'évaluation des résultats et leur déploiement (cf. Section 1.2.2.1).

L'immaturité de certaines approches : En effet, une approche comme le développement – de solutions traitant dynamiquement la composante géo-spatiales quoique fort intéressant peut se relever complexe dans sa mise en œuvre principalement du fait de l'immaturité des outils jusque là existant. Aussi, cette complexité peut entrainer des retards dans la mise à disposition sur le marché, d'outils permettant d'effectuer de la fouille de données spatiales.

Le prétraitement des relations spatiales : L'altération des corrélations entre les entités géo-spatiales en procédant à un prétraitement préalable des relations spatiales afin d'obtenir des données de types classiques, même si elle permet la réutilisation des outils de fouille existant, peut devenir très vite difficile à mettre en pratique lorsque le nombre de relations à prendre en compte augmente. Aussi, des problèmes liés d'une part à la visualisation cartographique des résultats de la fouille et d'autre part à la mise à jour lorsque de nouvelles relations spatiales sont à prendre en compte peuvent apparaître *(Klösgen, et al., 2002)*.

La diversité des relations spatiales : nombreux sont les outils de fouille de données géo-spatiales qui ne prennent pas en compte la diversité des relations spatiales existantes. En lieu et place, ces outils considèrent exclusivement la relation métrique (ou euclidienne) *(Ng, et al., 2002)* , même s'il est vrai que celle-ci sonne comme une mesure de similarité naturelle pour les données géo-spatiales. Pourtant, pour tirer amplement parti des connaissances dissimulées au sein de ces données, il est important de considérer d'autres types de relations que ces données peuvent entretenir (relation topologique, directionnelle, de reconnaissance de forme, etc.).

Les types de géométrie : certains outils de fouille de données géo-spatiales sont sélectifs par rapport au type de géométrie de l'information géographique en entrée *(Rinzivillo,*

et al., 2008). En effet, certains algorithmes sont efficaces dans le traitement de données ponctuelles tandis que d'autres sont performants quand il s'agit d'entités surfaciques. Pourtant, un bon outil de fouille de données devrait pouvoir traiter les données géo-spatiales indépendamment de leur dimension.

Montée en charge et haute dimensionnalité : Les données géo-spatiales, comparées aux données classiques, sont complexes *(Koperski, et al., 1996)* en ce sens qu'elles comportent une composante descriptive et géométrique. Également, ces données sont porteuses d'informations sur les relations avec d'autres entités géo-spatiales. L'ensemble combiné, fais naitre de la difficulté en termes de montée en charge et de haute dimensionnalité lors de la réalisation de tâches de fouille de données. Les tâches faisant intervenir les données géo-spatiales consomment donc plus de ressources notamment pour certains algorithmes où le temps de calcul peut se montrer très prohibitif.

Disponibilité des outils existants : le domaine de la fouille de données fait face à la non disponibilité de certains outils ailleurs que dans le domaine académique *(Bogorny, et al., 2006)*. Une large disponibilité pourrait entraîner une implication plus grande de la communication et donc permettre l'amélioration desdits outils.

À la vue des insuffisances ci-dessus mentionnées, la problématique abordée dans la présente étude est la **proposition d'une nouvelle approche de fouille de données qui intègre de façon transparente, complète et cohérente la composante spatiale**.

1.4. Objectifs

Au regard des limites mentionnées dans la section précédente, on est en droit de se poser quelques questions :

- Outre les approches précédemment évoquées, n'y a-t-il pas une autre manière de concevoir la fouille de données géo-spatiales? En d'autre termes, ne devront on pas réfléchir à tout autre façon d'aborder la fouille de données géo-spatiales? Si oui, quelles devront être les fondements de cette nouvelle approche?

- Ne peut-on pas songer à une « dé-complexification » de la fouille de données géo-spatiales ? De notre point de vue, les différentes approches sont pour le moins complexes et certainement à juste raison au regard de la complexité de l'information géo-spatiale. Mais ne pourrait-on pas songer à une « banalisation » du type géo-spatial? Faire en sorte que ce type soit disponible afin de pouvoir manipuler aisément cette composante tout au long du processus de fouille de données (prétraitement, analyse exploratoire, modélisation,…)
- Comment exploiter l'interdépendance des entités géo-spatiales au sein d'outils efficaces et éprouvés de fouille de données « traditionnelles »?
- Comment assurer une interopérabilité avec d'autres outils décisionnels. Il s'agit par exemple de permettre une interaction avec un entrepôt de données. Ce qui pourra assez nettement faciliter la fouille données *(Jambu, 1999)*. En effet, l'existence d'un entrepôt de données permet de gagner un temps substantiel ; temps qu'on aurait pu passer à la collection des données depuis différentes sources, leur intégration, leur nettoyage, leur agrégation et regroupement sous forme d'indicateurs *(Ester, et al., 1997) (Han, 1997)*.

L'objectif principal de la présente recherche est de proposer une nouvelle approche intégrée de fouille de données qui supporte de façon cohérente, complète et transparente la composante spatiale. Cette nouvelle approche devrait permettre le support de la géométrie à toute les étapes du processus de fouille de données avec comme principale hypothèse la réutilisation des outils existant de fouille de données « traditionnelle ».

L'atteinte de l'objectif principal, à savoir la proposition d'une nouvelle approche intégrée de fouille de données, passe par la réalisation des objectifs secondaires suivant :

Sous-objectif 1 : définir l'approche qui consistera à l'intégration de la composante spatiale: il s'agit à ce stade de jeter les bases de ce que sera la nouvelle approche de

fouille de données géo-spatiales. Il faudra donc définir ce que sera cette nouvelle approche et ce qu'elle entend réaliser.

Sous-objectif 2 : concevoir et décrire un cadre d'intégration de la composante spatiale: qui décrit comment est ce que l'approche que nous proposons entend tirer parti des corrélations entre entités géo-spatiales en mettant à profit l'existant tant en matière de fouille « traditionnelle » que de bibliothèques de traitement de données géo-spatiales.

Sous-objectif 3 : développer un outil intégré de fouille de données géo-spatiales. Il s'agit de proposer une preuve de concept de l'approche que nous aurons préalablement décrite. Pour cela, plusieurs sous étapes sont à effectuer au niveau de cet objectif.

- ***Étape 1* : *choix de l'outil de fouille de données*** : l'objectif ici est d'une part de choisir un outil de fouille qui puisse être facilement étendu et d'autre part qui supporte une grande variété d'algorithmes de fouille de données.
- ***Étape 2 : choisir les types d'algorithmes de fouille à enrichir :*** Il s'agit de voir au niveau des différentes types d'algorithmes existants (clustering, d'association, de régression, etc.), lesquels sied le mieux à un enrichissement spatial.

Sous-objectif 3 : Tester et valider l'approche avec des données réelles :

Pour cela, des données sur la criminalité de la ville de San Francisco seront utilisées ceci pour la validité des algorithmes enrichis spatialement; c'est à dire voir si ceux-ci fournissent des résultats conformes avec la réalité du terrain et enfin démontrer l'efficacité du couplage fouille « traditionnelle » et composante géo-spatiale.

1.5. Méthodologie

Afin d'atteindre les objectifs cités ci haut, il est nécessaire d'adopter une méthodologie qui nous permettra à terme de produire une approche et un outil qui puisse prendre efficacement en compte la composante géo-spatiale lors d'une fouille de données.

1.5.1. Choix d'une démarche agile – AUP (Agile Unified Process)

Avant d'entrée en profondeur dans les détails de cette méthodologie, il est important de noter que la démarche qui sous-tendra celle-ci sera d'emblée une démarche informatique. De préférence, nous adopterons une démarche agile qui à la fois sera itérative et incrémentale à l'image du processus AUP (Agile Unified Process). AUP est une démarche qui provient du RUP (Rational Unified Process) et par conséquent, hérite des avantages de cette dernière. Mais contrairement à cette dernière, elle prône une simplicité tout au long de la démarche et l'adoption de cycles d'itérations courts *(Amber, 2005)*. AUP est composée de quatre (4) principales phases dont le contenu est personnalisable à souhait selon le contexte:

Conception : il s'agit à ce stade de comprendre le projet, d'en définir l'envergure. Également à ce stade, on prépare l'environnement du projet, tout en évaluant les risques potentiellement encourus. Pour relativiser cette phase par rapport au contexte présent, il s'agira à cette étape de comprendre le domaine de la fouille de données en premier lieu. En second lieu, il s'agit de comprendre le domaine géo-spatial et la spécificité de ce dernier vis-à-vis de la fouille « traditionnelle ». De façon plus pragmatique, il s'agit de mener une revue de littérature afin de comprendre les tendances actuelles en matière de fouille de données « traditionnelles » et géo-spatiales. Le principal livrable à ce stade constitue la proposition de recherche qui pour rappel est un document qui jette les bases de la recherche future à mener. Ce document, en plus de décrire brièvement l'état de l'art, situe la problématique, annonce les objectifs de l'étude, définit la méthodologie et donne un chronogramme des différentes activités à mener dans le cadre de l'étude.

Élaboration : à ce stade, on définit l'architecture du système. Cette phase débute par un léger retour arrière vers la phase de démarrage pour faire un tour d'horizon sur les différents outils de fouille de données existants. Par la suite, il sera effectué une étude détaillée des fonctionnalités de l'outil qui devrait être enrichi avec la composante géo-spatiale. Également, au cours de cette phase, les différentes bibliothèques de traitement de la donnée géo-spatiale devraient être étudiées afin de voir dans quelle mesure elles peuvent être intégrées dans l'outil à enrichir. Le livrable attendu à ce niveau devrait résumer

l'architecture globale du système. Ce document devrait se composer d'un ensemble de diagrammes UML en charge de décrire l'architecture selon divers points de vue.

Construction : il s'agit de l'implémentation du système à développer c'est-à-dire mettre en pratique ce qui a été annoncé dans le livrable qui retrace l'architecture du système.

Transition : il s'agit de valider et de déployer l'outil résultant. Il faudra également s'assurer que l'outil développé fournit toujours les résultats attendus quand il y'a une montée en charge. Cela signifie que l'outil reste performant quand croît le volume et la dimensionnalité des données.

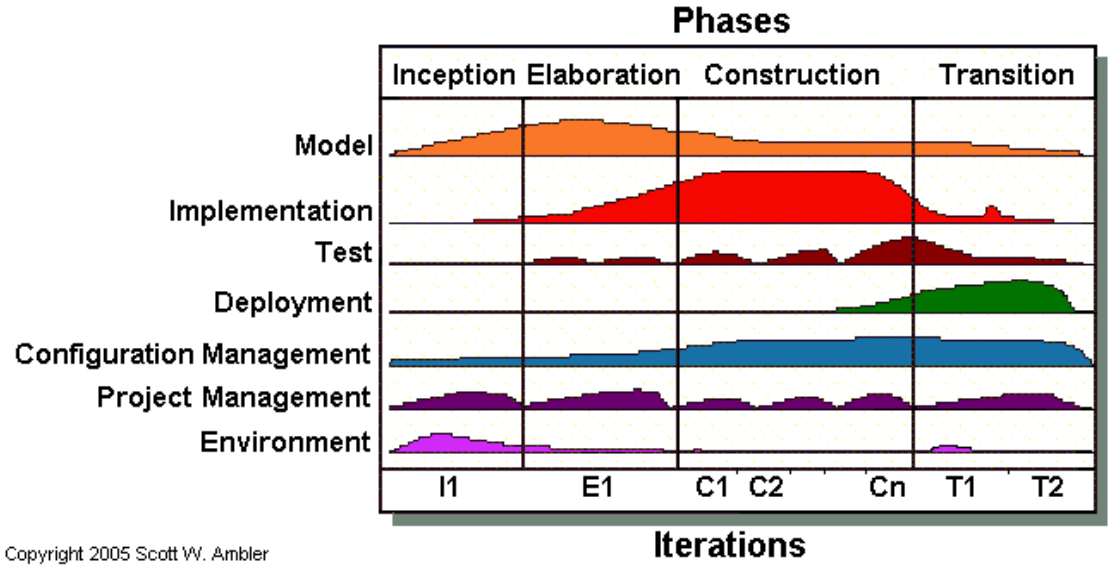


Figure 1.5-1: Les phases de AUP tiré de *(Amber, 2005)*

Il est important de revenir sur le cycle itératif et incrémental de la démarche choisie. Cela est d'autant plus important que les différentes tâches à mener tout au long des phases décrites plus haut ne sont pas réalisées en une fois et avec le même niveau d'effort. Tout comme le montre le schéma ci-dessus (voir Figure 1.5-1), chacune des tâches est réalisée au cours d'une phase donnée mais pas avec la même intensité. Ainsi donc, la revue de littérature se poursuivra jusqu'à la fin du projet. L'implémentation consistera quant à elle en une somme d'incréments dont l'ensemble constituera le produit final.

1.5.2. Description des tâches - Diagramme d'activité

Le diagramme d'activité UML ci-dessous (voir Figure 1.5-2) résume les principales activités qui prendront cours lors de la réalisation de ce projet. Ce digramme décrit l'ordre dans lequel les activités auront lieu sans toutefois donner une idée de quand est ce qu'elles seront menées. Le diagramme contient les activités suivantes :

- ***Acquisition de notions sur le domaine géo-spatial*** : il s'agit de l'ensemble des activités menées afin de s'imprégner du domaine de la géomatique. Cela concerne notamment les cours concernant la géomatique et ses référentiels ainsi que des notions d'analyses spatiales.
- ***Acquisition de notions sur la fouille« traditionnelle »*** : il s'agit d'un tour d'horizon sur ce qu'est la fouille de données et le processus de fouille (KDD) de façon générale. Il est important d'avoir une idée de ces deux notions avant de plonger dans la spécificité de la fouille géo-spatiale.
- ***Récupération des données de test*** : il s'agit de récupérer les données sur lesquelles les tests devront porter.
- ***Revue de littérature sur la fouille géo-spatiale*** : il s'agit de voir en détail la documentation en rapport avec la fouille géo-spatiale afin de comprendre d'une part sa spécificité et les tendances actuelles dans le domaine.
- ***Choix d'une classe d'algorithmes à implémenter*** : plusieurs classes d'algorithmes existent au niveau de la fouille de données (classification, association, régression,…). Bien que l'objectif de chacune de ces classes soit la mise à nu d'informations potentiellement utiles et implicites, chacune possède ses caractéristiques propres. Vu qu'on ne peut pas passer en revue toutes ces classes, au regard du temps qui nous est imparti dans le cadre de ce travail de maîtrise, il est nécessaire de choisir une ou deux classes et d'implémenter quelques algorithmes de ces classes.
- ***Étude d'un outil de fouille de données***: il s'agit d'étudier les outils de fouille de données et de choisir celui dans lequel la composante spatiale sera intégrée.
- Choix de l'outil : il s'agit de la résultante de l'activité précédente.

- ***Étude des bibliothèques géo-spatiales libres*** : il s'agit d'étudier les différentes bibliothèques géo-spatiales afin de voir laquelle ou lesquelles pourraient être intégrées dans l'architecture de l'outil à mettre en œuvre.
- ***Conception de l'architecture*** : à l'aide des différentes notions accumulées, une ébauche de l'architecture peut être réalisée.
- ***Implantation de l'outil*** : à ce niveau, il s'agit de procéder à l'implémentation de la nouvelle architecture.
- ***Validation de l'approche et test de l'outil*** : il s'agit de tester d'une part la validité des algorithmes enrichis et d'autres part de tester la robustesse de l'outil.
- ***Rédaction du mémoire et d'un article***

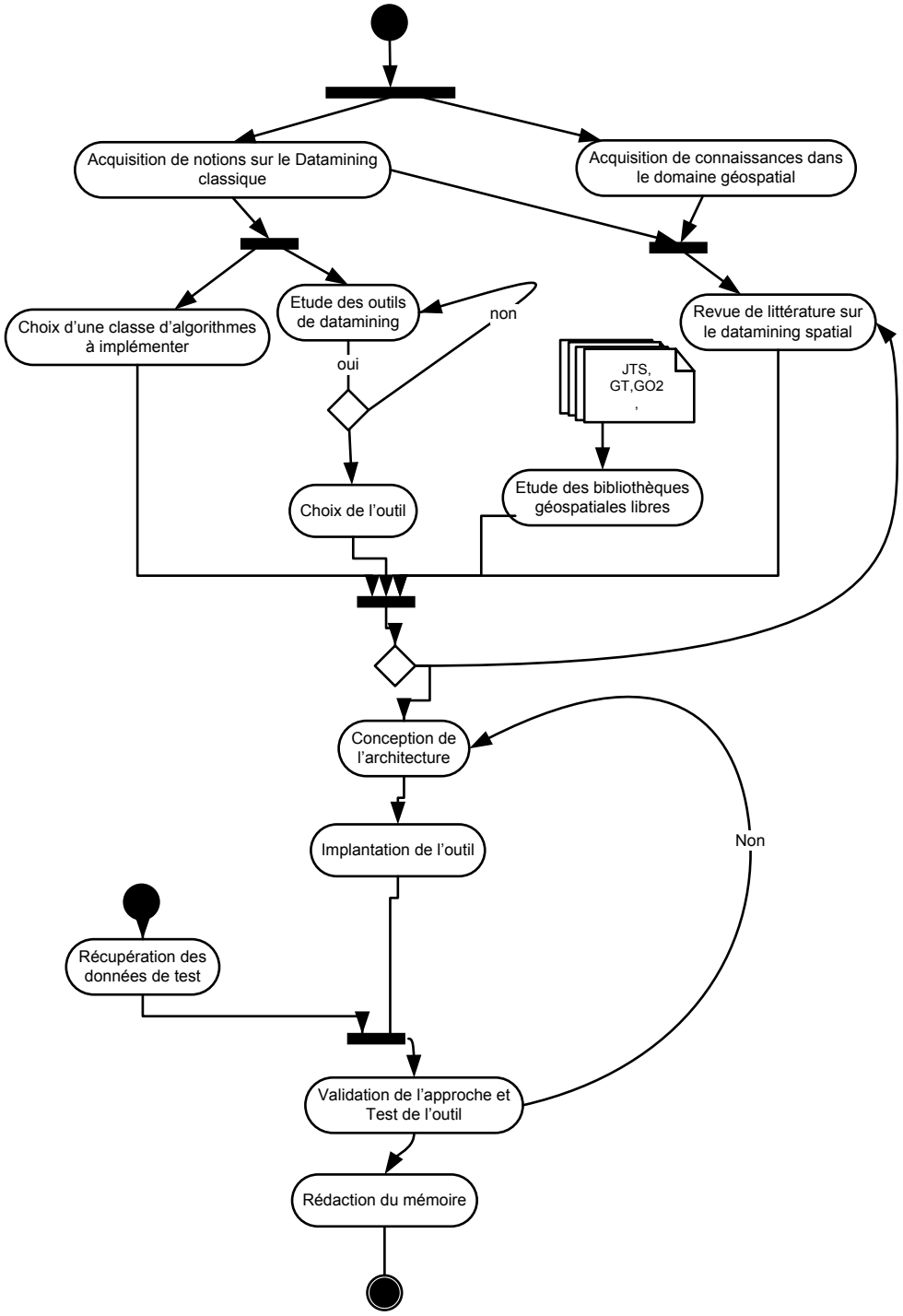


Figure 1.5-2: Diagramme d'activité UML

1.6. Conclusion

L'Extraction de Connaissances à partir des Données est de nos jours l'un des moyens de plus en plus utilisés pour apprendre de nos données. Alors usitée dans divers domaines, cette science s'est transposée dans le monde géo-spatial et s'est érigée en un domaine à part entière, l'Extraction de Connaissances à partir de Données Géographiques ou GKD, afin de prendre en compte la spécificité de ce domaine qui pour rappel, est intrinsèquement liée à la nature de l'information géo-spatiale.

Le GKD est de ce fait, un domaine relativement jeune qui possède bien de défis à relever dont entre autre la nécessité de faire face à la complexité de l'information géo-spatiale et celle de la prise en compte de l'interdépendance entre entités géographiques.

Pour faire face à ces défis, diverses approches, qu'on peut catégoriser en deux(2) grandes, ont été développées : d'une part, on note celle qui préconise le prétraitement de l'information géo-spatiale comme préalable à l'utilisation des outils de fouille de données « traditionnelle ». D'autre part, l'approche qui prône la mise en œuvre d'outil qui traitent dynamiquement de la composante géo-spatiale.

S'il est vrai que l'approche dynamique de la composante géo-spatiale permet une meilleure exploitation des corrélations spatiales, l'approche de prétraitement, même si elle entraine une décomposition de l'information géo-spatiale encourage plutôt l'utilisation des outils de fouille de données classiques. De notre point de vue, il est important de tirer parti des avantages qu'offrent les deux approches précédemment citées. Cela permettra de mettre à profit l'existant d'outils et d'algorithmes éprouvés de la fouille de données classiques.

C'est à cela que s'attache la présente étude; c'est-à-dire réfléchir à une nouvelle approche de la fouille de données géo-spatiales qui non seulement tire avantages des précédentes approches mais va plus loin en permettant la manipulation de la composante géo-spatiale à toutes les étapes de la fouille de données. Et atteindre cet objectif passe par l'intégration de cette composante de façon cohérente, complète et transparente.

1.7. Structure du document

Le présent document est structuré en quatre (4) chapitres.

Le premier se veut un chapitre (cf. Chapitre 1 – Mise en contexte) introductif dans lequel le lecteur est sensibilisé sur le contexte dans lequel la recherche s'inscrit, l'état de l'art sur la fouille de données géo-spatiales, la problématique, les objectifs ainsi que la méthodologie à adopter en vue d'atteindre les objectifs fixés.

Le deuxième chapitre (cf. Chapitre 2 - Une nouvelle approche intégrée pour la fouille de données géo-spatiales) apporte une réponse à l'objectif principal de l'étude à savoir a la proposition d'une approche intégrée qui tient compte de la composante spatiale à différentes étapes d'une fouille de données. Dans ce chapitre, nous décrivons différentes relations géo-spatiales qui constituent le fondement de l'approche (cf. Section 2.3). Par la suite, en vue de décrire l'implémentation de cette approche, un cadriciel est proposé (cf. Section 2.3.4.5).

Le troisième chapitre (cf. Chapitre 3 - Implémentation et test de GeoKNIME un nouvel outil de fouille de données géo-spatiales) décrit l'implémentation de la composante spatiale au sein d'un outil de fouille de données open-source. Il s'agit à ce niveau de décrire la preuve de concept mis en œuvre afin d'appuyer l'approche théorique que nous avons proposée dans le chapitre précédent. On y décrit l'architecture du nouvel outil enrichi, l'implémentation du type géo-spatiale (cf. section 3.3.1) ainsi que quelques algorithmes enrichis « spatialement ». Dans ce chapitre, nous décrivons également les tests effectués sur l'outil

Enfin dans le quatrième chapitre (cf. Chapitre 4 – Conclusions et perspectives), nous apportons une conclusion à l'étude et évoquons des perspectives potentielles d'évolution.

Chapitre 2 - Une nouvelle approche intégrée pour la fouille de données géo-spatiales

2.1. Introduction

L'Extraction de Connaissances à partir de Données (ECD) est un domaine d'étude qui a pour finalité l'extraction de connaissances au sein d'entrepôts de données, connaissances qui pourront potentiellement servir à la prise de décisions *(Agrawal, et al., 1993)*. Bien qu'existant depuis plusieurs années, ce domaine a été transposé quelque peu « tardivement » dans le monde spatial pour donner naissance à l'Extraction de Connaissances à partir de Données Géographiques. L'ECDG (GKD) est donc un domaine relativement jeune, qui fait face à de nombreux défis au regard de sa spécificité – intrinsèquement liée à la nature des données géo-spatiales (cf. section 1.2.3.2).

Pour rappel, plusieurs outils ont été mis en œuvre afin de traiter cette spécificité. Ces outils peuvent être catégorisés selon diverses approches dont celles de prétraitement versus celles de traitement dynamique des données géo-spatiales.

La première de ces approches, celle de prétraitement, vise une extraction préalable et explicite de relations spatiales entre entités géo-spatiales pour par la suite utiliser ces données prétraitées en entrée d'algorithmes « traditionnels » de fouille de données *(Bogorny, et al., 2005) (Zeitouni, 2006)*.

La seconde approche préconise plutôt un traitement dynamique de l'information spatiale grâce à des algorithmes dédiés exclusivement à la donnée géo-spatiale ou au travers d'outils permettant un traitement « à la volée » *(Ester, et al., 1999) (Han, et al., 1997)*.

Les limites objectives des approches évoquées plus haut - immaturité pour les unes, stockage redondant pour les autres et surtout l'absence d'outils intégrés dédiés à la fouille de données spatiales - pour ne citer que celles-ci (cf. *(Klösgen, et al., 2002)*) – combiné à la grande quantité d'outils performants de fouille de données « traditionnelle » existants, nous amène à nous questionner sur d'autres approches possibles de fouille de données géo-spatiales notamment celles permettant de faire cohabiter au sein d'un même outil les

deux(2) variantes de fouille (spatiale et non spatiale). En effet, autant l'information spatiale devient de plus en plus présente dans nos systèmes d'information, autant l'information non-spatiale reste tout aussi incontournable. D'où la nécessité de s'intéresser à une intégration possible de la composante géo-spatiale au sein d'un outil existant de fouille de données, ce à quoi s'attache la présente recherche.

Loin de constituer, tout comme dans la première approche - de fouille géo-spatiale - en une simple extraction et exposition des relations entre entités géo-spatiales sous forme d'attributs de type classique (numérique, chaine de caractère, booléen, etc.), notre approche sonne comme un compromis des deux (2) précédentes : Elle se donne pour objectif principal la prise en compte de la composante géo-spatiale dans sa totalité (l'entité et ses relations). À termes, elle vise principalement:

- la reconnaissance du type géo-spatial au même titre que les types de données classiques : cela devrait permettre une intégration aisée et une prise en compte de ce type au sein des algorithmes de fouille « traditionnelle » ; mais également une disponibilité de la composante géo-spatiale au niveau des autres phases du processus de découverte de connaissances (compréhension des données, analyse exploratoire, etc.).
- la prise en compte des différentes relations que peuvent entretenir les entités géo-spatiales en les considérant soit comme des mesures de similarité, soit comme des facteurs pouvant servir à effectuer une discrimination au sein d'un ensemble de données (exemple : dans la génération d'un arbre de décision).

L'objectif du présent chapitre est d'abord de décrire la nouvelle approche intégrée de fouille de données permettant de répondre à la problématique énoncée dans les sections précédentes. Ensuite nous proposerons un cadre (framework) décrivant comment l'approche proposée prend en compte la composante spatiale.

Pour ce faire, nous aborderons dans un premier temps les relations entre entités géo-spatiales qui constituent le cœur de notre approche. Par la suite, nous aborderons le cadre que nous proposons en décrivant les différentes couches le constituant.

2.2. Une approche intégrée de fouille de données

Les précédentes approches de fouille de données géo-spatiales ont montré leurs limites quant au traitement efficace de la composante spatiale. Au-delà des inconvénients propres à chaque approche (cf. section 1.2.5), le dénominateur commun de ces insuffisances est l'incapacité de prise en compte de la composante spatiale à toutes les étapes de la fouille de données (cf. section 1.2.2.1). Au regard de cela, nous avons proposé une approche assurant le support de la composante spatiale à toutes les stades de la fouille de données; et cela de façon cohérente, complète et transparente (cf. Figure 2.2-1).

La nécessité d'une approche assurant le traitement de la composante spatiale à toutes les étapes de la fouille de données s'explique par le fait que l'extraction des connaissances ne se limite plus à l'étape de modélisation (cf. processus CRISDM section 1.2.2.1). En effet, de par le passé, la fouille de données se résumait principalement à l'utilisation de techniques et algorithmes en vue d'extraire la connaissance. Or il a été prouvé que la fouille de données ne se limite pas seulement à l'utilisation d'algorithmes et méthodes *(Kurgan, et al., 2006) (Miller, et al., 2001)*. Les étapes en amont et en aval sont toutes aussi importantes : comprendre le domaine et les données, les préparer convenablement ainsi que la visualisation des résultats, leur évaluation et déploiement sont d'une grande importance dans l'extraction de connaissances et par extension dans le support du processus de décision dans nos organisations. D'où la nécessité pour la fouille de données géo-spatiales - qui est jeune comparée à la fouille traditionnelle – d'adopter cette vision qui consiste à voir l'extraction de connaissances à partir d'entrepôt de données comme un processus.

Bien des éditeurs de logiciels de fouille de données ont compris cet état de fait. Ainsi, on note de plus en plus sur le marché et dans le monde académique des outils de fouille de données classiques assurant des fonctionnalités couvrant les différentes étapes de la fouille de données. Dans la suite de l'étude, nous mettrons d'ailleurs à contribution ces outils en vue d'assurer la manipulation de la composante géo-spatiale à toutes les phases du processus d'extraction de connaissances (cf. Chapitre 3).

Vis-à-vis des différentes approches existantes, la proposition d'une nouvelle approche intégrée, cohérente et complète de la composante géo-spatiale est justifiée parce que permet de préserver les corrélations spatiales. Par ailleurs des approches « hybrides » comme celle que nous proposons, vont de plus en plus émerger du fait de la grande quantité d'outils de fouille de données disponibles ainsi que la nécessité de préserver les relations entre entités géographiques. En effet, il sera plus facile de mettre à contribution les outils existants que de développer de nouveaux outils de fouille de données géo-spatiales ou de préférer une approche de prétraitement. Parce que, malgré l'apparente facilité de mise en œuvre des approches dites de prétraitement, elles exigent beaucoup d'efforts dans le prétraitement (cf. section 1.2.5.1) et entrainent une dénaturation des relations entretenues par les entités géo-spatiales (cf. section 1.2.3.2). D'un autre côté, les approches de traitement dynamique restent limitées dans la mesure où il est difficile de trouver un outil traitant tous les types de géométries, de relations et couvrant les différentes phases de la fouille de données.

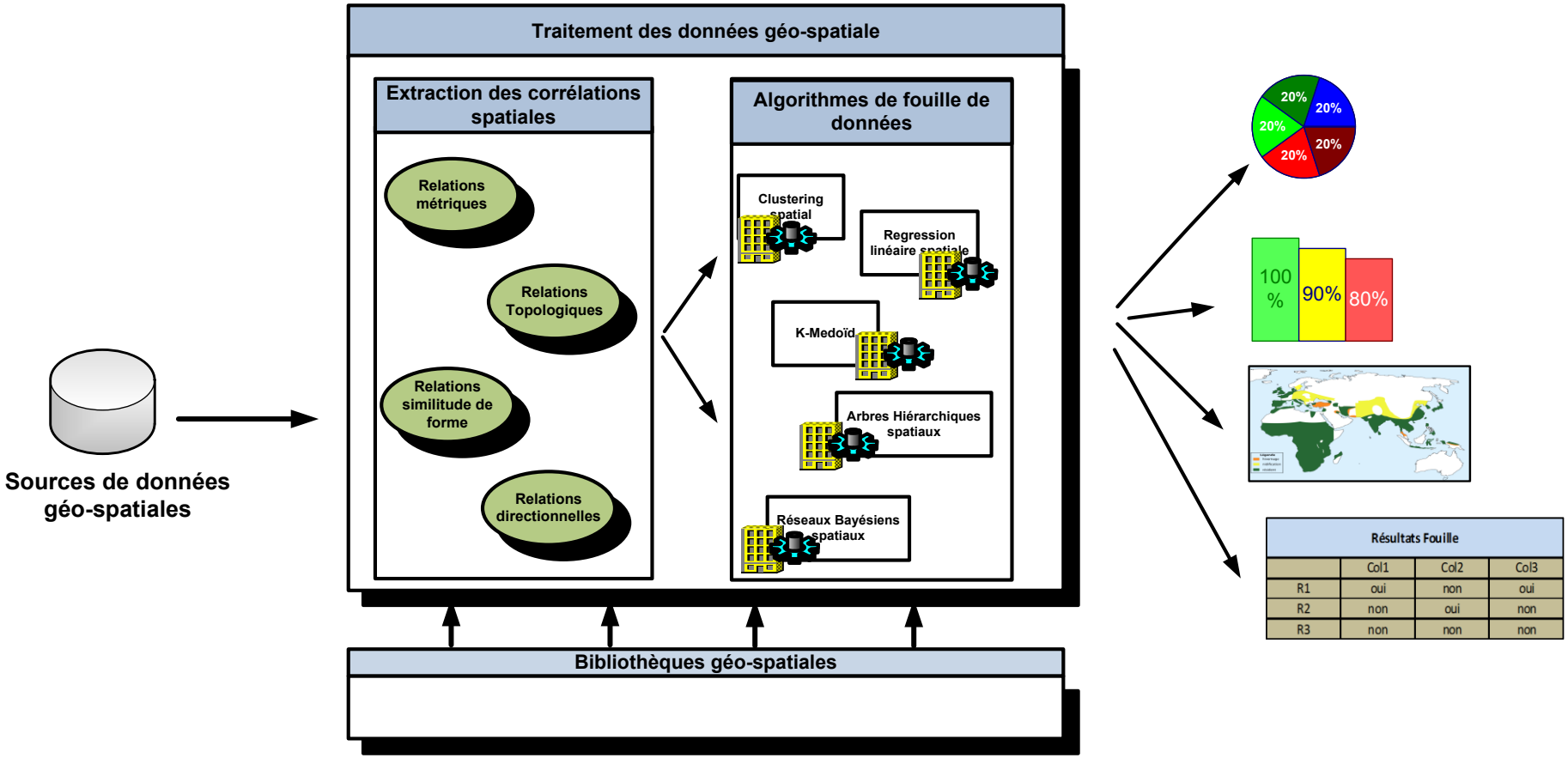


Figure 2.2-1: une approche à cheval entre le prétraitement et le développement d'algorithmes spatiaux

La figure ci-dessus résume le cœur de l'approche que nous proposons à savoir l'extraction des corrélations spatiales. Comme le montre la Figure 2.2-1, cette extraction, réalisée avec le concours des bibliothèques de traitement de données géo-spatiales, se fait au sein des différents algorithmes de fouille de données préalablement enrichis spatialement.

L'approche proposée est intégrée parce qu'en lieu et place de proposer une nouvelle approche qui part de rien, nous mettons à profit l'existant. Vu que la fouille de données « traditionnelle » a prouvé son efficacité et sa maturité, il est bon de tirer partie de ses avantages en effectuant un couplage fouille « traditionnelle », composante géo-spatiale.

Ce couplage fouille de données et composante géo-spatiale devrait se faire de façon transparente; i.e. offrir la possibilité de manipuler aisément la composante spatiale. Réaliser cette intégration de façon transparente passe par ériger la composante spatiale comme un type à part entière qui à l'instar des autres, disposera de ses propres opérateurs.

La complétude de cette approche réside principalement en deux grands points. Premièrement, assurer le traitement non seulement de tout type de géométries mais également de tout type de relations spatiales. Pour arriver à cette fin, l'approche se sert des bibliothèques de traitement des donnée géo-spatiales comme support. Ces bibliothèques offrent comme avantage, la possibilité de traiter tout type de géométries (point, ligne, polygone). Elles mettent également à disposition des fonctions capable d'exploiter les relations spatiales usuelles.

Deuxièmement, assurer la disponibilité de la composante géo-spatiale à toutes les étapes d'une fouille de données. Pour cela, il est important d'utiliser des outils de fouille de données qui offrent déjà des fonctionnalités couvrant l'ensemble des étapes de la fouille de données.

2.3. Relations spatiales et fouille de données

La spécificité de la fouille de données géo-spatiales comme notée dans les précédentes sections est principalement liée à la nature des entités géo-spatiales et plus spécifiquement au fait que ces entités entretiennent entre elles des relations. Toute démarche de fouille qui se veut applicable sur la donnée géo-spatiale devrait en premier lieu se focaliser sur ces relations – au risque d'être en porte-à-faux vis-à-vis de la première loi de la géographique ou loi de Tobler (cf. section 1.2.3.2).

Il existe différents types de relations dont les plus usuelles sont celles métriques et topologiques. Toutefois, dans un contexte de découverte de connaissances, il est important voir utile d'aller au-delà de ces relations usuelles pour prendre en considération d'autres

relations du genre celles directionnelles et de similarité de formes. La considération de ces différentes relations est au cœur de notre framework dédié à la fouille de données géo-spatiales.

Il est de plus important de noter que les relations spatiales restent incontournables quelque soit l'algorithme de fouille utilisé. Autrement dit, qu'il s'agisse d'effectuer une classification (clustering) de données géo-spatiales selon leurs caractéristiques communes ou d'effectuer de la prédiction d'une classe de valeur, on procédera toujours à la détermination (quantification ou qualification) de la/des relation(s) existante(s) entre les entités et y associer par la suite, au besoin, les préoccupations propres aux attributs descriptifs.

Cependant, ce qui change selon l'algorithme ou la classe d'algorithmes, est la sémantique rattachée à chaque relation ainsi que l'interaction possible avec les données descriptives. Ainsi, pour reprendre l'exemple précédent, dans une tâche de clustering, les relations géo-spatiales pourraient être vues comme des mesures de similarité (ou de dissimilarité), alors que pour effectuer de la prédiction basée sur les arbres de décision, ces relations peuvent être utilisées comme facteur discriminant et servir par la suite de nœud de décision.

Aussi, la plupart des relations, comme nous le verrons dans les sections à suivre, peuvent se décliner sous une forme quantitative et qualitative qui sera plus ou moins adaptée selon la classe d'algorithmes considérée. Ainsi de notre point de vue, la valeur qualitative d'une relation spatiale est beaucoup plus adaptée pour effectuer une prédiction basée sur les arbres de décision[11] tandis qu'une valeur quantitative sied mieux lorsqu'on effectue une tâche de clustering. Bien entendu, cela n'empêche pas d'utiliser des valeurs quantitatives pour construire un arbre de décision pour peu que l'algorithme offre une fonction de discrétisation à la volée de ce type de valeur (respectivement pour le clustering pour peu que l'utilise la mesure de similarité descriptive qui convient).

---

[11] La plupart des algorithmes de construction d'arbres de décision discrétisent préalablement les valeurs quantitatives afin de ramener à des valeurs qualitatives

2.3.1. Relations métriques entre entités géo-spatiales

Considérée comme mesure de similarité naturelle des entités géo-spatiales – dans un contexte de clustering – les relations métriques permettent généralement de calculer la distance « réelle » séparant deux entités géo-spatiales. A titre d'exemple, il pourra s'agir de la distance séparant un nid d'oiseaux d'un cours d'eau ou les distances séparant une maison située dans une zone d'habitation de l'établissement scolaire le plus proche.

Dans un contexte de fouille de données, la distance pourrait ne pas se résumer seulement à une expression quantitative. On pourra également s'intéresser à la distance d'un point de vue qualitatif *(Hernandez, et al., 1995) (Bogorny, et al., 2005)*.

2.3.1.1. Distance quantitative

Cette distance représente le plus court chemin entre deux (2) entités géo-spatiales (cf. Figure 2.3-1). Son calcul est effectué sous réserve des préoccupations évoquées dans la section précédente.

Il s'agit de la relation spatiale la plus utilisée dans le monde de la fouille de données géo-spatiales car considérée comme une des relations « naturelles » que les objets spatiaux entretiennent. À ce titre, elle s'est vue implémentée dans divers algorithmes de clustering sous forme de mesure de similarité.

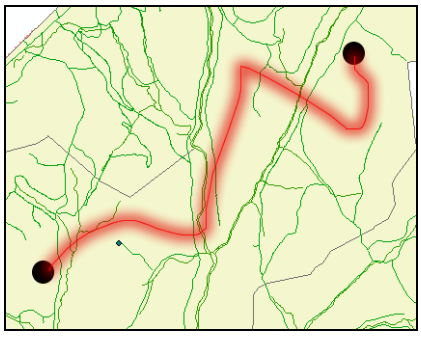


Figure 2.3-1 : Distance entre deux entités géo-spatiales de type ponctuel (en suivant la route)

2.3.1.2. Distance qualitative

Il s'agit de l'utilisation d'opérateurs qualitatifs pour l'évaluation de la proximité entre objets. L'idée sous tendue ici est que dans le langage naturel, la proximité est exprimée qualitativement plutôt que quantitativement. Partant de cela, on distingue différents opérateurs selon le niveau de découpage que l'on désire (très proche de, proche de, loin de, très loin de, …). Ces opérateurs utilisent également une distance (euclidienne ou toute autre distance) de façon sous-jacente.

Au regard de la nature qualitative de ces opérateurs, il est nécessaire au préalable de s'accorder sur le sens véhiculé par chacun d'eux. En d'autres termes, il s'agit de quantifier les différents qualificatifs. À quelle distance (en mètre ou kilomètre) peut-on juger qu'un objet est proche ou loin d'un autre ? Bien sûr, il s'agit d'une opération totalement subjective fonction du domaine d'étude, d'un problème particulier.

Contrairement à la valeur quantitative de la distance, utiliser la valeur qualitative dans une opération de clustering par exemple, nécessite l'adjonction d'une mesure de similarité pour attributs descriptifs.

2.3.2. Relations topologiques entre entités géo-spatiales

Au nombre des relations entretenues par les entités géo-spatiales, la relation topologique occupe une place de choix. Pour rappel, il s'agit d'une relation binaire entre deux(2) entités géo-spatiales lorsqu'on prend en considération leurs intérieurs, limites et extérieurs (cf. Figure 2.3-3). Ces relations entre intérieur, limite et extérieur ont été formalisées dans une matrice dénommée matrice à 9-intersections *(Egenhofer, et al., 1991)*.

Notons A^o, ∂A, A^- représentant dans cet ordre l'intérieure, la limite et l'extérieur d'une entité géo-spatiale A (respectivement B^o, ∂B, B^-). La matrice à 9-intersections est donnée par la figure ci-dessous:

	A^o	∂A	A^-
B^o	$A^o \cap B^o$	$\partial A \cap B^o$	$A^- \cap B^o$
∂B	$A^o \cap \partial B$	$\partial A \cap \partial B$	$A^- \cap \partial B$

Figure 2.3-2: Matrice à 9-intersection représentant la configuration entre deux objets quelconques

Les relations topologiques peuvent être catégorisées en cinq(5) grandes relations :

- Disjonction : elle traduit une non-existence de relation (contact) entre l'intérieur et la limite de deux objets.
- Adjacence : elle traduit une relation entre seulement les limites ; les intérieurs n'étant pas en contact.
- Intersection (limite et intérieur) : l'intersection limite renvoie à une relation entre intérieur d'un objet et limite d'un autre ; sans que les intérieurs des deux objets ne soient en contact. L'intersection intérieure renvoie à un contact entre l'intérieur d'un premier objet et l'intérieur et l'extérieur d'un second objet.
- Inclusion (limite et intérieur) : l'inclusion limite traduit la situation où un objet inclut un autre avec contact entre leurs limites respectives. L'inclusion totale, elle renvoie à une situation où un objet inclut totalement un autre (sans contact avec les limites).
- Egalité : il s'agit du cas où deux objets spatiaux sont exactement les mêmes (forme et localisation)

Tout comme au niveau des relations métriques, les relations topologiques sont également considérées sous deux points de vue : qualitatif et quantitatif.

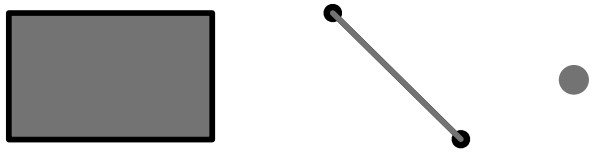


Figure 2.3-3: Intérieur (gris), Limite (noir) et Extérieur (en blanc) d'objets géométriques

2.3.2.1. Mesure topologique qualitative

La matrice des 9 intersections telle que décrite par *(Egenhofer, et al., 1991)*, traduit des relations topologiques qualitatives. En effet, grâce à cette matrice, implémentée d'ailleurs dans la plupart des bibliothèques de traitement de données géo-spatiales, on est à même de savoir si deux entités géo-spatiales sont adjacentes, disjointes, ou incluse l'une dans l'autre. Par exemple une maison est elle adjacente à une école, est ce que l'épicerie et le magasin de vente d'armes sont disjoints ? Est ce que deux routes se croisent ? On pourrait se poser autant de questions, mais les réponses possibles sont oui/non ou de manière plus formelle $\emptyset\ (vide)$ ou $1\emptyset(non\ vide)$.

Tout comme au niveau des relations métriques qualitatives, l'utilisation d'une mesure topologique qualitative, dans certaines tâches de fouille de données notamment le clustering, doit se faire en conjonction avec d'autres mesures de similarité en l'occurrence celles applicables aux variables de type binaire (cf. Annexe C). D'un point de vue pratique, il s'agit dans un premier temps d'évaluer la relation topologique entre les entités considérées et par la suite d'utiliser une mesure de similarité descriptive sur le résultat obtenu.

A titre d'exemple, supposons une opération de clustering spatial basée sur l'adjacence d'entités géo-spatiales « Maison ». Dans un premier temps, on évaluera l'adjacence des différentes maisons les unes par rapport aux autres. On obtient alors des données de type descriptif comme celles présentées à la Figure 2.3-4. Par la suite, on appliquera sur les données descriptives obtenues, une distance comme celle de Hamming, de Kendall ou de Cayley (cf. Annexe C).

Adjacence	Maison 1	Maison2	….	Maisonk
Magasin1	1∅	∅	1∅	1∅
Magasin2	1∅	1∅	∅	1∅
…	∅	1∅	∅	∅
Magasin k	1∅	1∅	1∅	1∅

Adjacence	Maison 1	Maison2	….	Maisonk
Magasin1	1	0	1	1
Magasin2	1	1	0	1
…	0	1	0	0
Magasin k	1	1	1	1

Figure 2.3-4: transformation de données topologiques qualitatives en données binaires en vue d'appliquer une distance de Hamming

La réponse qualitative, dépendamment du contexte, peut être suffisante ou non. Mais avoir un ordre de grandeur des relations topologiques est préférable dans un contexte de fouille de données dans la mesure où cela permet d'avoir plus de détails en ce qui concerne les connaissances découvertes.

2.3.2.2. Mesure topologique quantitative

La matrice à 9 intersections nous donne un renseignement de nature qualitative sur les relations topologiques entre entités géo-spatiales. Toutefois, il arrive des cas où on désire avoir plus de détails sur une relation topologique particulière ; c'est-à-dire disposer d'une valeur quantitative décrivant ladite relation.

Pour illustrer l'utilité de quantifier une relation topologique, prenons l'exemple de trois entités géo-spatiales : deux(2) entités linéaires et une surfacique (cf. Figure 2.3-5). Supposons que les deux objets linéaires soient partiellement inclus à différent degré dans le polygone. En se basant sur la topologie qualitative, on note simplement qu'effectivement les deux objets linéaires sont inclus dans le polygone – sans plus de précision. En procédant cependant à une évaluation quantitative de la relation topologique, on est à même de s'apercevoir du degré d'inclusion de chacune des formes linéaires.

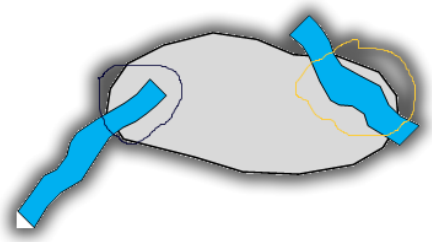


Figure 2.3-5 : configuration d'entités géo-spatiales avec différents degrés d'inclusion

Face à ce type de problématique, *(Egenhofer, et al., 1998) (Shariff, et al., 1998) (Nedas, et al., 2007)* proposent des indicateurs qui traduisent quantitativement les relations topologiques que les entités géo-spatiales entretiennent. En réalité, il s'agit d'une extension de la matrice à 9-intersection en vue de délivrer des mesures topologiques plus détaillées.

Toutefois, comme on le verra dans les sections à venir (cf. section 2.3.2.2.1-2), ces « détails » topologiques ne s'appliquent pas à tous les cas de configurations topologiques ni à tous les types d'entités géo-spatiales.

En effet, toutes les relations topologiques ne peuvent faire l'objet d'une quantification (l'égalité par exemple). Aussi, en ce qui concerne les entités géo-spatiales, les mesures de distance décrites dans cette section ne s'applique qu'à celles ayant une dimension supérieure ou égale à 1. Appliquées sur des objets ponctuels, certaines mesures sont impossibles à déterminer ; ou même aberrantes.

Les relations topologiques quantitatives sont résumées par deux grands concepts : le partitionnement (ou splitting) et la proximité (ou closeness).

2.3.2.2.1. La proximité ou closeness

Le concept de proximité ou closeness permet de décrire la distance séparant deux entités géo-spatiales selon deux principaux cas de figures. D'une part, il est utile pour décrire la distance séparant les limites d'une ligne et d'un polygone dans le cas où :

- Cas#1 (cf. Figure 2.3-6)
  - La limite de la ligne est située à l'extérieur du polygone (a)
  - La limite de la ligne est située à l'intérieur du polygone (b)

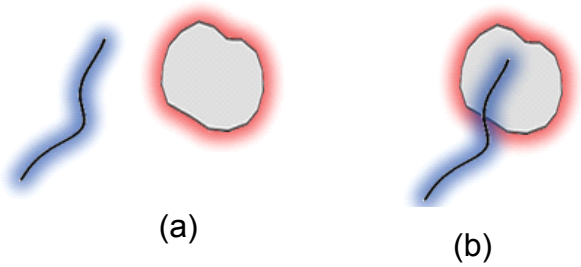


Figure 2.3-6: Configuration où les limites de la ligne sont à l'extérieur du polygone (a) et avec une limite à l'intérieur du polygone (b)

D'autre part, il permet d'évaluer la distance entre l'intérieur d'une ligne et la limite d'un polygone dans le cas où :

- Cas#2 (voir Figure 2.3-7)
  - L'intérieur de la ligne est localisé à l'extérieur du polygone (a)
  - L'intérieur de la ligne est localisé à l'intérieur du polygone (b)

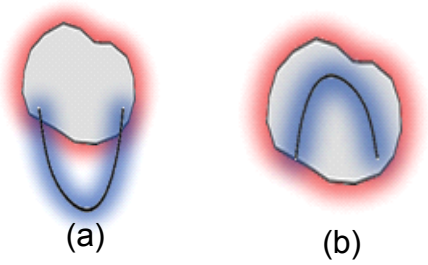


Figure 2.3-7: Configuration dans lesquelles l'intérieur de la ligne est localisé à l'extérieur du polygone (a) et à l'intérieur (b)

Plusieurs distances, en l'occurrence quatre(4), sont associées à la proximité et ces différentes distances correspondent aux différents sous cas de figure présentés ci-dessus. À chacune des mesures de distance, *(Egenhofer, et al., 1998) (Shariff, et al., 1998)* associent plusieurs ratios. Ces ratios ont essentiellement pour but de relativiser les distances vis-à-vis soit de la longueur de l'objet linéaire en considération, soit de l'entité surfacique ; cela afin de s'affranchir de l'échelle. Toutefois, ces ratios ne feront pas l'objet de détail dans le présent document dans la mesure où l'on estime suffisant les différentes distances (cf. Annexe E section E.2).

2.3.2.2.2. Le partitionnement ou splitting

Le splitting permet de décrire la nature du partitionnement produit entre une entité géo-spatiale linaire et une entité surfacique. Selon que l'on considère que c'est l'entité linéaire ou surfacique qui partitionne, on note deux(2) valeurs quantitatives: une première valeur qui correspond à une longueur et l'autre à une surface.

En effet, si l'on considère que l'objet surfacique partitionne l'objet linéaire, on s'intéressera à la longueur d'objet linéaire se trouvant dans l'objet surfacique ou la longueur du segment que l'entité linéaire partage avec l'objet surfacique. Ou plutôt, en considérant le partitionnement sous l'angle de l'entité linéaire qui divise l'objet surfacique, on s'intéressera à l'aire des différentes partitions de l'objet surfacique.

Que l'on s'intéresse à la longueur ou à la surface, il est important de faire une proportionnalité entre la valeur mesurée et la longueur de l'entité linéaire ou la surface de l'entité surfacique. Cela a pour avantage de relativiser les valeurs obtenues vis-à-vis de la

taille ou de la forme des objets considérés. En ce sens, l'utilisation des ratios est plutôt indispensable pour les mesures du splitting.

Dans le contexte d'une fouille de données basée sur du clustering, plus la valeur du ratio est élevé, plus cela signifie une forte similarité.

Il existe des configurations précises dans lesquelles il est inutile d'effectuer une mesure des différents ratios du splitting (cas de disjonction par exemple). Il faut en effet qu'il existe une interaction entre l'intérieur et la limite d'une entité géo-spatiale linéaire et l'intérieur, la limite et l'extérieur d'une entité géo-spatiale surfacique (cf. Figure 2.3-8).

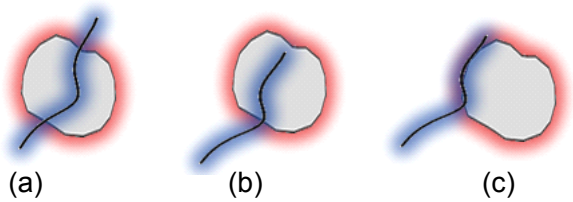


Figure 2.3-8: cas de configurations dans lesquelles on peut effectuer une mesure de splitting

On note plusieurs ratios (au nombre de 10). Mais dans la présente étude, nous jugeons que tous ces ratios ne sont pas pertinents d'autant plus que certains se recoupent. Seul ceux concernant la longueur seront l'objet d'une description. En effet, avec des ratios sur la longueur, on est à même d'évaluer le degré d'inclusion (Figure 2.3-8-a et Figure 2.3-8-b) ou d'adjacence (Figure 2.3-8-c) d'une ligne dans un polygone (cf. Annexe E section E.2).

2.3.3. Relations de similarité de formes

En plus de procéder à une fouille de données sur la base des relations usuelles dont on a discuté dans les sections plus haut – relations topologiques, métriques (cf. section 2.3.4) - que les données géo-spatiales entretiennent ; on peut effectuer une fouille de données sur la base d'une similarité de forme entre entités géo-spatiales. En effet, la mise à contribution de ce type de relation vise essentiellement à servir de complément ou plutôt à pallier à la « limitation » de la relation topologique de type égalité qui se contente de dire si oui ou non deux entités géo-spatiales sont égales mais ne donne pas une idée sur l'ordre de grandeur de cette relation. Les mesures de distance que nous décrirons dans cette section

sont non seulement adaptées pour l'évaluation de la similarité en termes de position et forme mais donnent des valeurs de mesure plus détaillées.

Les mesures de similarité de forme d'entités géo-spatiales relèvent de la reconnaissance de formes qui est utilisée dans des domaines tels le traitement et l'analyse d'images, celui de l'évaluation de la qualité géométrique et par extension le domaine de l'appariement de données géométriques.

Il existe diverses mesures permettant d'évaluer le degré de similarité entre deux formes géo-spatiales selon l'espace de représentation des éléments géographiques (espace cartésien, représentation par les signatures de polygones, représentation par les fonctions angulaires, etc.) *(Bel Hadj ali, 2001)*. À ces différents espaces de représentation, sont associées diverses mesures de distance. : Les distances de Hausdorff, de Fréchet, les distances entre fonctions angulaires, distances surfaciques, distance entre signatures de polygones, etc.

Dans les sections à venir, nous proposerons une description de certaines de ces distances à savoir la distance de Hausdorff, de Fréchet et la distance surfacique. L'objectif, dans le cadre de ce document, n'est pas d'être exhaustif sur les différentes mesures existantes. Ni de proposer systématiquement de nouvelles implémentations de ces algorithmes permettant le calcul de ces distances mais plutôt de mettre à profit des travaux réalisés dans le domaine *(Vauglin, 1997) (Bel Hadj ali, 2001)*.

Dans la plupart des travaux de recherche en matière de similarité de forme, en l'occurrence ceux de Bel Hadj Ali, deux formes sont jugées similaires si la valeur de la mesure atteint un certain seuil préalablement fixé. Dans notre contexte, celui de la fouille de données géo-spatiales, la considération d'un seuil n'est pas indispensable. En effet, plus la valeur de la mesure de similarité sera faible entre deux entités, plus elles s'influenceront moins lorsqu'on voudrait par exemple effectuer de la prédiction de valeur où un clustering (ces deux entités ne seront tout simplement pas dans le même cluster). On pourra toutefois se servir de la notion de seuil pour effectuer une fouille sur un sous ensemble particulier des données. Par exemple, effectuer de la prédiction de valeurs uniquement sur les entités géo-spatiales ayant un taux de similarité d'au moins 70% par rapport à une entité géo-spatiale de référence.

2.3.3.1. Distance de Hausdorff

La distance de Hausdorff est une mesure de dissimilarité largement utilisée dans le domaine de l'analyse et du traitement d'image. À titre d'exemple, on peut grâce à la distance de Hausdorff déduire sur une image, les formes qui se rapproche le plus d'une forme de référence donnée.

Tout comme la plupart des mesures de dissimilarité, la distance de Hausdorff quantifie le degré de dissemblance entre deux entités géo-spatiales en tenant compte de la position et de la forme.

De façon générale, la distance de Hausdorff se décrit comme étant le maximum de deux(2) quantités représentant le maximum des distances minimales entre deux entités géométriques. Si on note *A* et *B* deux (2) éléments de l'espace *P*, l'expression mathématique de la distance de Hausdorff est donnée par :

$$H(_{A,B}) = Max\,(C_{AB}, C_{BA}) \quad \text{(Eq 2.3-1)}$$

Avec $C_{AB} = \mathbf{Max}\left(\mathbf{Min}(d_{AB})\right)$ et $C_{BA} = \mathbf{Max}\left(\mathbf{Min}(d_{BA})\right)$

Les quantités *CA* et *CB* sont appelées composantes de la distance de Hausdorff et sont calculées en utilisant la distance euclidienne (on peut utiliser toute autre mesure de distance).

On note que plus deux entités géométriques sont de forme et de taille similaire, plus les deux composantes de la distance de Hausdorff sont équivalentes. Inversement, plus les deux entités sont dissemblantes, plus il existe un écart entre les deux composantes de cette distance. Aussi, plus la distance de Hausdorff est minime, plus les deux entités en considération sont similaires en termes de position et de forme.

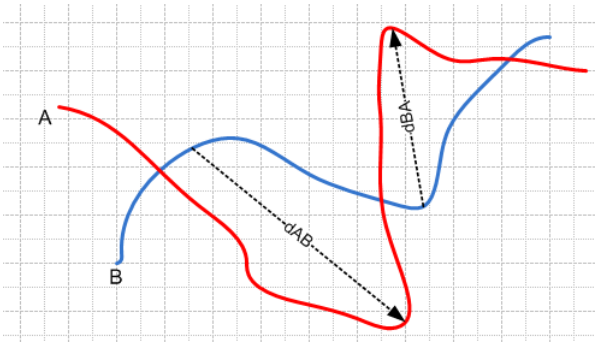


Figure 2.3-9: Distance de Hausdorff entre deux entités linéaires

La distance de Hausdorff est généralement utilisée pour quantifier la dissemblance entre des ensembles de points qui forment chacun les contours d'entités géo-spatiales. Cette distance s'applique de ce fait sur des entités géo-spatiales de dimension supérieure ou égale à 1. En effet, ramené à des entités géométriques de dimension 0, le calcul de la distance de Hausdorff, revient tout simplement à utiliser une distance euclidienne.

D'un point de vue pratique, *(Vauglin, 1997) (Devogele, 1997)* notent que la distance de Hausdorff sied mieux pour les entités géo-spatiales de type linéaire ; l'utilisation de cette distance restant toutefois complexe sur des entités surfaciques. Aussi, l'utilisation de cette distance sur les objets surfaciques considère ces derniers comme de simples contours. Bien entendu, cela n'est pas sans incidence sur la qualité de la mesure obtenue. En effet, si l'utilisation des contours d'un objet surfacique permet de rendre compte des écarts de la position, il n'en est pas de même pour l'écart de forme *(Bel Hadj Ali, 2001)*. D'où la nécessité d'utiliser d'autres types de mesures en ce qui concerne la caractérisation de l'écart de forme entre deux entités surfaciques.

2.3.3.2. Distance de Fréchet

Tout comme la distance de Hausdorff, la distance de Fréchet est une mesure de dissimilarité permettant de quantifier le degré de non ressemblance entre deux(2) formes géométriques. A la différence de la première (distance de Hausdorff), la distance de Fréchet permet de mieux capturer cette dissimilarité *(Eiter, et al., 1994) (Devogele, 1997) (Bel Hadj Ali, 2001) (Buchin, et al., 2006)* parce qu'elle tient compte de la position des points composant un contour ainsi que de leur ordre. De ce fait, on dit de cette mesure qu'elle permet de quantifier la dissimilarité entre deux contours orientés.

De façon intuitive, la distance de Fréchet est définie comme étant le minimum de chaîne – en termes de longueur - nécessaire pour qu'un homme puisse tenir en laisse son chien; les deux marchant respectivement le long d'une ligne imaginaire, pouvant varier leur vitesse mais n'ayant pas la possibilité de faire marche-arrière.

D'un point de vue formel, la distance de Fréchet se définit comme suit : tout contour peut être considéré comme une fonction continue de l'espace des réels ($\mathbb{R}$) vers un espace

vectoriel V muni d'une métrique d[12]. Soient $\mathcal{C}_1$ et $\mathcal{C}_2$ deux(2) courbes quelconques. A ces deux courbes, on associe deux(2) fonctions continues f et g (respectivement) telle que

$$f:[a,b] \rightarrow V et\ g:[a',b'] \rightarrow V' avec\ a \leq b\ (resp\ a' \leq b')$$

La distance de Fréchet entre ces deux courbes est définie par la quantité suivante :

$$d_1F\ (f,g) = [Inf]_1(\alpha,\beta\ t \epsilon [0,1])\ Max\ (d(f(\alpha(t)),g(\beta(t))))$$

Où α (respectivement β) est une fonction continue croissante de [0,1] vers [a, b] (respectivement [a', b']).

Quoique donnant de meilleurs résultats – comparée à la distance de Hausdorff – l'inconvénient majeur de la distance de Fréchet demeure sa complexité qui est de l'ordre de $O(pq \log_2(pq))$ *(Eiter, et al., 1994)* (où p et q représentent le nombre de segments de deux courbes quelconques f et g). Face à cette complexité, *(Eiter, et al., 1994)* propose une version discrète de la distance de Fréchet dénommée *distance discrète de Fréchet*. Cette version, en plus d'être simple dans sa mise en œuvre, permet de réduire la complexité qui dévient de l'ordre de $O(pq)$ (Une version discrète de la distance de Fréchet a également été proposée par *(Mosig, et al., 2005)* mais avec une complexité $O(p^2q^2)$ toutefois supérieure à celle de *(Eiter, et al., 1994)*).

Par analogie avec l'homme et son chien marchant le long de leur courbes respectives, l'idée sous-tendue par la version de *(Eiter, et al., 1994)* est de supposer que l'homme autant que le chien ne peuvent s'arrêter qu'au niveau des sommets de leurs courbes respectives et qu' à cet instant précis, on détermine la distance de Fréchet entre les sommets de courbes considérées.

(Eiter, et al., 1994) propose un algorithme récursif permettant le calcul de la variante discrète de la distance de Fréchet ; dont la substance est la suivante : soient P et Q deux(2) courbes ayant respectivement n et m sommets respectivement ((p_1, p_2, …p_i,… p_n), (q_1, q_2, …q_i,… q_m)). On a :

[12] Cette métrique peut être une distance euclidienne ou toute autre distance

$$\delta_F(P_n, Q_m) = Max\left(\begin{matrix} d(p_n, q_m) \\ Min\left(\begin{matrix} \delta_F(p_{n-1}, q_m)\, si\ n \neq 1 \\ \delta_F(p_n, q_{m-1}) si\ m \neq 1 \\ \delta_F(p_{n-1}, q_{m-1}) si\ n\ et\ m \neq 1 \end{matrix}\right) \end{matrix}\right)$$

Où $d(p_n,q_m)$ représente la distance euclidienne entre les deux points p_n,q_m

Il faut noter cependant que cette distance (distance discrète de Fréchet) n'est qu'une approximation de la distance exacte de Fréchet en ce sens qu'elle n'en donne qu'un ordre de grandeur. On note la relation suivante : Si d_f représente la distance de Fréchet et δ_F la distance discrète de Fréchet, on a : $d_F \leq \delta_F$.

Tout comme nous l'avons mentionné pour la distance de Hausdorff, la distance de Fréchet possède une limitation majeure ; à savoir la restriction à des objets géométriques de type linéaire. D'où la nécessité de considérer tout autre mesure de distance pour la reconnaissance de forme entre deux entités géographiques de type surfacique.

2.3.3.3. Distance surfacique

Suite à la limitation des mesures de distances précédentes quant à la prise en compte d'entités géo-spatiales de type surfacique, *(Vauglin, 1997)* introduit une mesure de distance dénommée distance surfacique permettant de remédier à cette limitation. En effet, la distance surfacique consiste en une évaluation de la proportion de surface commune entre deux(2) entités géo-spatiales *(Bel Hadj Ali, 2001) (Devogele, 1997)*.

Soient deux entités surfaciques A et B ayant respectivement pour aire S(A) et S(B). La distance surfacique entre ces deux entités est donnée par:

$$d_s = 1 - \frac{S(A \cap B)}{S(A \cup B)}$$

Où $S(A \cap B)\ et\ S(A \cup B)$ représentent respectivement la surface de l'intersection et de l'union des polygones A et B.

Dans un contexte où il s'agit de comparer la forme d'une entité géographique surfacique vis-à-vis de plusieurs (correspondance 1:m), la formule de calcul de la distance surfacique est:

$$d_s = 1 - \frac{\Sigma_{i=1}^{n} S(A_i \cap B)}{\Sigma_{i=1}^{n} S(A_i \cup B)}$$

Qu'il s'agisse d'un cas de correspondance 1 vers 1(1:1) ou 1 vers plusieurs (1:m), la distance surfacique est une valeur qui oscille entre l'intervalle [0,1]. Lorsque cette valeur est égale à 1, cela signifie que les deux entités géo-spatiales sont disjointes tandis qu'une valeur de 0 signifie que les deux(2) entités sont égales.

2.3.4. Relations directionnelles entre entités géo-spatiales

Les relations directionnelles que les entités géo-spatiales entretiennent entre elles peuvent être amenées à occuper une place importante dans un contexte de fouille de données spatiales. En effet, tout comme les relations topologiques, les relations directionnelles peuvent être utilisées comme relation à part entière ou servir de complément aux relations topologiques.

A ce titre, prenons l'exemple d'objets géo-spatiaux « zones de tension » mutuellement disjoints au sein desquels on s'intéresse à la progression d'un phénomène quelconque afin de pouvoir regrouper (clustering) ces entités géo-spatiales selon leur caractéristiques communes. En s'intéressant exclusivement aux relations topologiques, on notera qu'elles sont similaires. De ce fait, on risque de retrouver ces objets au sein d'un seul cluster Par contre, en associant les relations directionnelles à la relation topologique de type disjonction, le résultat sera certainement différent : on caractérise mieux le type de rapport que ces entités entretiennent.

Les résultats d'une opération pour caractériser les positions directionnelles d'entités vis-à-vis les unes des autres sont pour la plupart qualitatifs. On obtient en effet des valeurs du type Nord, Sud, Sud-Est, etc. Pareil type de résultat peut directement être exploité par certaines classes d'algorithmes de fouille. Toutefois pour certains types d'algorithmes, en l'occurrence les algorithmes de clustering ou de plus proche voisins(KNN), il est important de disposer de mesures de similarité basées sur ces relations.

Pour se rendre compte de cette problématique, partons de l'exemple suivant. Supposons trois(3) entités géo-spatiales E1, E2, E3 ayant respectivement pour position cardinale par rapport à une entité référence Ouest, Sud-est, Nord-Ouest. Si on désire regrouper ces entités, il est difficile de dire laquelle de E1 ou E3 est plus proches de E2. En

effet, sur quelle base pourra-t-on affirmer que le Nord est plus proche de Sud ou le Sud du Nord-Est?

Cette illustration dénote de la nécessité, pour certaines classes d'algorithmes de fouille de données, de disposer d'une mesure de dissimilarité basée sur ces relations. Toutefois avant de parler de mesure de dissimilarité, il est important de disposer en premier lieu d'outils permettant de caractériser les relations directionnelles entre deux entités. Les relations directionnelles ne sont pas véritablement implémentées dans les API géo-spatiales (ex. JTS, GeoTools, GeOxygene, etc.) contrairement à d'autres relations telles celles topologiques. Pour cela, il est important de disposer d'un cadre (framework) directionnel qui puisse être adapté pour effectuer une fouille de données géo-spatiales qui prend en compte tout type d'entités géo-spatiales.

2.3.4.1. Le framework directionnel de Goyal

Nombre de recherches en ce qui concerne la mise en œuvre de modèles pour permettre de caractériser les relations directionnelles entre entités géo-spatiales, ont été mis en œuvre. On peut citer le modèle:

- basé sur les projections planes ;
- basé sur les projections coniques ;
- basé sur les tableaux symboliques.

Ces modèles ont toutefois montré leurs limites quant à la caractérisation de la relation selon leur *(Goyal, 2000)*:

- Dimension,
- taille,
- complexité des entités géo-spatiales considérées.

Fort de cela, *(Goyal, 2000) (Goyal, et al., 2001)* proposent un cadre (framework) qui permet de capturer les relations directionnelles entre objets géo-spatiaux - indépendamment des facteurs limitatifs que nous avons cités plus haut.

Avant d'en venir aux différents éléments constitutifs de ce framework, il est important de noter comment est ce que *(Goyal, 2000)* décrit le modèle permettant de caractériser les relations directionnelles entre entités géo-spatiales.

Dans un contexte de caractérisation de relations spatiales, en l'occurrence celles directionnelles, il est important d'avoir un framework robuste ; robuste dans le sens de la possible caractérisation de relation directionnelle entre entités de divers types (point, ligne, polygone) ; mais également capable de tenir compte de la complexité des entités en considération.

Au regard de la limitation des modèles directionnels existants, Goyal propose un framework qui étend le model basé sur les projections (cf. *(Frank, 1996)* pour plus de détails); qui à l'origine se limite seulement aux objets ponctuels. La robustesse que nous avons évoquée pour ce framework est également assurée par les trois(3) matrices directionnelles qui le composent. Ces matrices directionnelles - brute, détaillée et étendue - sur lesquelles nous reviendrons dans les paragraphes qui suivent (cf. sections 2.3.4.2;2.3.4.3 et 2.3.4.4), sont à même de fournir divers types de résultats (quantitatif, qualitatif) selon le niveau de détail attendu.

Qu'il s'agisse de la matrice « brute », « détaillée », « étendue », le modèle proposé par Goyal, tout comme celui basé sur les projections, consiste en un partitionnement de l'espace entourant l'objet source en neuf(9) parties dénommées « quadrants directionnels[13] » et correspondant pour chacune d'elle à un point cardinal (Nord, Sud, Est, Ouest, Nord-est, Nord-Ouest, Sud-est, Sud-ouest) plus un point représentant un zéro directionnel, c'est-à-dire une configuration dans laquelle deux(2) objets se confondent - d'un point de vue directionnel (cf. Figure 2.3-10).

[13] Traduction libre

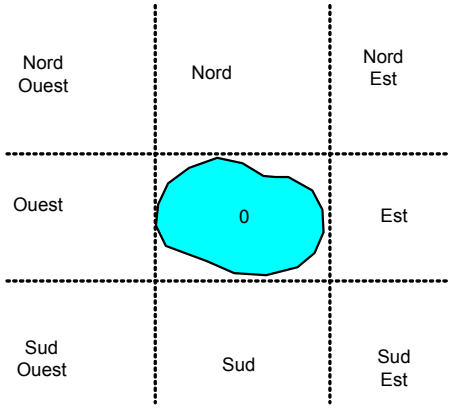


Figure 2.3-10: Subdivision de l'espace en « quadrants directionnels » – adapté de *(Goyal, 2000)*

2.3.4.2. La matrice directionnelle-cardinale « brute »

La matrice directionnelle « brute » ou « coarse cardinal-direction matrix » est une matrice 3x3 qui permet de caractériser les relations directionnelles entre deux entités géo-spatiales surfaciques (polygones). Cette matrice fournit des résultats de nature booléenne ; ce qui peut être convenable selon la situation. Ainsi, dans une configuration où un objet cible est situé dans un ou plusieurs « quadrants directionnels » (cf. Figure 2.3-11) d'un objet de référence, on enregistre la valeur 1 pour les cases correspondantes dans cette matrice et 0 pour toutes les autres cases.

L'inconvénient majeur de cette matrice réside dans le fait qu'elle s'intéresse exclusivement aux objets surfaciques.

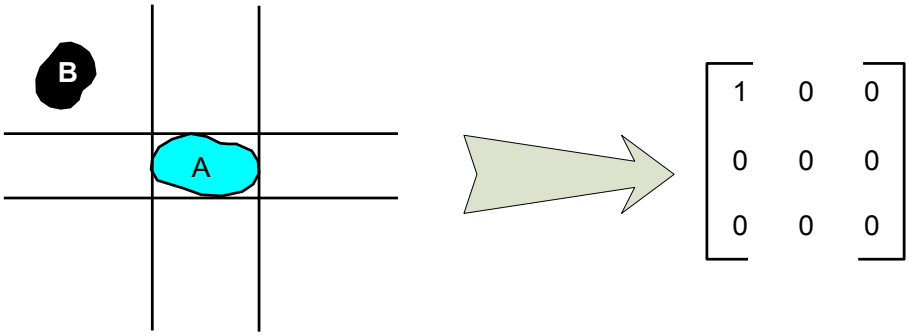


Figure 2.3-11: configuration et matrice dans le cas où un objet cible est situé au Nord-Ouest d'un objet de référence (adapté de *(Goyal, 2000)*)

Il faut noter par ailleurs que la matrice est sujette à une contrainte majeure notamment la « quadri-contigüité » (4-Neighbors) des valeurs non nulles[14] et à des configurations non réalistes. Par exemple, une matrice n'ayant que des valeurs nulles ne peut pas exister car signifiant l'absence d'un objet cible.

2.3.4.3. La matrice directionnelle-cardinale « détaillée »

Tout comme la matrice directionnelle-cardinale « brute », celle « détaillée » ne s'intéresse qu'aux objets surfaciques parce que ne pouvant pas s'appliquer aux objets ponctuels et linéaires comme nous le verrons par la suite. Toutefois à la différence de la matrice « brute », elle va au-delà d'une simple caractérisation de la présence d'un objet cible dans un « quadrant directionnel » en donnant le degré de présence dans ledit ou lesdits « quadrants ». Le modèle détaillé trouve son utilité dans les cas de configurations où un objet cible s'étend sur plus d'un « quadrant directionnel ». En effet, le modèle considère la répartition de la surface de l'objet cible entre les différents « quadrants directionnels » (cf. Figure 2.3-12).

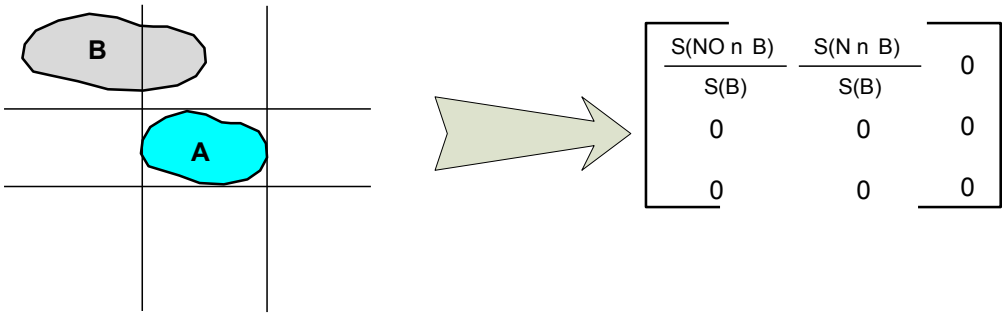


Figure 2.3-12: Matrice détaillée dans la configuration où un objet s'étend sur deux (2) « quadrants directionnels » (adapté de *(Goyal, 2000)*)

Comme on le note sur la figure ci-dessus, chaque élément de la matrice détaillée est obtenu en faisant un rapport entre la surface de l'intersection entre la cible et un « quadrant directionnel » ou plus exactement, la somme des intersections et la surface de la cible (cf. Figure 2.3-13). De ce fait, chaque élément à une valeur comprise entre l'intervalle fermé [0,1].

[14] Cela signifie que pour une matrice contenant plus d'une valeur non nulle, ces valeurs doivent être horizontalement ou verticalement contigües

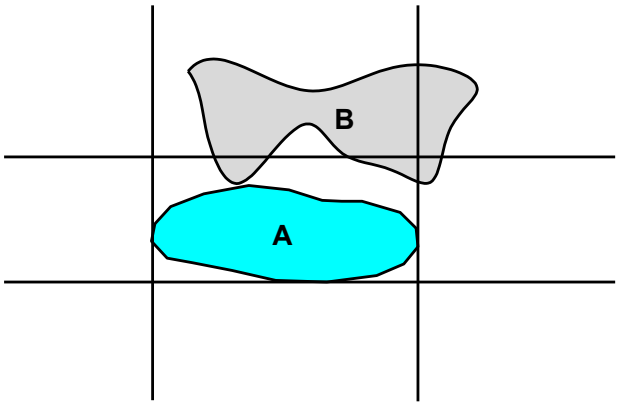


Figure 2.3-13: configuration dans le cas d'une cible qui intersecte plusieurs fois un même « quadrant directionnelle »

2.3.4.4. La matrice directionnelle-cardinale « étendue »

Il s'agit d'une extension de la matrice « brute » en vue de prendre en compte n'importe quel type d'objet géo-spatial indépendamment de sa dimension (point, ligne, polygone). Pour atteindre ce but, chacun des éléments d'une matrice étendue est codée sur neuf(9) bits désigné par le nom « Neighbors code ». Chaque bit contient une valeur (0 ou 1) représentant l'intersection entre les « quadrants directionnels ». Ainsi, les bits au sein d'un « Neighbor code » sont organisés comme suit (cf. Figure 2.3-14):

- X_0 : représente l'intersection avec le « quadrant directionnel »
- X1 a X9 : représente l'intersection avec les « quadrants directionnels » voisins (Nord, Est, Sud, Sud-est,…).

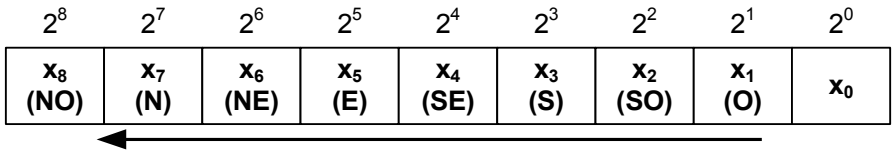


Figure 2.3-14: Structure d'un Neighbor code

La valeur que prend chaque bit du « Neighbor code » est soumise à certaines règles.

- Ainsi, X_0 prend la valeur 1 si l'entité géographique cible se retrouve exclusivement dans un seul « quadrant directionnel » i.e. sans intersection avec ces voisins (0 dans le cas contraire).

- Si X_0 est à 1, tous les autres bits sont automatiquement fixés à 0. Au cas où, X_0 a la valeur 0 (preuve que l'entité cible a des relations avec le voisinage), alors pour chaque intersection de la cible avec un « quadrant directionnel » donné, le bit correspondant prend la valeur 1 (sinon 0).

La valeur totale du « Neighbor code » est une somme pondérée – selon les puissances de 2 - des différents bits le composant au regard de leur position.

2.3.4.5. Matrices de direction et Mesure de dissimilarité

Les modèles directionnels définis au sein du framework proposé par Goyal, permettent chacun de produire une matrice retraçant la position cardinale d'une entité géographique cible vis-à-vis d'une autre. Toutefois, il est important en vue de pouvoir effectuer certaines tâches de fouille de données, disposer d'un moyen de comparer des matrices directionnelles. C'est dans ce sens que *(Goyal, 2000)* décrit une mesure de dissimilarité (une distance) permettant de quantifier le degré de dissemblance entre deux matrices directionnelles.

La mesure de dissimilarité décrite par Goyal, se fonde sur un graphe de voisinage conceptuel. Ce graphe conceptuel, basé sur les neuf(9) « quadrants directionnels » (cf. Figure 2.3-10), permet d'une part de décrire la distance entre deux(2) « quadrants directionnels » qui soit dit en passant est d'une unité entre deux « quadrants » adjacents ; d'autre part, définit le chemin à parcourir en partant d'un « quadrant » vers un autre (voir Figure 2.3-15).

Ainsi, pour une entité cible quittant un « quadrant » vers un autre, le chemin suit l'adjacence des « quadrants » (verticalement ou horizontalement) en privilégiant celui qui minimise la distance totale. À titre d'exemple, un objet quittant le Nord-Est pour le Sud-Ouest passe soit par le Nord soit par l'Est.

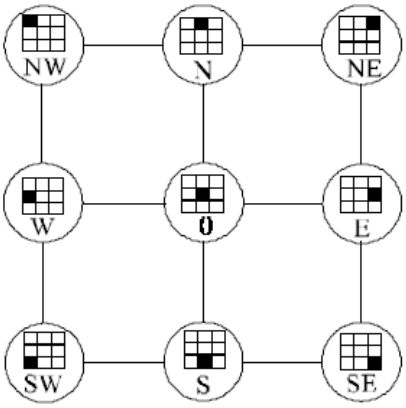


Figure 2.3-15: Graphe de voisinage conceptuel (tiré de *(Goyal, 2000)*)

Dans ce graphe de voisinage, la distance maximale possible est de 4 unités (cas d'un objet se situant au Nord-Ouest et migrant vers le Sud-Est) tandis que celle minimale est de 0 (distance séparant un objet de lui-même).

À partir du graphe de voisinage conceptuel, on construit une matrice de dissimilarité[15] qui décrit la distance séparant les différents points cardinaux les uns des autres.

	N	NE	E	SE	S	SO	O	NO	0
N	0	1	2	3	2	3	2	1	1
NE		0	1	2	3	4	3	2	2
E			0	1	2	3	2	3	1
SE				0	1	2	3	4	2
S					0	1	2	3	1
SO						0	1	2	2
O							0	1	1
NO								0	2
0									0

Figure 2.3-16: Matrice de distances entre points cardinaux ou matrice de dissimilarité directionnelle

Comme le note *(Goyal, 2000)*, le degré de différence entre deux(2) matrices de direction est obtenu en évaluant le coût de transformation d'une matrice source vers une matrice destination. Pour être plus exact, il s'agit du coût de la migration des éléments non

[15] Il s'agit dans notre cas d'une matrice de dissimilarité symétrique. Ce qui explique pourquoi seule la partie supérieure de la matrice comporte des éléments

nuls de la matrice source vers ceux de la matrice cible. Le coût de transformation est obtenu en tenant compte de deux(2) paramètres :

- la distance séparant les points cardinaux source et destination (obtenu par une lecture de la matrice de dissimilarité) ;
- la valeur de l'élément.

Pour illustrer le mode de calcul, supposons deux matrices de direction données ci-après:

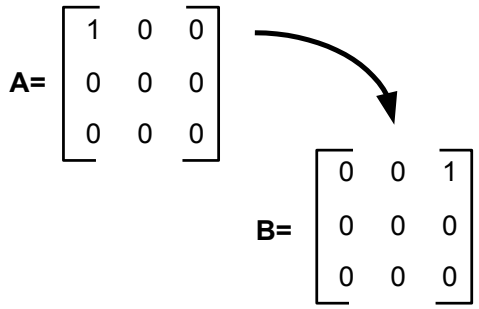


Figure 2.3-17: Calcul du coût de transformation entre deux matrices directionnelles

Ces matrices traduisent une situation où on désire estimer la distance séparant un objet cible situé au Nord-Ouest d'une référence et un objet situé au Nord-Est. Le coût de la transformation est donné par : *C=1 x (Distance (NO, NE)=1 X (2)=2*.

Pour le cas de figure présenté ci-dessus, le coût est assez simple à obtenir parce que nous sommes dans une situation où l'entité géographique cible est entièrement située dans un seul « quadrant » d'où la valeur 1 dans la case correspondante dans la matrice. Le coût de transformation dans un tel cas, revient tout simplement à lire la matrice de dissimilarité (cf. Figure 2.3-16).

Dans le cas d'une entité géo-spatiale cible s'étendant sur plus d'un « quadrant directionnel» (cf. Figure 2.3-18), le coefficient qui pondère la distance entre les points cardinaux est la proportion de surface de l'entité se trouvant dans le « quadrant » considéré. Ainsi le Coût *C=0.25 X Distance (NO, NE) + 0.75 X Distance (N, NE)=0.25 X (2) + 0.75 X (1)=1.25*

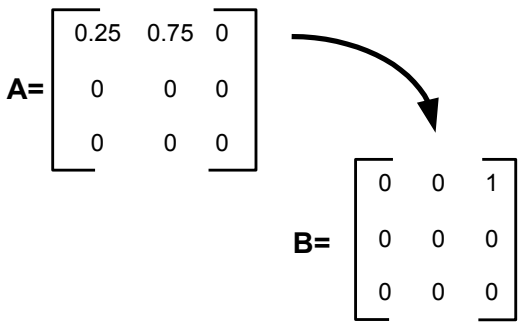


Figure 2.3-18: coût de transformation dans le cas d'une cible se trouvant répartie sur deux « quadrants » directionnels.

En réalité les cas de figures présentés précédemment (cf. Figure 2.3-17 et Figure 2.3-18) sont plutôt simplistes parce que soit la matrice de départ ou de destination représente une situation où l'entité géographique cible se retrouve dans un seul « quadrant directionnel ».

On pourrait se retrouver dans une situation bien plus complexe où les deux matrices représentent des situations où l'objet cible s'étend sur plus d'un « quadrant ». Dans pareil situation et à des fins de généralisation, *(Goyal, 2000)* ramène le calcul du coût de transformation, en un problème d'optimisation ; problème dans lequel il s'agit de minimiser une fonction-objectif - le coût de transformation – sous certaines contraintes. Pour une résolution efficiente de ce problème d'optimisation, *(Goyal, 2000)* propose un ensemble de concepts dont la « commonality-matrix », la « direction-difference matrix ».

2.4. Un cadriciel pour l'intégration de la composante spatiale au sein d'un outil de fouille de données

La complexité des données géo-spatiales et la nature des relations qu'elles entretiennent est telle qu'il est difficile, voir impossible d'utiliser « aveuglement » les algorithmes de fouille« traditionnelle » de données. Dans la section précédente (cf. section 2.3), nous avons décrit quelques relations spatiales qui devraient être au centre de toute démarche de fouille de données géo-spatiales et plus spécifiquement pour une démarche dont la finalité est l'intégration de la composante géo-spatiale dans un outil de fouille « traditionnelle ».

La problématique de prise en compte des relations géo-spatiales est certes une condition sine qua non mais demeure juste une partie du problème consistant à intégrer la composante géo-spatiale dans un outil de fouille. En effet, quand vient le moment de l'intégration, plusieurs difficultés émergent, au nombre desquelles :

- la diversité des algorithmes de fouille : on note plusieurs classes d'algorithmes de fouille de données. Chacune des classes d'algorithmes vise des objectifs différents et a un mode de fonctionnement différent. On note par exemple que les algorithmes effectuant de la prédiction/estimation (arbre de décision, réseaux de Bayes) fonctionnent différemment de ceux effectuant du clustering. En effet tandis que les premiers se basent sur le degré d'impureté[16] de chaque attribut (nombre de valeurs différentes pour chaque attribut), les seconds se fondent sur des mesures de similarité *(Tekmono, 2006) (Teknomo, 2009)*. Aussi tandis que les algorithmes de clustering se basent sur des valeurs numériques pour fonctionner efficacement ; ceux de prédiction peuvent se contenter de valeurs qualitatives des mesures obtenues sur la base des relations spatiales.
- la problématique de prise en compte des données descriptives dans la fouille de données : la donnée géo-spatiale est formée d'une composante géométrique et d'une autre descriptive qu'il est important de prendre en compte lorsqu'on effectue de la fouille. Non seulement, il faut prendre en compte l'information descriptive associée à une donnée géométrique, mais également prendre soin d'attribuer un poids à chaque composante s'il y a lieu, i.e. décider quelle composante on désire le plus mettre en valeur ; parce que selon le contexte, la composante descriptive peut être porteuse de plus de connaissances que celle géométrique (réciproquement). à titre d'exemple, prenons des zones de criminalités localisées spatialement pour lesquelles on désire faire un regroupement selon qu'elles partagent des caractéristiques communes.il est bon de tenir compte du nombre d'incidents criminels, du revenu moyen, du taux de chômage relevés dans chacune de ces zones. Par

[16] Voir mesure d'entropie, index de Gini, erreur de classification (Teknomo, 2009)

la suite indiquer à l'algorithme de fouille que l'on désire accorder plus de poids au nombre d'incidents criminels plutôt qu'à l'influence – éventuelle – que ces zones pourraient avoir les unes sur les autres.

Fort de cela, nous proposons un cadriciel (cf. Figure 2.4-1) dont l'objectif principal est de compléter l'approche que nous avons proposée (cf. section 2.2) en schématisant les grandes étapes à franchir pour implémenter pratiquement l'approche.

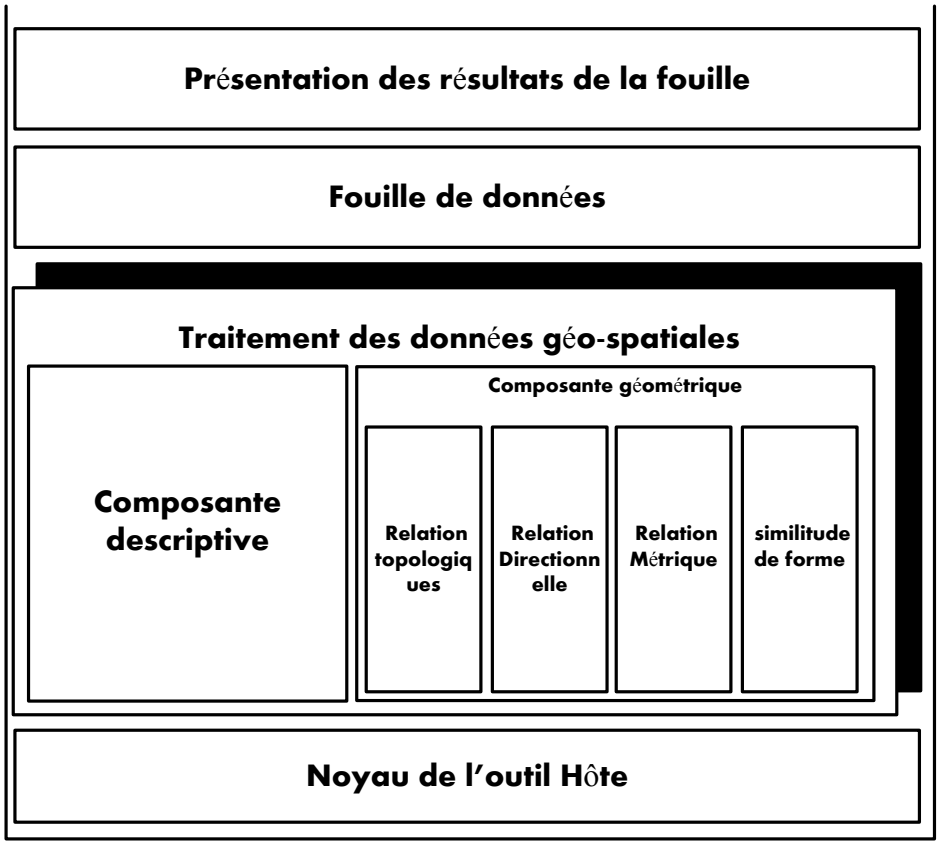


Figure 2.4-1 : Framework pour l'intégration du spatial dans un outil de fouille de données

Pour implémenter efficacement une approche intégrée de fouille de données qui se veut complète, cohérente et transparente, il faut passer principalement par quatre(4) grandes étapes :

- Sélection de l'outil hôte ;
- Traitement de la donnée géo-spatiale ;

- Fouille de données proprement dite ;
- Présentation/visualisation des résultats.

2.4.1. La couche noyau de l'outil hôte

L'objectif de la présente recherche est de proposer une nouvelle approche de fouille de données qui tient compte de la composante spatiale à toutes les étapes d'une fouille de données. En plus de l'approche qui devrait être la résultante des travaux de recherche, un prototype implémentant l'approche devrait être mis en œuvre. Pour ce faire, il est nécessaire de partir d'un outil de fouille de données existant qui de préférence offre des fonctionnalités pour le support des différentes étapes du processus de fouille de données . D'où la nécessité d'avoir une couche dénommée « Noyau de l'outil hôte » qui constitue le socle de ce cadre.

Pour rappel, l'objectif de l'utilisation d'un outil existant de fouille de données et par extension l'utilité de la couche basse, s'inscrit dans un objectif de réutilisation du potentiel existant des outils de fouille de données « traditionnelles ». En lieu et place donc de (re) développer des outils de fouille, il est préférable de tirer parti de la puissance qu'offrent les outils existants.

Plusieurs logiciels ou bibliothèques peuvent servir d'outils de base au sein duquel la composante géo-spatiale sera intégrée. Mais le choix de tels outils devra se faire sous le respect de certains critères dont (cf. Annexe A):

- l'extensibilité : il faudra que l'outil choisi se prête à une extension, un enrichissement des algorithmes disponibles avec le type géo-spatial.
- la licence d'utilisation : ce deuxième critère va de pair avec le premier dans la mesure où l'extensibilité pourrait être fonction de la licence d'utilisation. En effet, si on ne dispose pas d'une licence open-source, il sera difficile d'accéder au code source et par conséquent procéder à un enrichissement de l'outil. A défaut d'avoir une licence open-source, il faudra vérifier si l'outil dispose de documentation décrivant les procédés pour enrichir ledit outil.
- les fonctionnalités offertes : plus l'outil offre diverses fonctionnalités, plus cela est souhaitable. Parlant de fonctionnalités, il s'agit principalement de la diversité des algorithmes de fouille (clustering, association, réseaux de

neurones, prédiction/régression, etc.) ; mais également des fonctionnalités situées en amont et en aval de la fouille elle-même (préparation des données, réduction de dimensionnalité, visualisation des résultats de la fouille, export des résultats de la fouille, etc.).

- la robustesse : l'outil choisi devra être robuste en termes de support de gros volume de données

Au nombre des bibliothèques pouvant servir de base pour une intégration du spatial, on note :

- KNIME[17] ;
- ADAM [18] ;
- WEKA[19] ;
- ALPHAMINER[20]
- ORANGE[21] ;
- TANAGRA [22];
- …

Dans la suite des travaux, nous avons opté pour l'outil KNIME au regard d'un certains nombre d'avantages liés à cet outils (extensibilité, richesse dans les fonctionnalités (cf. Annexe A section A.3).

2.4.2. La couche traitement des données géo-spatiales

Comme on l'a mentionné maintes fois, cette couche constitue le cœur de ce cadriciel. Il s'agit de la couche qui permet de traiter la composante géo-spatiale dans toute sa spécificité. L'objectif principal pour cette couche est de traiter l'information géo-spatiale et fournir le résultat sous une forme adaptée à l'algorithme de fouille de données. Pour le traitement des corrélations spatiales, cette couche met à profit les fonctionnalités offertes

[17] http://www.knime.org/
[18] http://datamining.itsc.uah.edu/adam/index.html
[19] http://weka.sourceforge.net/wekadoc/index.php/en:Weka_3.4.13
[20] http://www.eti.hku.hk/alphaminer
[21] http://www.ailab.si/orange/
[22] http://chirouble.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html

par les bibliothèques géo-spatiales. L'utilisation des ces bibliothèques est intéressante à plus d'un titre dans la mesure où elles offrent la possibilité de traiter différents types de géométries et de relations spatiales.

On peut subdiviser cette couche en sous couche traitant chacune différente composante de la donnée géo-spatiale. Ainsi, la sous couche « composante descriptive » s'occuperait des attributs descriptifs; tandis que la sous couche « composante géométrique » s'occuperait d'extraire les corrélations spatiales.

En plus des sous couches précédemment citées, il devrait également exister au niveau de la « couche traitement des données géo-spatiales » une sous couche chargée d'effectuer la pondération des résultats issus de chacune des sous couches (descriptive et géométrique) avant de les passer à la couche au dessus à savoir la couche fouille de données.

À titre d'exemple, intéressons nous à une tâche de clustering spatial. On considèrerait alors les relations spatiales comme des mesures de similarité à partir desquelles on déterminera quelles sont les entités géo-spatiales qui sont les plus proches. D'un autre côté, la composante descriptive se verra traiter grâce aux mesures de similarité propres à ce type de données (cf. Annexe C). L'utilisateur pourra choisir, en ce qui concerne la composante géométrique, les relations qu'il désire prendre en considération; et pour chacune des relations, attribuer une pondération.

Il va s'en dire qu'avec d'autres techniques de fouille de données, on procèdera tout autrement. Ainsi pour les algorithmes de génération de règles d'association ou de construction d'arbre de décision, nul besoin d'utiliser des mesures de similarité descriptive. En revanche, l'extraction de corrélations sur la base des relations géo-spatiales est pour le moins incontournable.

2.4.3. La couche fouille de données

La tâche effectuée au niveau de cette couche consiste en la fouille des données à proprement parler. Il s'agit à ce niveau d'utiliser différentes techniques de fouille afin d'extraire de la connaissance au sein des données. Plusieurs algorithmes peuvent être

utilisés à ce niveau mais la démarche d'intégration du type géo-spatial doit se faire selon la spécificité de chacun de ces algorithmes.

2.4.4. La couche présentation

Cette couche a pour rôle, essentiellement la présentation des résultats de la fouille. Une attention particulière devrait être portée à cette couche parce qu'elle représente le reflet des résultats de la fouille de données. En effet, c'est à travers la visualisation des résultats que l'on peut se faire une idée de ce que la fouille donne comme résultat.

Au niveau de la fouille « traditionnelle » de données, les résultats sont présentés à travers des tableaux et graphiques statistiques (nuages de points, droites de régression linéaire, etc.) ou des représentations de type arbres de décision.

Il est impensable de s'imaginer une fouille de données géo-spatiales sans une représentation cartographique des données. D'où la nécessité d'associer aux représentations traditionnelles (graphiques et tableaux statistiques), une visualisation cartographique qui bien entendu, se fera en associant une sémiologie adéquate. A titre d'exemple, dans une opération de clustering spatial, on pourrait s'attendre à voir les entités géo-spatiales qui sont dans le même cluster, avoir des attributs graphiques communs.

À propos de la visualisation des résultats, on pourrait pousser plus loin notre réflexion en imaginant une fouille basée sur la représentation cartographique. À partir des résultats d'une fouille initiale, l'utilisateur pourra sélectionner sur la carte, un sous ensemble de données géo-spatiales afin de restreindre la fouille à ces données.

La couche présentation des résultats, en plus de permettre une visualisation des résultats de la fouille, devrait être à même de fournir les résultats sous un format interopérable. Cela permettra de mettre à profit les résultats issus d'une fouille au sein d'un autre outil pour peu que celui-ci soit capable de lire un tel format de données. D'une manière plus spécifique, il s'agirait d'étendre le PMML[23] de sorte à tenir compte des données géo-spatiales.

---

[23] Prédictive Model Markup Language. Il s'agit d'un langage (le seul normalisé actuellement) de marquage basé sur XML conçu pour définir des modèles de données et visant à rendre interopérables les systèmes de datamining (source wikipedia.org)

2.5. Conclusion

Les précédentes approches de fouille de données géo-spatiales ayant montrées leur limites quand au traitement de la données géo-spatiales, il nous a paru nécessaire de réfléchir à une nouvelle approche de fouille de données tirant parti des avantages des précédentes approches et mettant à contribution les outils de fouille « traditionnelle » existants. C'est dans ce sens que nous avons proposé une approche intégrée de fouille de données qui supporte de façon complète, cohérente et transparente la composante géo-spatiale.

Afin de tirer parti des corrélations entre entités géo-spatiales, cette approche se fonde essentiellement sur les relations spatiales en allant au-delà de celle usuelles pour prendre en compte les relations directionnelles et celles liées à la similitude de forme. La sémantique associée à ces relations spatiales, selon l'algorithme utilisé, pourra éventuellement être différente. Ainsi face à des algorithmes fondés sur des mesures de similitude entre objets, des mesures de similarité pourraient être élaborées sur la base de ces relations. On pourra noter que pour la plupart des relations spatiales évoquées dans ce chapitre, la valeur obtenue se décline généralement sous deux(2) modes : quantitatif et qualitatif qui selon l'algorithme utilisé, pourrait être plus ou moins adaptés.

Par ailleurs, pour certaines relations qui dans la plupart du temps donnent comme résultats une valeur qualitative, en l'occurrence les relations topologiques, nous avons trouvé qu'il était nécessaire, dans le cadre d'une fouille de données, de prendre en considération des mesures topologiques détaillées. Pour d'autres relations, notamment celles directionnelles, il était important d'aller au delà de la mesure qualitative fournie, pour prendre en considération des mesures de distance entre points cardinaux afin de pouvoir utiliser de telles relations dans certaines algorithmes de fouille de données (clustering, k plus proches voisins par exemple).

Dans le souci de faciliter la mise en œuvre de notre approche, nous avons proposé un cadriciel qui décrit de façon schématique les différentes étapes à franchir lorsqu'on désire procéder à l'enrichissement d'un outil de fouille « traditionnelle ». Ce cadriciel se

veut général certes, mais pour chaque outil enrichi « spatialement », il va sans dire qu'il faut tenir compte de la spécificité dudit outil. Aussi, il est important de mentionner que le cadriciel proposé peut être étendu avec d'autres couches pour peu que l'utilisateur ait des besoins spécifiques. On pourrait par exemple imaginer une couche permettant une interaction avec des cubes SOLAP plutôt qu'uniquement avec des sources de données transactionnelles.

L'approche proposée ainsi que le cadriciel restent pour le moment au stade de théorie. Pour démontrer la faisabilité de l'approche proposée, nous proposons une implémentation de la composante géo-spatiale dans un outil de fouille de données ainsi que l'enrichissement de certains algorithmes afin de tirer parti des connaissances éventuelles cachées au sein des données géo-spatiales. Ceci fait l'objet du chapitre suivant de ce mémoire.

Chapitre 3 - Implémentation et test de GeoKNIME un nouvel outil de fouille de données géo-spatiales

3.1. Introduction

La fouille de données est un condensé de plusieurs disciplines dont la finalité est l'extraction de connaissances depuis des entrepôts de données volumineux. À cette fin, plusieurs outils ont été mis en œuvre au sein desquels sont implémentés plusieurs algorithmes de fouilles. De plus en plus, ces outils renferment d'autres fonctionnalités notamment celles d'analyse des données, de prétraitement et d'interprétation des résultats, qui vont au-delà de la fouille proprement dite.

Il est bon, dans une démarche de « spatialisation » d'un outil de fouille « traditionnelle », de choisir un outil offrant diverses fonctionnalités et ce pour plusieurs raisons. Premièrement parce que l'extraction de connaissances ne se limite pas à la tâche de fouille de données, contrairement à ce que l'on a pu penser de par le passé. L'extraction couvre d'autres étapes toutes aussi importantes que la fouille elle-même (cf. section 1.2.2.1). La seconde raison qui pourrait être un corollaire de la première, serait que l'existence de telles fonctionnalités, aiderait à gagner en temps de traitement d'autant plus dans un contexte de fouille géo-spatiales quand on sait que ces données sont complexes. En effet, passer par les étapes de compréhension des données, d'analyse exploratoire, etc. permettrait de s'intéresser aux sous-ensembles des données géo-spatiales « digne d'intérêt ».

Ceci pour dire que le choix d'un outil offrant diverses fonctionnalités est une étape importante dans la démarche de « spatialisation ». Et plus, ce choix ne devrait pas être fondé que sur les fonctionnalités de l'outil mais doit aller au-delà pour couvrir d'autres critères tels l'extensibilité, la modularité, la licence d'utilisation, la robustesse, les fonctionnalités offertes (cf. section 2.4.1 et Annexe A).

Aux termes d'une revue d'un certain nombre d'outils de fouille de données sous le respect des critères dont nous évoquions plus haut, nous avons jeté notre dévolu sur l'outil open source de fouille KNIME (cf. Annexe A).

Dans les sections qui suivent, nous décrirons l'enrichissement de cet outil avec la composante géo-spatiale ainsi que la mise en œuvre d'algorithmes permettant d'effectuer un Géo-clustering, une Géo-classification basée sur les K plus proches voisins et la construction d'arbres de décision spatiaux (cf. Annexe B pour la justification du choix de ces algorithmes).

3.2. KNIME (Konstanz Information Miner)

(KNIME, 2009) KNIME est un logiciel de fouille de données disponible sous une double licence : une licence open source(GPL) et une propriétaire. Construit autour de l'API d'Eclipse, KNIME bénéficie d'une grande modularité et d'une interactivité certaine grâce à ses multiples fonctionnalités qui peuvent être structurées sous forme de flot de données.

L'architecture de KNIME a été construite autour de trois (3) principes majeurs *(Berthold, et al., 2006)* :

- Framework interactif et visuel : possibilité de combiner divers composants en vue de mettre en œuvre des opérations de fouille de données.
- Modularité : indépendance totale ou moindre entre les composants et flexibilité des types de données. En effet aucun type n'est prédéfini, de nouveaux types peuvent être ajoutés facilement et déclarés compatibles avec ceux existants.
- Extensibilité : l'outil devrait être extensible sans grande modification.

Les fonctionnalités sous KNIME sont distribuées sous forme de composants représentées par des nœuds qui peuvent être inter reliés par des arcs pour former un flot de données. Les segments propagent les données entre les nœuds qui disposent chacun d'un état à un instant donné (configurer, en exécution, exécuté).

Comme fonctionnalités sous KNIME on note celles:

1. d'accès aux sources de données : elles permettent d'accéder à des sources aussi diverses que les fichiers, les bases de données grâce à JDBC mais aussi d'y écrire. La nouveauté dans KNIME réside dans la possibilité de lire et d'écrire des fichiers au format PMML (beta).
2. de manipulation et de transformation de données : ils permettent d'assurer les opérations de filtrage de colonnes, trie, jointure, fusion, échantillonnage de données
3. de fouille de données comportant des algorithmes de clustering, d'association, d'induction de règles, de régression, etc.
4. de réalisation d'opérations statistiques du type EDA[24] telles que les corrélations linéaires, le comptage de valeurs, de régression linéaire et polynomiale.
5. de visualisation permettant de se faire une idée sur les résultats de la fouille de données

Comme nous l'avons noté ci haut, KNIME est hautement extensible (cf. Figure 3.2-1) dans le sens où l'on peut y ajouter des extensions provenant d'autres outils de fouille de données (ex. Weka), des composants provenant d'outils statistiques et graphiques (ex. JFreeChart) mais aussi ses propres extensions en redéfinissant sans grande difficulté quelques classes.

[24] Exploratory Data Analysis ou Analyse Exploratoire de Données

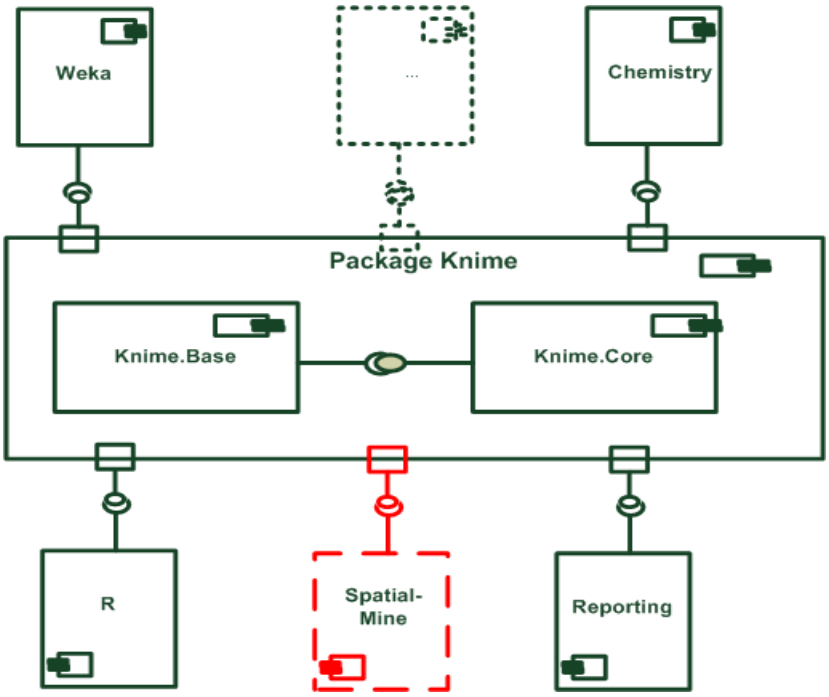


Figure 3.2-1 : Architecture de KNIME

Comme on peut le noter sur le diagramme de composants ci-dessus (cf. Figure 3.2-1), les différents modules de KNIME sont organisés autour des paquetages :

- *Knime.core* (contient les principales classes autour desquelles les fonctionnalités de base de KNIME sont développées) et
- *Knime.base* (contient les classes qui assurent les fonctionnalités de base).

Autour de ce noyau de base, peuvent se greffer d'autres composants. Le module assurant le support de la fouille géo-spatiales sera d'ailleurs intégré sous forme de plugin à l'architecture de KNIME.

L'ensemble des fonctionnalités seront réalisées grâce au concours de bibliothèques de traitement de données géo-spatiales open source notamment :

- JTS[25] : pour le support de la géométrie principalement la manipulation des différents types de géométries ainsi que des différentes relations spatiales
- GeoTools[26] : pour le traitement des géométries (lecture, visualisation, etc.)

[25] http://www.vividsolutions.com/jts/jtshome.htm

- GeOxygene[27] : pour certaines mesures de dissimilarité (Hausdorff, Distance surfacique)

Aussi, le choix de ces différentes bibliothèques trouve sa justification dans le fait que celles-ci offrent des fonctions permettant et le traitement de tout type de géométrie, et le traitement de plus d'une relation spatiale. Ce qui par ailleurs convient dans notre contexte dans la mesure où notre objectif est de proposer une approche de fouille traitant tout type de géométries et de relations. À terme, ce plugin se verra enrichi avec d'autres fonctionnalités telles celles permettant une réduction de dimensionnalité, d'extraction de sous ensembles intéressants de données, d'analyse exploratoire.

Il est important de noter que KNIME offre des outils permettant de réduire le temps de développement de ces plugins. Ces outils vont des générateurs de nœuds (unité de fonctionnalité sous KNIME) à ceux de génération de fichier XML qui décrivent la liaison entre le plugin et le noyau de KNIME. Nous avons mis à contribution ces outils pour assurer l'intégration de la composante géo-spatiale et ce de façon transparente, complète et cohérente au sein de KNIME.

3.3. GeoKNIME : un outil intégré de fouille de données géo-spatiales

L'enrichissement de l'outil de fouille de données « traditionnelle » KNIME en vue du support des l'information spatiale a consisté à développer un certain nombre de fonctionnalités permettant la manipulation de la donnée géo-spatiale à toutes les étapes de la fouille de données. Cela a donné naissance à un nouvel outil dénommé GeoKNIME dont la nouvelle architecture est présentée ci-dessous (cf. **Erreur ! Source du renvoi ntrouvable.**).

[26] http://www.geotools.org/
[27] http://oxygene-project.sourceforge.net/

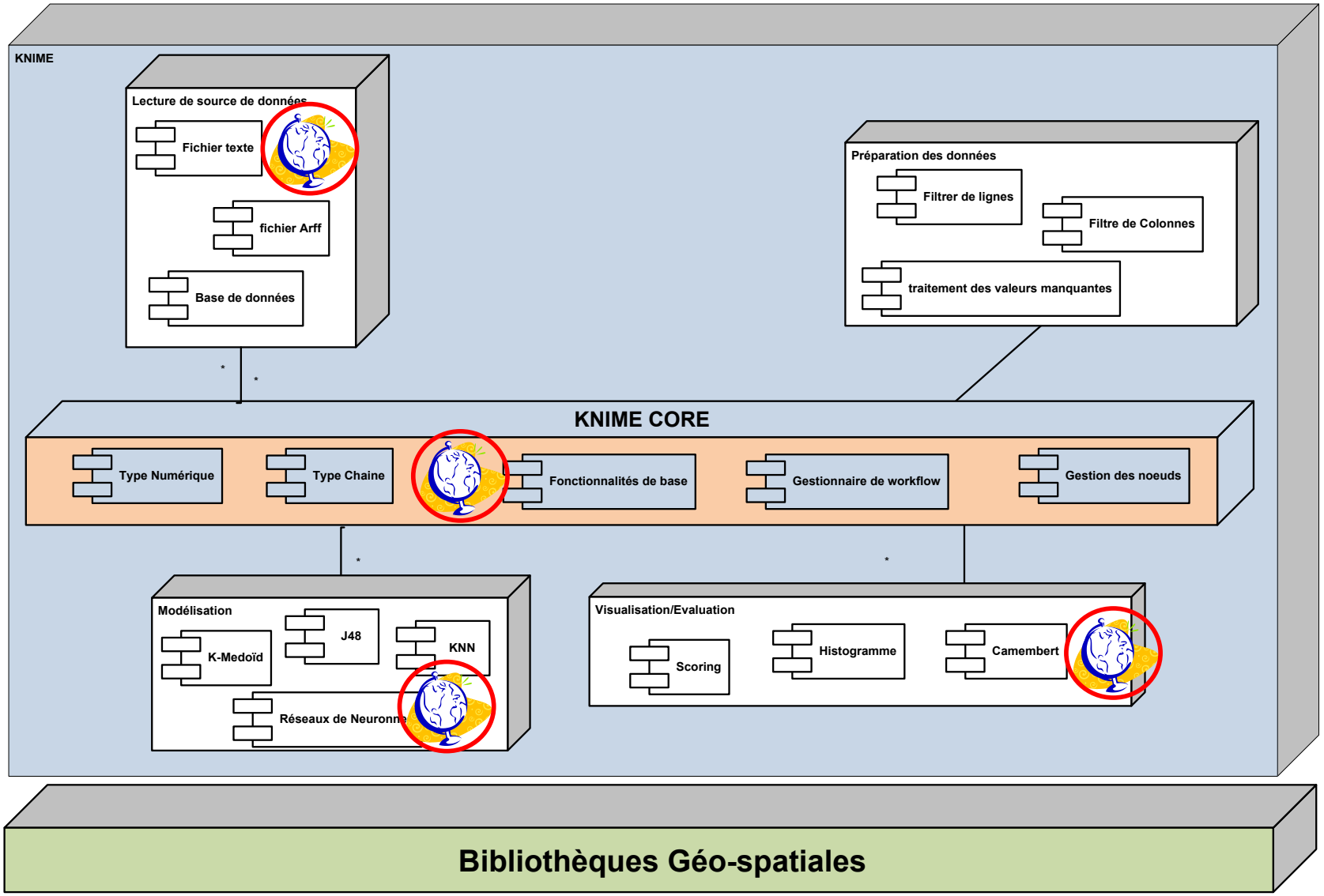


Figure 3.3-1 : architecture de GeoKNIME avec mention des composants ayant été enrichis spatialement

3.3.1. Le support du type géo-spatial

L'intégration de la composante géo-spatiale au sein d'une bibliothèque de fouille de données passe par la reconnaissance au sein dudit outil de l'information géométrique. Afin de permettre une manipulation aisée de ce type d'information au sein des outils de fouille, il est nécessaire de l'ériger en un type de données à part entière qui dispose de ses propres fonctionnalités de traitement. Cela permet de faire tomber la barrière de la complexité liée aux données géo-spatiales et du même coup dé-complexifie l'approche de la fouille de données géo-spatiales qui devient comme de la fouille « classique » à la différence que celle-ci implique des données d'un autre type en l'occurrence celui géométrique.

Fort heureusement, de par sa modularité et son extensibilité, KNIME permet d'arriver à de telles fins. En effet, KNIME permet la personnalisation ou la définition de nouveaux types de données au besoin. À proprement parler, la création de nouveau type se fait en deux(2) étapes :

1. La création d'une interface qui implémente *DataValue* qui est une interface exposant des méthodes permettant l'accès aux méta-informations d'un *DataCell*
2. La création d'une classe qui hérite de DataCell et implémente l'interface créée en (1). Le DataCell représente une cellule de données à laquelle est rattaché un type de donnée particulier.

En suivant cette démarche, nous avons réussi à intégrer un nouveau type de données, en l'occurrence le type geometry, dans KNIME. L'interaction entre les classes créées et celles existantes est donnée par les Figure 3.3-2 et Figure 3.3-3 qui décrivent respectivement cette interaction d'un point de vue statique et dynamique.

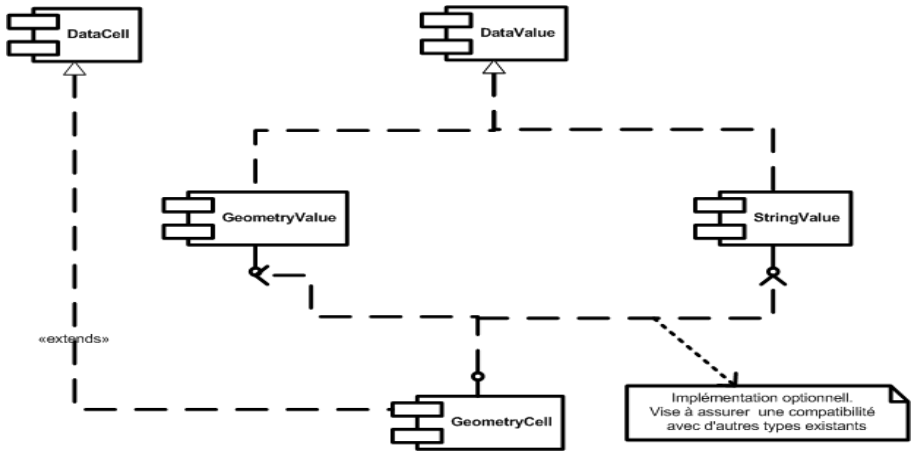


Figure 3.3-2 : Diagramme de composants – intégration d'un type géométrie dans l'outil KNIME

Comme on pourra le remarquer sur la Figure 3.3-3, la géométrie sous KNIME est construite grâce à la bibliothèque géo-spatiale open-source GeoTools qui manipule de façon sous-jacente des géométries JTS[28]. L'information géométrique est contenue à l'intérieur d'une *GeometryCell* qui représente une cellule d'information (à la manière d'un tableur). Les méta-informations concernant ce type sont fournies par la classe/interface *GeometryValue*.

[28] Java Topology Suite

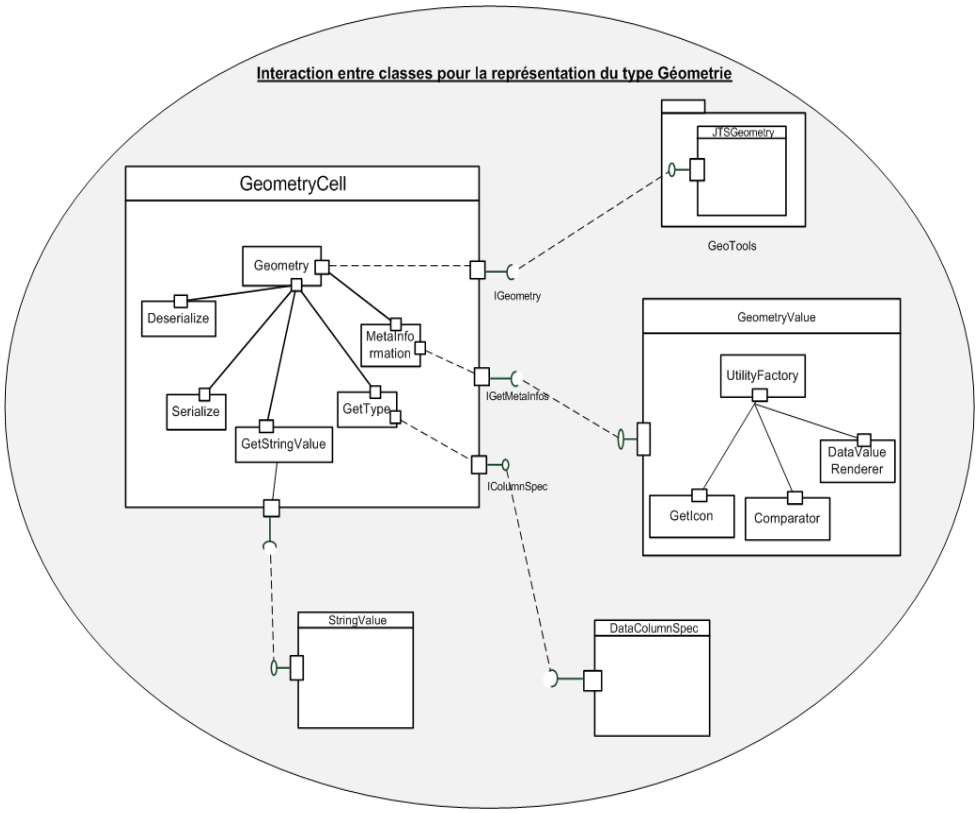


Figure 3.3-3 : interaction entre les classes lors de la représentation du type Geometry

L'information géométrique est représentée sous un format WKT[29]; ce qui permet de reconnaître facilement le type de géométrie affiché (polygone, point, ligne, etc.). Mais on pourrait toutefois envisager de créer des sous types de géométrie. En lieu et place donc d'avoir un type Geometry générique (cf. Figure 3.3-4), nous aurons des sous-types de géométrie. On aura ainsi un sous type POINT, LIGNE, POLYGONE, MULTILIGNE, etc. cela permet d'effectuer des traitements personnalisés (ex. utilisation de la distance de Hausdorff exclusivement sur des entités linéaires) pour chaque sous type en lieu et place d'utiliser de multiples structures conditionnelles dans le code.

[29] Well Known Text (cf. http://en.wikipedia.org/wiki/Well-known_text)

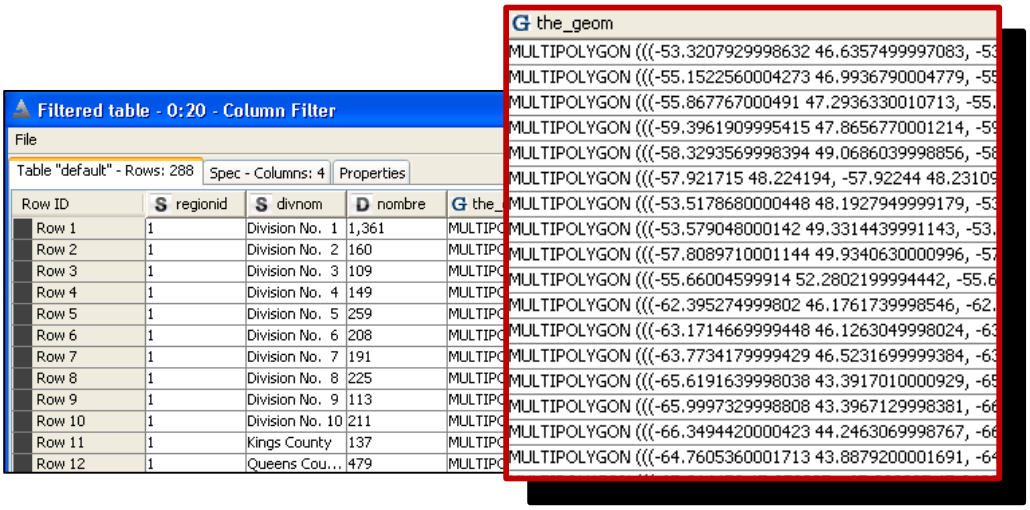


Figure 3.3-4: le type de données Geometry sous KNIME[30]

3.3.2. Mise à contribution du type Geometry pour la réalisation des tâches de fouille de données

Une fois franchie l'étape de création d'un type géométrie sous KNIME, la phase suivante consistait à développer des fonctionnalités de fouille de données mettant à contribution ce nouveau type. Pour ce faire, divers nœuds de traitement ont été développés en amont et en aval de la tâche de fouille proprement dite. Ces nœuds regroupent les fonctionnalités de lecture de données géo-spatiales ainsi que la visualisation des résultats de la fouille.

En ce qui concerne la lecture de données géo-spatiales, elle se résume pour l'instant à l'extraction de données depuis une base de données PostGIS (cf. Figure 3.3-5). Cette fonctionnalité pourrait être étendue facilement à la lecture de fichier contenant des informations géographiques (ex. Shapefile) et à d'autre base de données géo-spatiales. Plus important, il pourrait être intéressant de permettre la lecture de données depuis des cubes de données SOLAP. Cela offre, entre autre avantage, un gain de temps en permettant une fouille sur des données préalablement agrégées.

[30] Le type Geometry est identifiable grâce à l'icône G à l'extrême gauche de l'entête de la colonne.

La fonctionnalité de visualisation a été développée en aval de la fouille de données afin que l'utilisateur puisse se rendre compte visuellement des résultats de la fouille et les interpréter.

Pour les fonctionnalités de fouille, avec le support du nouveau type Geometry sous KNIME, nous avons enrichi spatialement deux(2) grandes classes d'algorithmes de fouille choisie principalement en raison de leur popularité et de la facilité de mise en œuvre (cf. Annexe B):

- Le clustering : à ce niveau nous avons implémenté une fonctionnalité de calcul de matrice de distance basée d'une part sur les relations métriques, topologiques et de similarité de formes. Ces matrices de distances sont ensuite utilisées pour effectuer un clustering basé sur K-Medoïd.

- La classification : nous avons introduit le type géométrique au niveau de la construction des arbres de décision ainsi qu'au niveau de la prédiction basée sur les K plus proches voisins (KNN)

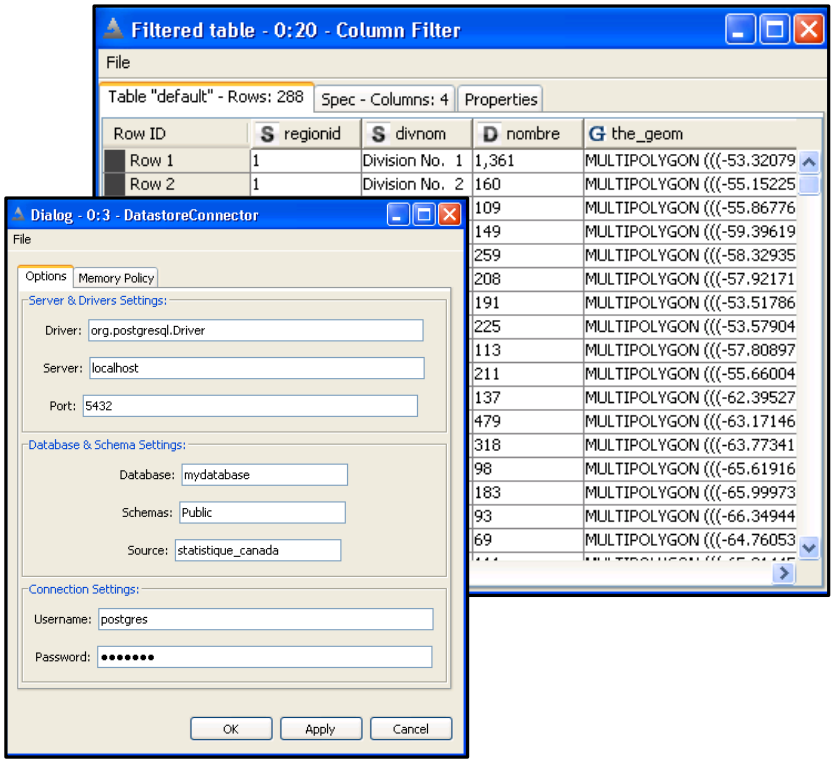


Figure 3.3-5 : Connexion et lecture d'une base de données géo-spatiales

3.3.3. Enrichissement spatial de quelques algorithmes de fouille de données « traditionnelle »

3.3.3.1. Géo-clustering basé des relations métriques, topologiques et de similitude de forme

Après l'intégration du type Geometry sous KNIME, nous avons procédé à l'implémentation de fonctionnalités de clustering qui met à contribution ce nouveau type. Comme nous le notions dans la section précédente, la fonctionnalité développée consiste au calcul d'une matrice de dissimilarité[31] basée sur les relations métriques, topologiques et de reconnaissance de forme. À l'heure actuelle et par manque de temps, le calcul d'une matrice de distance basée sur les relations directionnelles n'est pas encore mis en œuvre.

[31] Voir http://en.wikipedia.org/wiki/Distance_matrix pour plus de détails

Au niveau du Géo-clustering, des mesures de dissimilarité basées sur les relations spatiales sont construites. À titre d'exemple, avec les relations métriques spatiales, plus la distance séparant deux(2) entités géo-spatiales est minime, plus ces deux(2) entités sont susceptibles de s'influencer mutuellement.

Le Géo-clustering sous KNIME offre différents avantages au nombre desquels on peut citer :

- La prise en compte des attributs descriptifs lors de la fouille : en effet le clustering ne se fait pas exclusivement avec les attributs géométriques des données géo-spatiales mais tient aussi compte des attributs descriptifs.

- La normalisation des résultats obtenus : il s'agit de ramener les distances calculées à une valeur comprise entre 0 et 1 en divisant l'ensemble des valeurs par leur maximum. Pour rappel, l'objectif de la normalisation est d'éviter que l'échelle de mesure d'un attribut ne le rende plus prépondérant par rapport aux autres. À titre d'exemple, supposons deux attributs numériques taille et fortune donnant respectivement les tailles d'individus (exprimées en mètre) et leur fortune (exprimée en millier de dollar). En effectuant sur ces données un calcul sans normalisation, l'attribut taille aura moins d'incidence dans les résultats parce qu'exprimé dans une petite échelle.

- La pondération des attributs intervenant dans le clustering : cette opération est effectuée après la normalisation et permet la mise en valeur d'un attribut par rapport à un autre.

- La possibilité de réaliser un géo-clustering intra ou inter thème : le géo-clustering intra-thème réfère à la manipulation des données géo-spatiales d'un même thème. Il pourra s'agir à titre d'exemple de voir quelles sont les zones de criminalité – localisées spatialement - qui partagent des caractéristiques semblables. Le clustering inter-thème désigne la réalisation d'un clustering en considérant des thèmes différents. À titre d'exemple, le

regroupement de secteurs selon la distribution géo-spatiale de source de pollution.

Comme on le note sur la Figure 3.3-6, l'utilisateur effectuant un géo-clustering choisit de lui-même les relations qu'il désire voir impliquées dans l'opération de clustering et attribue les différentes pondérations.

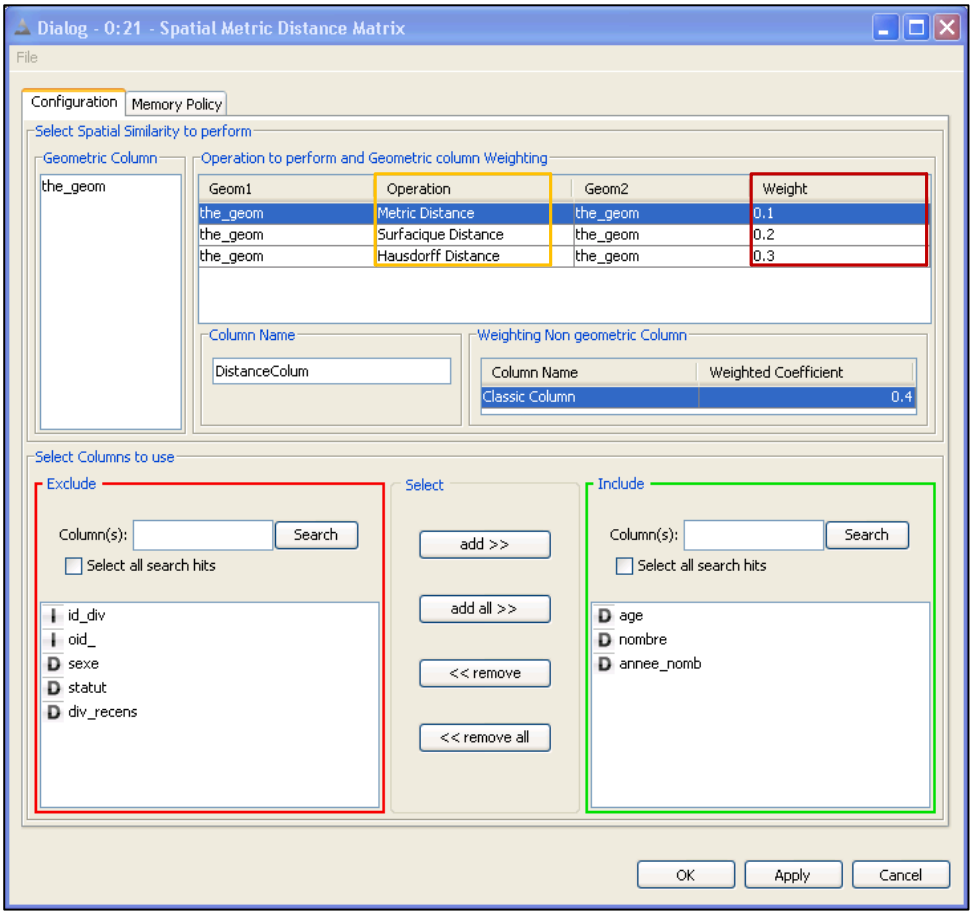


Figure 3.3-6: Fenêtre de paramétrage d'un Géo-Clustering basé sur les relations spatiales métriques et de similitude de forme

Afin de prendre en compte les attributs descriptifs dans le calcul de la matrice de distance, la formule suivante est appliquée pour un calcul intra-thème:

En notant P et Q deux(2) tuples extraits d'un ensemble de données géo-spatiales et possédant bien entendu une colonne géométrique, la distance entre P et Q est donnée par :

$$D(P,Q)=\sum_{i=1}^{n}[(a_i-b_i)^2]+ D_g$$

Où D_g représente la distance « géométrique » séparant les tuples P et Q et

$\sum_{i=1}^{n}(a_i-b_i)^2$ représente le carré de la distance euclidienne entre les attributs descriptifs.

Notons que D_g pourra être la distance de Fréchet, de Hausdorff, directionnelle ou une mesure de similitude de forme

En ce qui concerne le calcul inter-thème, il est un peu plus compliqué que le premier. En effet, pour chaque géométrie, il faut évaluer les distances séparant celle ci de toutes les géométries contenues dans un autre thème.

En notant P et Q deux tuples de données géo-spatiales et R un ensemble de données géo-spatiales, la formule est donnée par :

$$D(P,Q)=\sum_{i=1}^{n}[(a_i-b_i)^2] + \sum_{k=1}^{m}[(D_{PR_k}-D_{QR_k})^2]$$

Où D_{PR_k} représente la distance séparant la géométrie de P et la $k^{ième}$ géométrie de R.

Il est évident que ces différents calculs ne sont pas sans incidence sur les performances du système.

3.3.3.2. Géo Classification basée sur les k plus proches voisins

Nous avons également mis à profit le type Geometry pour enrichir spatialement l'algorithme de classification basée sur les k plus proches voisins (KNN). Le principe de calcul de distance demeure le même que pour le Géo-clustering à la différence que la Géo-Classification consomme moins de ressources. En effet, le Géo-Clustering implique généralement le nombre d'opérations suivant :

- Dans le cas d'un calcul intra-thème N^2 où N représente le nombre de géométries
- Dans le cas d'un calcul inter-thème, ce nombre est de $N*M$ ou N est le nombre de géométries du premier thème et M le nombre de géométries du second.

Or pour la Géo-Classification, le nombre total d'opérations impliquant des géométries est de $p*q$ avec p et q inférieur au nombre de géométrie du thème considéré.

Pour rappel, la classification basée sur le KNN permet de prédire la valeur d'une classe - généralement nominale (chaine de caractère) – en se basant sur la valeur de ses proches voisins; le nombre de voisin étant fourni en paramètre (cf. Annexe B). En plus de tenir compte des attributs descriptifs, la Géo classification basée sur le KNN permet de prendre en compte la proximité géo-spatiale. Cette proximité se limite actuellement aux relations métriques mais peut bien entendu s'étendre aux autres types de relations géo-spatiales.

3.3.3.3. Arbre de décision spatial

Le type Geometry a été enfin mis à contribution pour l'implémentation d'un algorithme permettant la construction d'un arbre de décision. Dans la sphère de la fouille de données, on note que les arbres de décision sont une des techniques les plus intuitives, simples et populaires. En effet, ils permettent de repartir un ensemble de données en fonction d'une variable cible et de variables discriminantes fournies en paramètre.

Parmi la multitude d'algorithmes de construction d'arbres de décision existant sous KNIME, nous avons choisi d'enrichir spatialement l'algorithme J48 qui est tiré du célèbre outil de fouille de données open-source WEKA[32]. L'un des avantages de cet algorithme est qu'il n'est pas besoin, à la différence d'autres, de discrétiser les variables numériques avant parce que l'algorithme effectue cette tâche à la volée.

[32] http://www.cs.waikato.ac.nz/ml/weka/

Comme nous le mentionnions dans les paragraphes plus hauts, le principe de la construction d'un arbre de décision est de se servir d'attributs numériques et chaines de caractères comme facteurs discriminants (i.e. qui apparaîtront comme des règles de décision une fois l'arbre produit). Pour cela, les algorithmes d'arbre de décision prennent en considération toute la table de données en entrée et calcule pour chaque attribut, l'indice d'impureté ou d'homogénéité. En effet, plus une variable contient des valeurs uniques, plus elle est impure et reste difficile à utiliser pour construire l'arbre de décision. Le problème majeur avec la composante géométrique de l'information géo-spatiale est que cette dernière est à priori inexploitable car ne pouvant pas servir de facteur discriminant dans la construction de l'arbre. Il est donc nécessaire d'extraire les relations entre ces données et de les fournir à l'algorithme. La construction de l'arbre de décision spatial s'effectue comme suit :

- Extraction des relations entre entités géo-spatiales : cette étape est la plus exigeante en ressource car nécessitant un nombre d'opérations de l'ordre de N^2 ou N représente le nombre de géométries.

- Discrétisation personnalisée des relations extraites s'il ya lieu : cela en notant que les relations extraites sont de nature quantitative. par exemple lorsque la relation extraite est la distance euclidienne, l'objectif de la discrétisation personnalisée est de transformer la distance quantitative obtenue en qualitative du type proche de, loin de, etc. ce qui est d'autant plus compréhensible une fois l'arbre produit. Bien entendu, lorsque la relation considérée est de type qualitatif, par exemple une relation directionnelle, il n'est plus besoin de passer par cette étape de discrétisation. Il est important de noter que même pour des valeurs quantitatives, la discrétisation personnalisée peut être optionnelle dans la mesure où l'algorithme permet une discrétisation à la volée.

- Traitement des données extraites : les données extraites depuis les géométries sont jointes aux données descriptives pour être passées à l'algorithme.

On note ainsi qu'outre le temps d'extraction des corrélations géo-spatiales, la construction de l'arbre de décision spatiale est plutôt simple à réaliser. Par contre, il faut noter que l'algorithme d'extraction des corrélations est implémenté doublement : au niveau de l'algorithme permettant la construction de l'arbre (Learner) et de celui permettant de classer de nouvelles valeurs en fonction de l'arbre préalablement construit (Predictor).

3.4. Tests de l'outil et validation de l'approche

L'objectif de cette phase est de valider d'une part l'approche intégrée de fouille de données géo-spatiales que nous avons proposée et de tester les algorithmes enrichis spatialement. L'idée derrière les tests des différents algorithmes est de voir d'une part s'ils fournissent des résultats en conformité avec les réalités du terrain. D'autre part de voir si – pour la géo-classification – on peut accorder un certain crédit aux prédictions faites.

Pour cela, les données qui devront être utilisées pour les tests doivent être des données recueillies sur le terrain. Pour aller dans ce sens, nous avons utilisé des données concernant la criminalité de la ville de San-Francisco pour diverses périodes notamment 2003-2010. Ces données[33] concernent l'ensemble des appels au 911. Chaque appel est caractérisé par :

- Un numéro identifiant
- La catégorie de crime concerné
- La description du crime
- Le district de police concerné par l'appel
- La date et l'heure d'appel
- La localisation géographique sous forme de géométrie (point)

Les tests ont été effectués sur une machine de processeur Intel cadencé à 2.4Ghz avec une mémoire RAM de 2Go. Pour ces tests, nous avons principalement exploité la

[33] Ces données sont disponibles à l'adresse http://www.datasf.org/index.php?category=geography

relation spatiale de type métrique (la distance). En effet, après plusieurs jeux d'essai avec différent type de relation, celle qui offrait des résultats pertinents était celle métrique.

3.4.1. Géo-Clustering : Analyse de la typologie des crimes de San-Francisco

Pour l'opération de Géo-clustering, nous considérons un sous ensemble des données de San-Francisco, notamment les trois catégories de crime les plus répandus : prostitution, la violence armée et les drogues et narcotiques recensées pour l'année 2010. Ce que l'on cherche à démontrer à travers cette opération de fouille, est la classification de la ville de San Francisco par degré de crime et de voir si les résultats produits reflètent la réalité constatée sur le terrain.

Il est important de noter que contrairement à la Géo-classification, faire du Géo-clustering nécessite une bonne connaissance du domaine. En effet, contrairement à la prédiction basée sur les plus proches voisins (KNN) ou les arbres de décision, où le résultat de la fouille est assez explicite – car le résultat est palpable (la variable prédite) – au niveau de clustering, il faut pouvoir interpréter les résultats afin d'y déceler de la connaissance.

Plus exactement, la réalisation de cette tâche a consisté à mettre à profit des fonctionnalités existant sous KNIME et d'autres que nous avons développées pour en arriver aux résultats. La tâche a consiste principalement à:

- produire une matrice de distance spatiale sur la base de la relation métrique entre entités géo-spatiales – la distance séparant les différents points où est survenu un crime.
- Utiliser les résultats de la matrice de distance spatiale pour répartir les clusters grâce à l'algorithme K-Medoïd[34] déjà implémenté sous KNIME.

On peut, pour l'opération de Géo-clustering, tenir compte des attributs descriptifs et aussi appliquer une pondération à chaque attribut afin de privilégier une relation par rapport à une autre. Toutefois, afin de voir l'influence des relations géo-spatiales dans la

[34] K-Médoïd est le seul algorithme de clustering sous KNIME prenant en paramètre une matrice de distance pour construire les clusters

nature des résultats produits, nous sommes exclusivement intéressés à la composante géométrique.

La structure sous forme de workflow de l'opération de géo-clustering ainsi que la configuration de la matrice de distance spatiale sont illustrées aux Figure 3.4-1 Figure 3.4-2

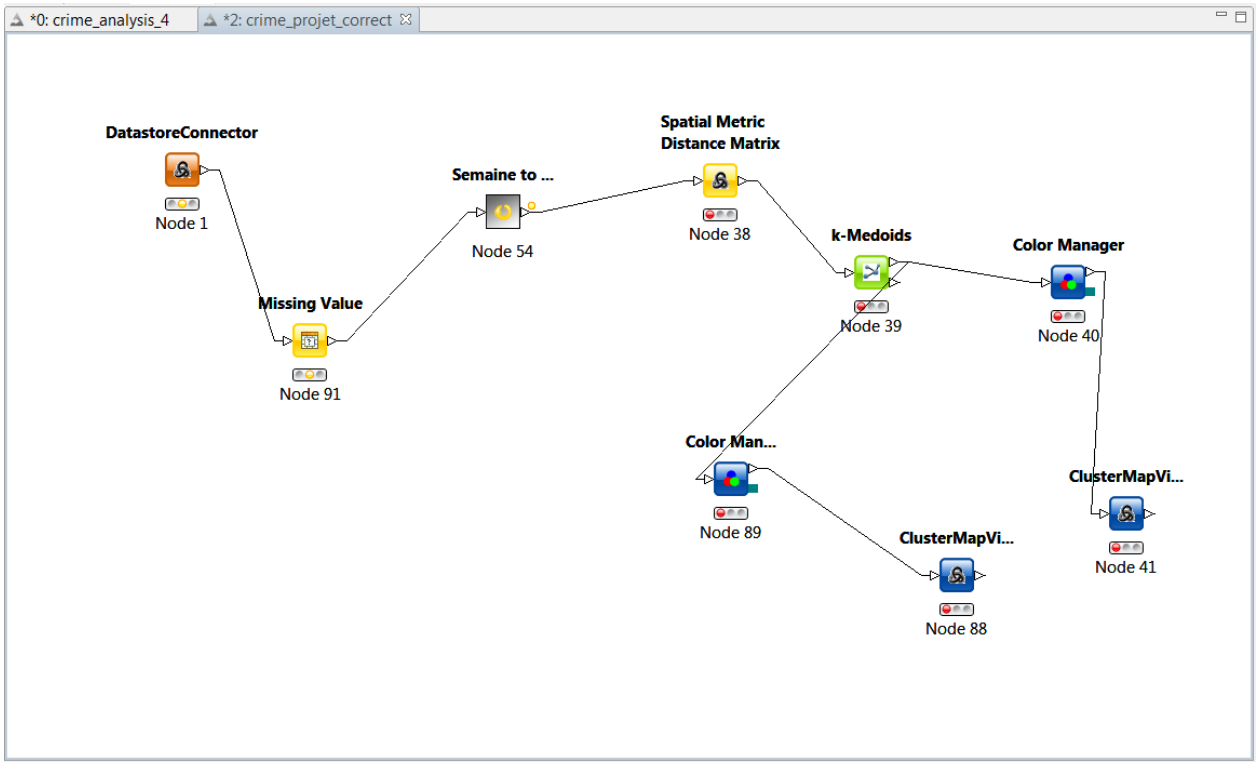


Figure 3.4-1 : structure sous forme de workflow de l'opération de Géo-clustering basée sur une relation métrique

La figure ci-dessus (cf. Figure 3.4-1) représente un ensemble de nœuds concourant à la réalisation d'une opération de Géo-clustering. Par soucis de clarté, la figure a été allégée afin de montrer les nœuds essentiels à la réalisation de l'opération. On note principalement les nœuds;

- d'accès aux données géo-spatiales (node1) : permet d'accéder à une source de données géo-spatiales stockée sous PostGis

- de traitement des valeurs manquantes (node 91) : permet d'appliquer un traitement particulier lorsqu'une valeur est manquante.
- de calcul d'une matrice de distance spatiale (node 38): nœud essentiel du géo-clustering, permet de calculer les distances (euclidienne, Hausdorff, Fréchet, ..) entre différentes entités géographiques en tenant compte des valeurs descriptives – lorsque spécifié par l'utilisateur.
- de clustering selon les K-Medoïds (node 39): prend en paramètre la matrice de distance spatiale et subdivise l'ensemble des données en un nombre fixe de clusters défini par l'utilisateur.
- de visualisation cartographique (node 41): permet de visualiser « cartographiquement » le résultat de l'opération de clustering.

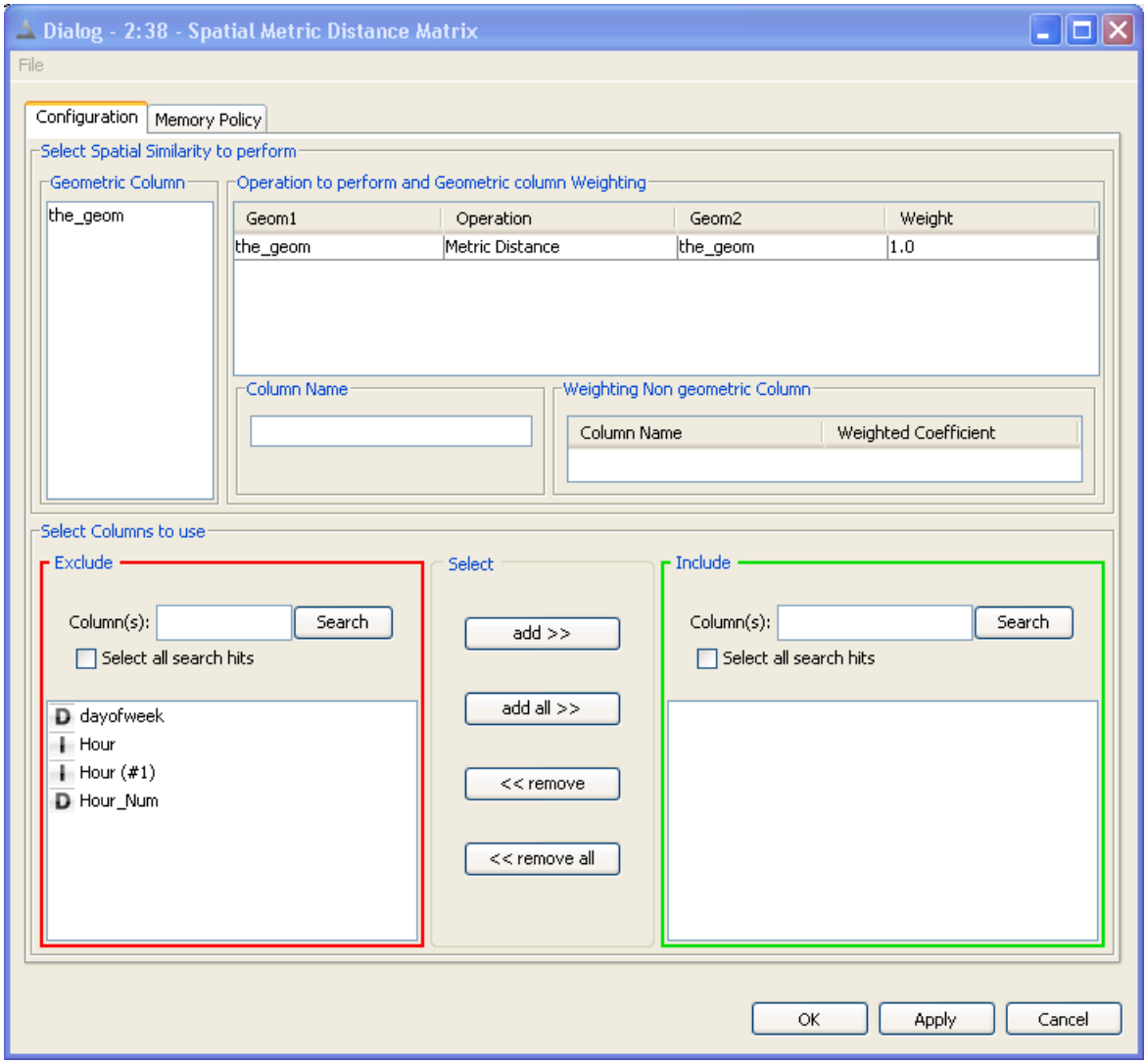


Figure 3.4-2 : configuration du calcul de la matrice de distance spatiale

Une fois la matrice de distance[35] calculée, nous mettons à profit le nœud de clustering Médoïds de KNIME en lieu et place de développer un autre nœud afin de partitionner les données en trois (3) groupes (cf. Figure 3.4-3). La visualisation « cartographique » des clusters produits donne le résultat suivant (cf. Figure 3.4-4):

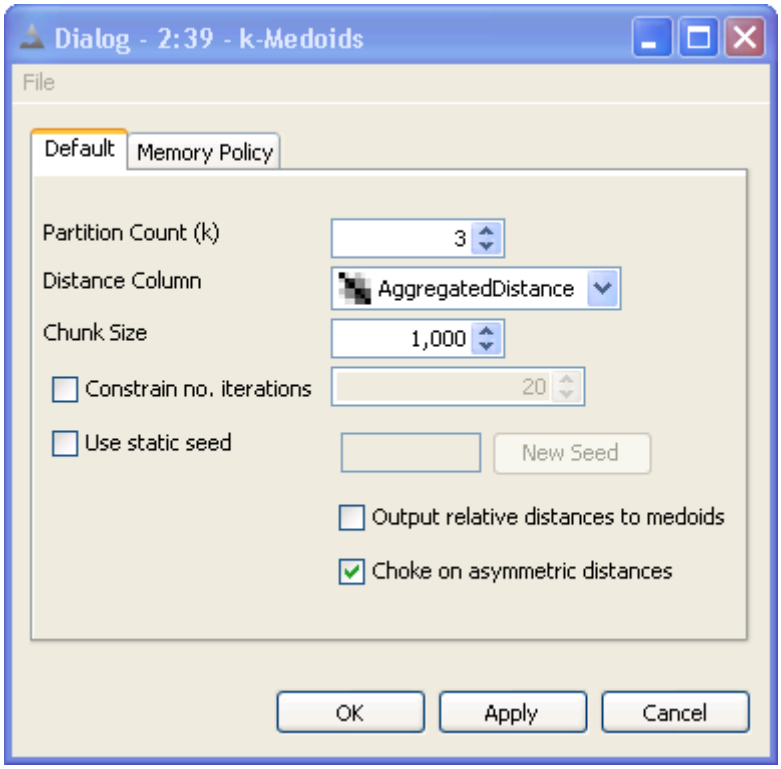


Figure 3.4-3: configuration du nombre de clusters

[35] Temps de calcul 6mn37s pour 5256 crimes analysés

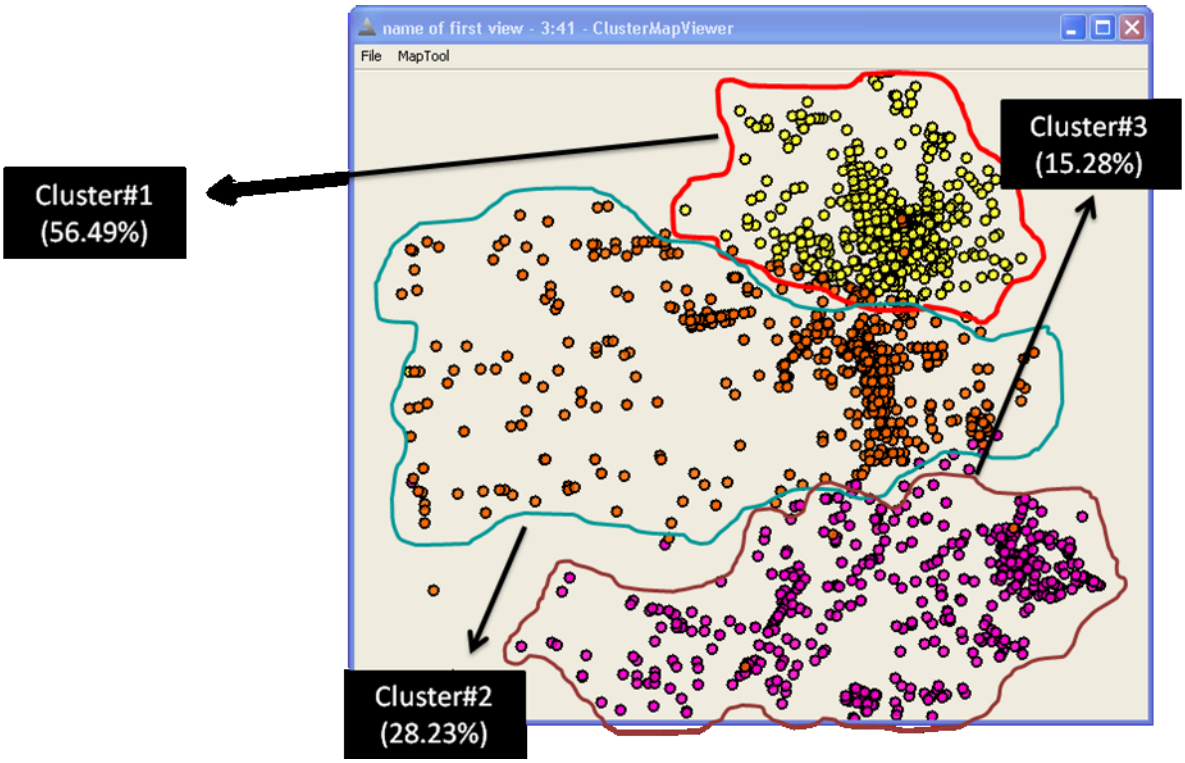


Figure 3.4-4: Répartition des clusters et pourcentage de crime

L'analyse de la typologie des crimes pour chaque cluster (cf. Figure 3.4-6, Figure 3.4-7) nous indique que les crimes liés à la drogue et aux narcotiques sont partout présents dans la ville de San-Francisco. On note toutefois que si le taux de crimes liés à la prostitution et violences armées est constant pour le cluster 1, ce taux varie pour les deux autres clusters où on note une prépondérance des crimes liés à la prostitution et aux violences armées respectivement pour les clusters 2 et 3.

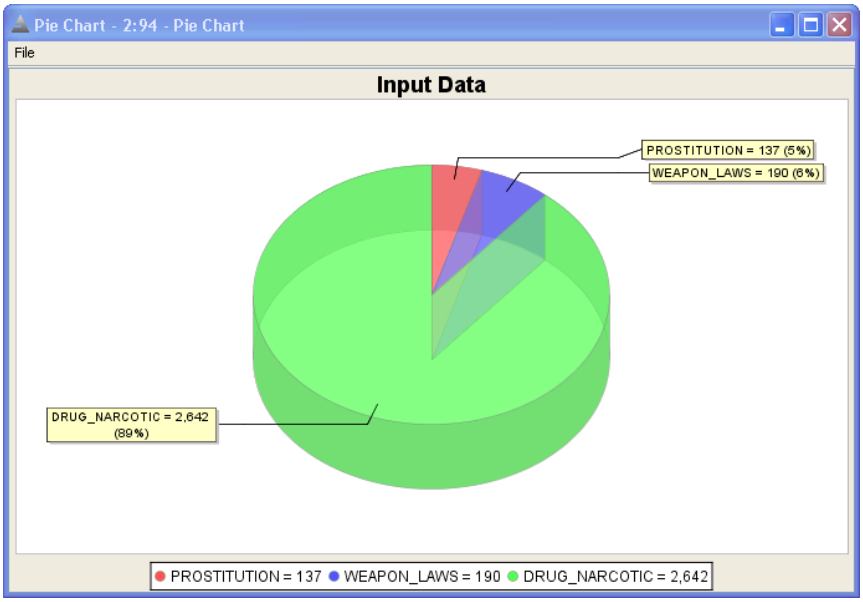


Figure 3.4-5: Typologie des crimes pour le cluster#1

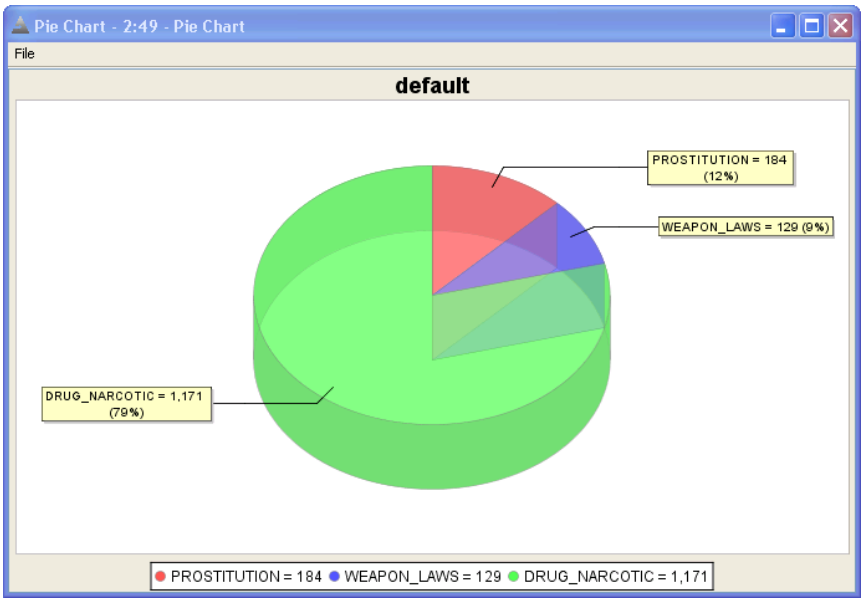


Figure 3.4-6: Typologie des crimes pour le cluster#2

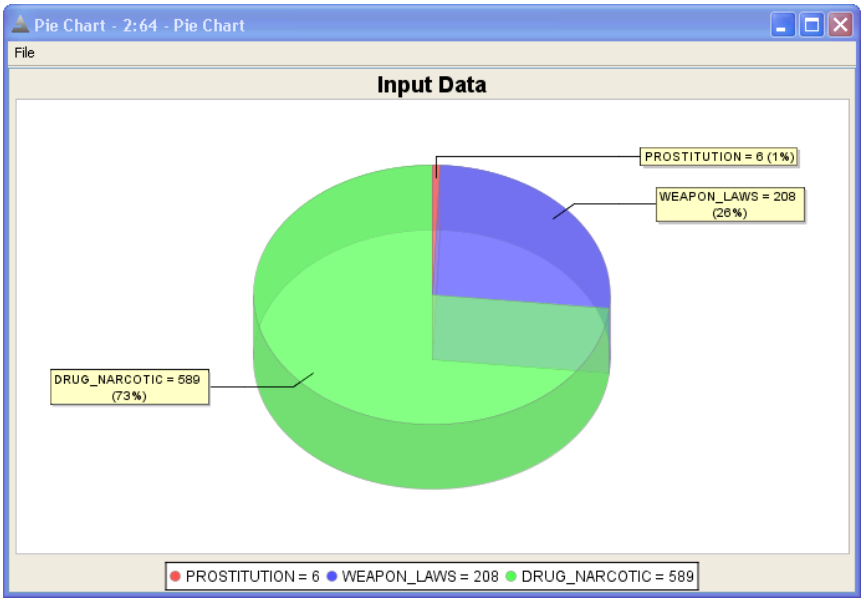


Figure 3.4-7: Typologie des crimes pour le cluster#3

Une analyse des résultats des clusters formés selon les différents quartiers de la ville de San Francisco nous indique que les quartiers les plus sujets à la criminalité sont Tenderloin, Mission et Bayview (respectivement pour les clusters 1, 2 et 3) (cf. Figure 3.4-8, Figure 3.4-9 et Figure 3.4-10). Ces quartiers qui constituent par ailleurs les centres de leurs clusters respectifs font partie des quartiers les plus pauvres de San-Francisco ou règnent effectivement violence, prostitution et drogue.

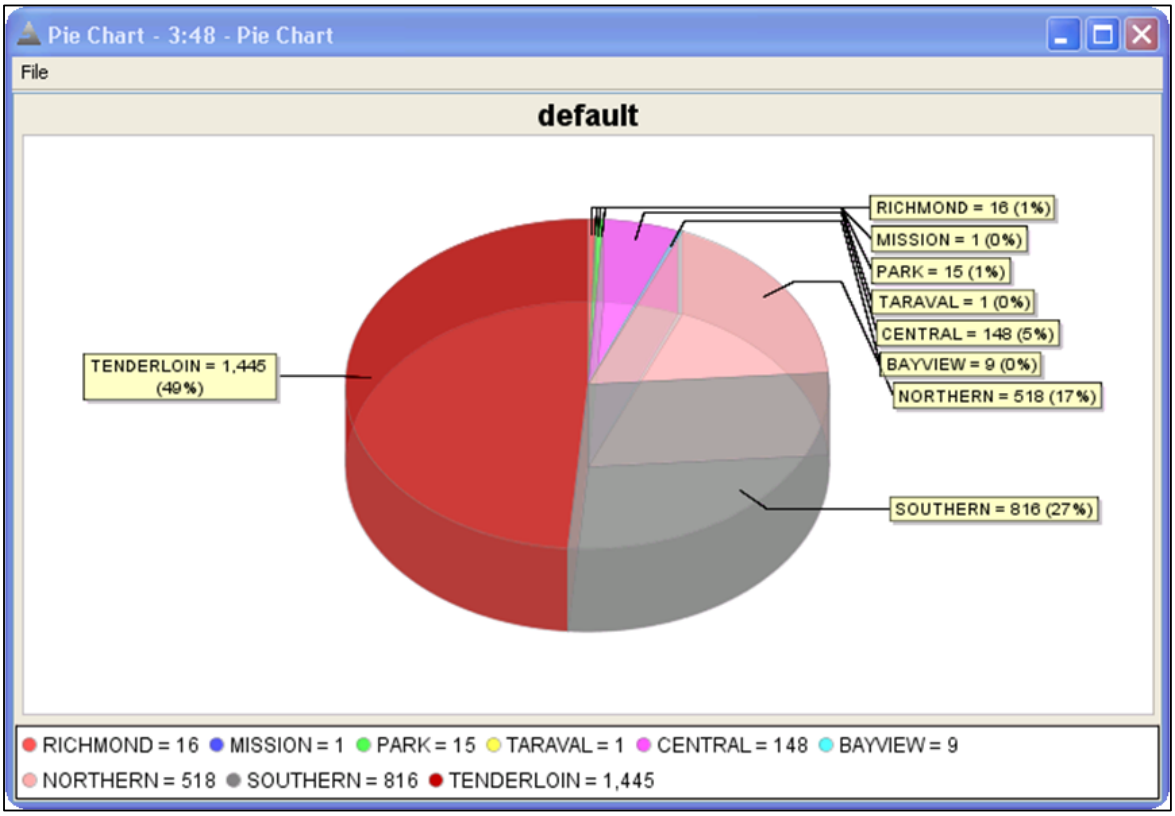


Figure 3.4-8: Répartition des crimes par quartier cluster#1

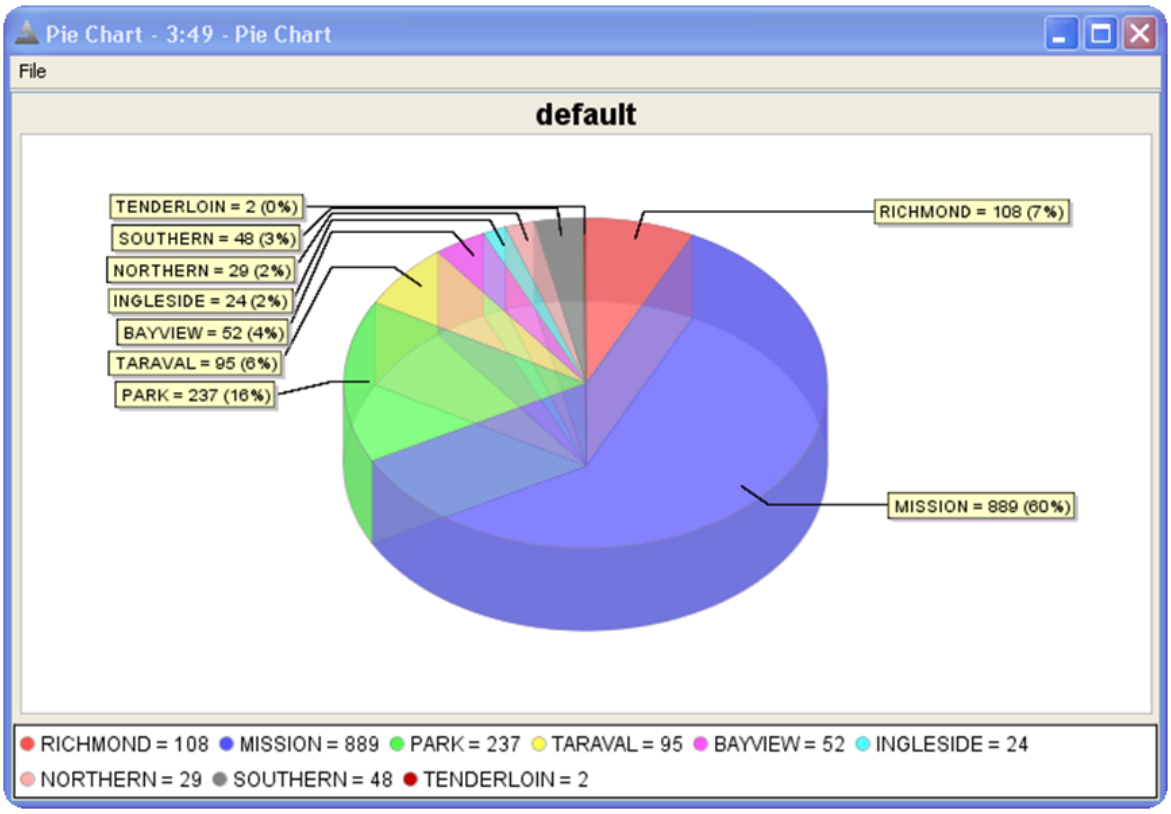


Figure 3.4-9: Répartition des crimes par quartier cluster#2

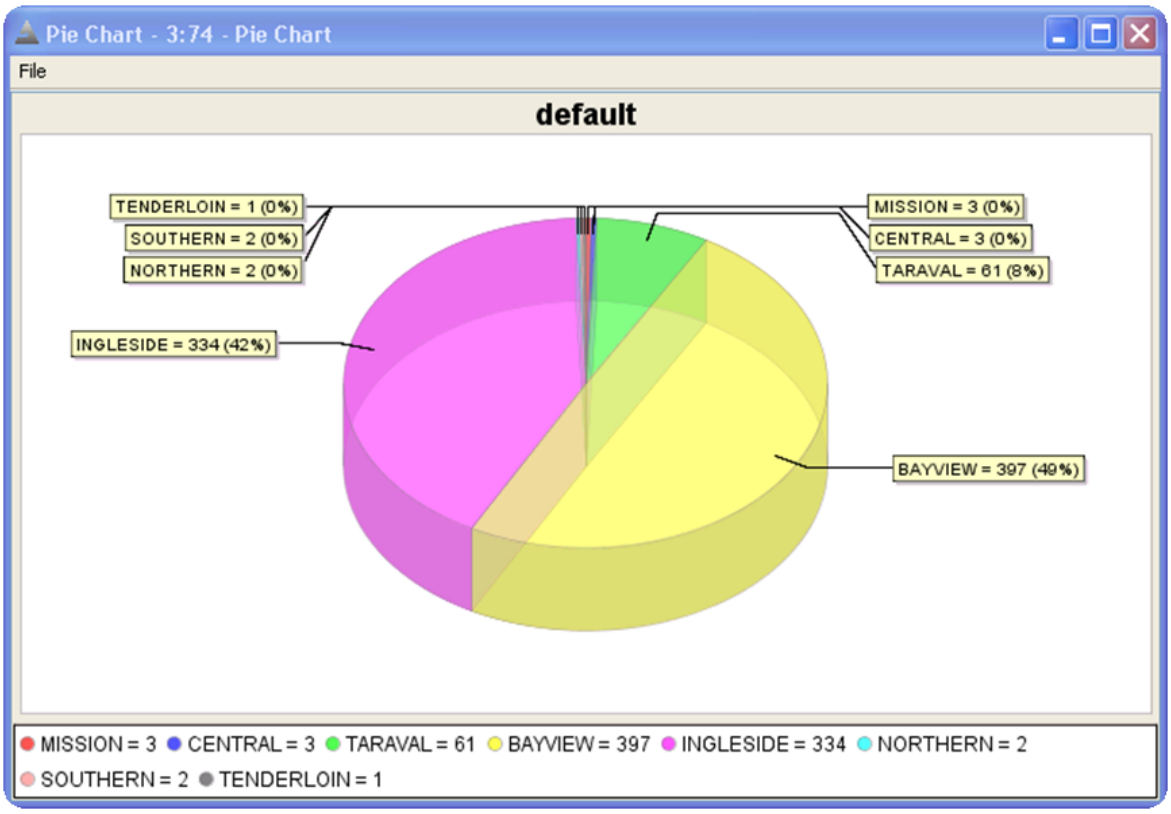


Figure 3.4-10: Répartition des crimes par quartier cluster#3

Pour conclure en ce qui concerne la fouille basée sur le géo-clustering, on peut dire que les résultats tendent à valider les constats faits sur le terrain; à savoir que les quartiers les plus pauvres sont les foyers des crimes.

3.4.2. Géo-KNN : Prédiction des catégories de crime de 2009

Les données antérieures de criminalité (2003-2008) ont été mises à profit pour effectuer la prédiction des catégories de crime de l'année 2009. L'algorithme utilisé pour effectuer cette tâche est le Géo-KNN. Plus pratiquement, la prédiction d'une catégorie de crime est faite sur la base des vingt(20) plus proches voisins.

Il faut noter au passage que le choix du nombre de voisins est subjectif et reste à la discrétion de l'utilisateur et expert du domaine. Dans notre contexte, nous avons lancé plusieurs tests en faisant varier le nombre de voisins pour finalement fixé ce nombre à vingt(20) afin de prendre en compte une variété de crime entourant la zone géographique dont la catégorie de crime est à prédire. Notons qu'un nombre plus élevé de voisin entraine

une régression de performance de l'algorithme sans pour autant augmenter le taux de confiance (pourcentage de valeurs vraies) des résultats prédits (cf. Annexe B – choix du K). Comme le note *(Larose, 2005)*, il est important de faire un compromis entre la précision de la prédiction et la généralisation des résultats issus de celle ci. En effet, plus on atteint un certain pourcentage dans la prédiction, plus les résultats de ladite prédiction ne peuvent pas se généraliser à l'ensemble des données (cf. Figure 3.4-11). D'où la nécessité de choisir un nombre de K optimal. Dans notre contexte, ce nombre de K optimal correspond au nombre vingt(20). Au-delà de ce nombre, le taux de confiance de la prédiction demeure statique avec toutefois une augmentation du temps d'exécution de l'algorithme.

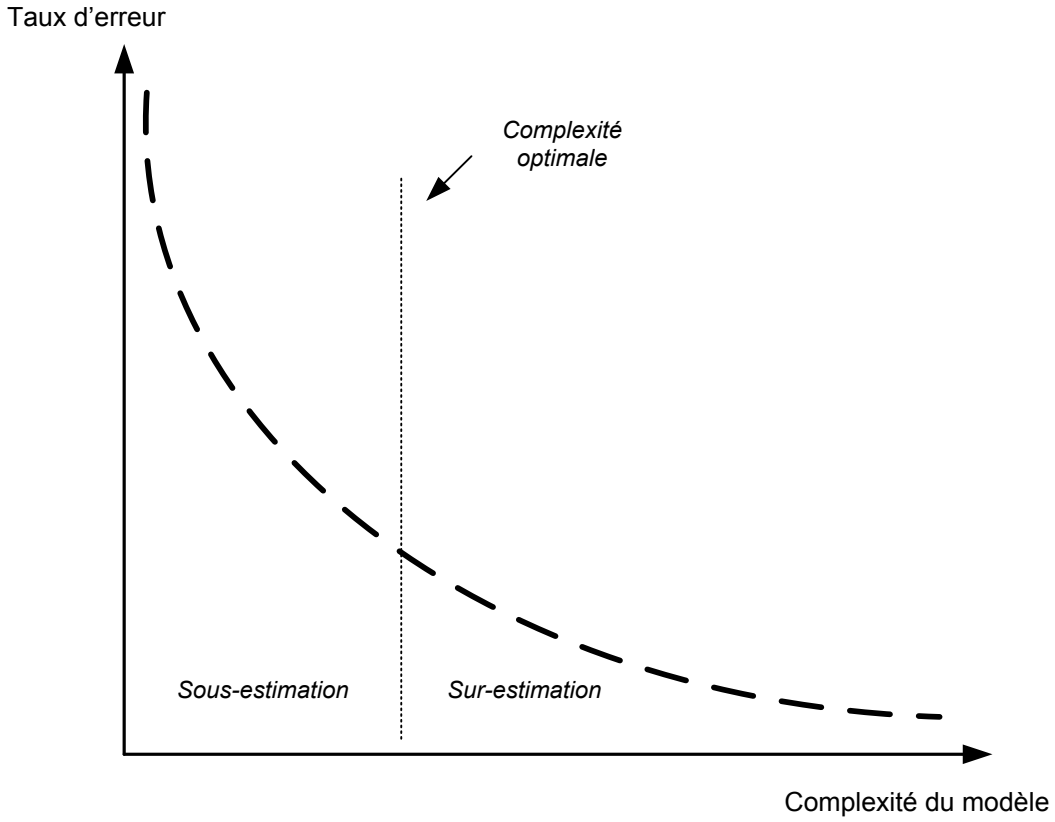


Figure 3.4-11 : compromis entre le taux d'erreur et la complexité dans le choix du K (KNN) – adapté de (Larose, 2005)

L'idée pour cette tâche de fouille de données est de vérifier si la prédiction basée sur la proximité spatiale (relation métrique) est conforme ou non avec les valeurs réelles notées au cours de l'année 2009.

Le workflow sous GeoKNIME ainsi que la configuration de la tâche de Géo-KNN sont donnés à la Figure 3.4-12.

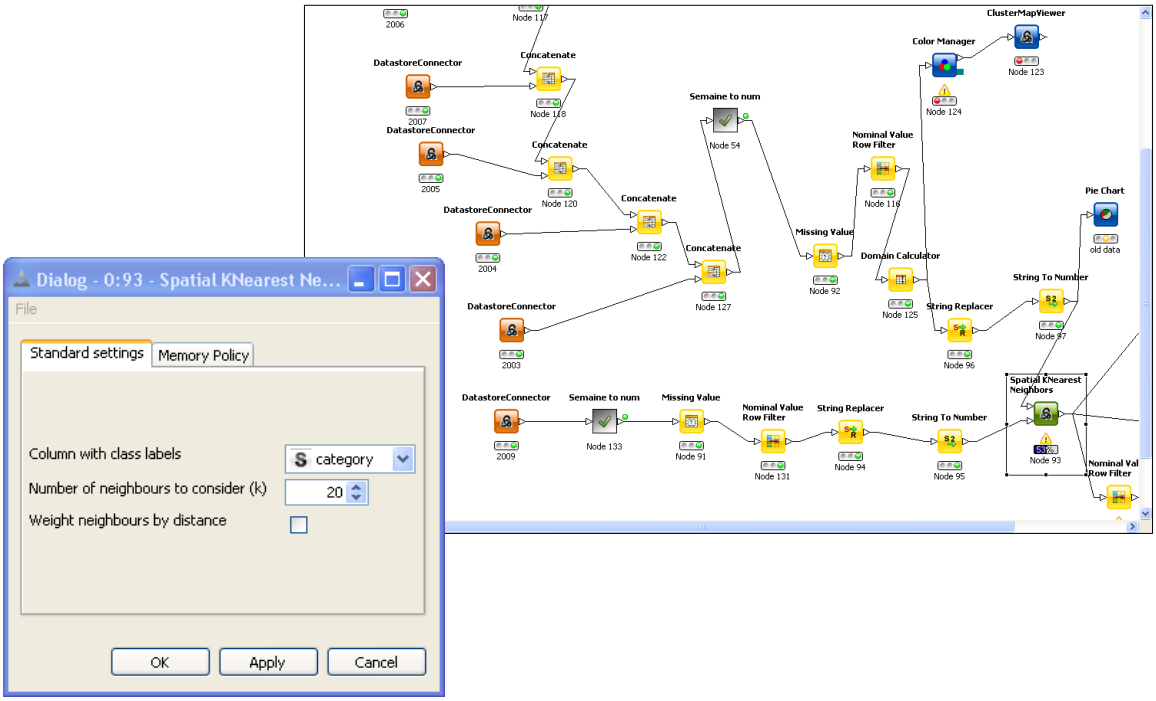


Figure 3.4-12: Configuration d'un Géo-KNN

Le temps mis par l'algorithme pour la prédiction d'environ trois(3) mn (2mn59s) pour 14570 crimes à prédire en se basant sur les données antérieurs (75955 crimes)

On note que l'algorithme prédit pour l'année 2009 une légère augmentation (7% de plus) des crimes liés aux drogues et narcotiques avec une nette diminution de la violence armée (cf. Figure 3.4-13 et Figure 3.4-14).

En effectuant une comparaison des valeurs réelles et celles prédites, on note 85%[36] de confiance dans la prédiction (cf. Figure 3.4-15). Cela témoigne de la justesse de prédiction de l'algorithme et que les résultats produits ne sont pas éloignés de la réalité.

[36] Fort certainement ce résultat pourrait être amélioré parce que la fouille de données ne produit pas un résultat unique. Pour peu que l'on prétraite autrement les données, on peut arriver à des résultats qui pourront être inférieur ou supérieur au taux de confiance actuel.

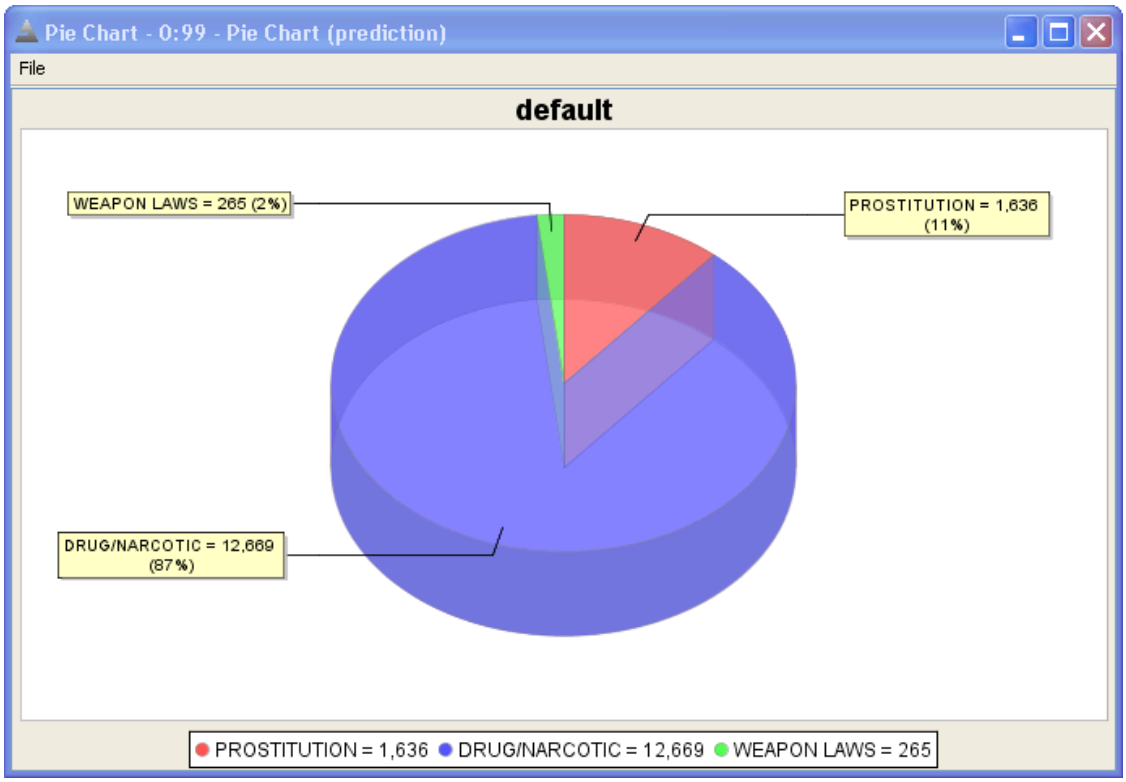


Figure 3.4-13: Prédictions basées sur le GeoKNN des crimes de 2009

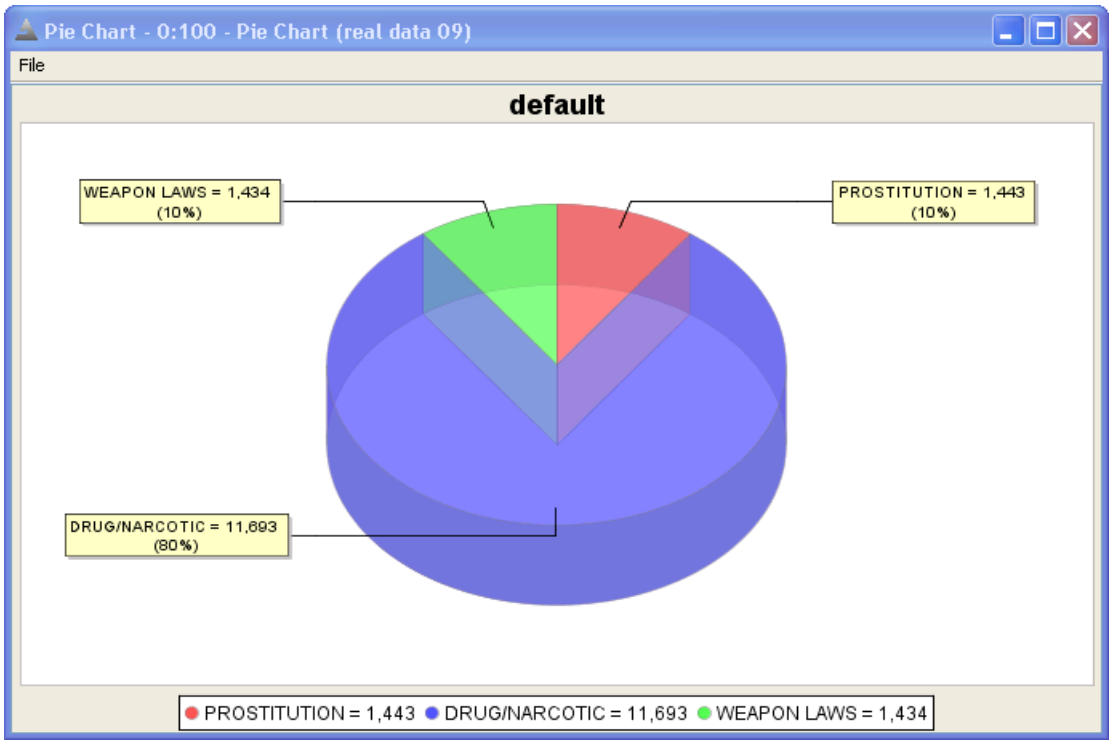


Figure 3.4-14: Pourcentage des crimes répertoriés pour l'année 2009

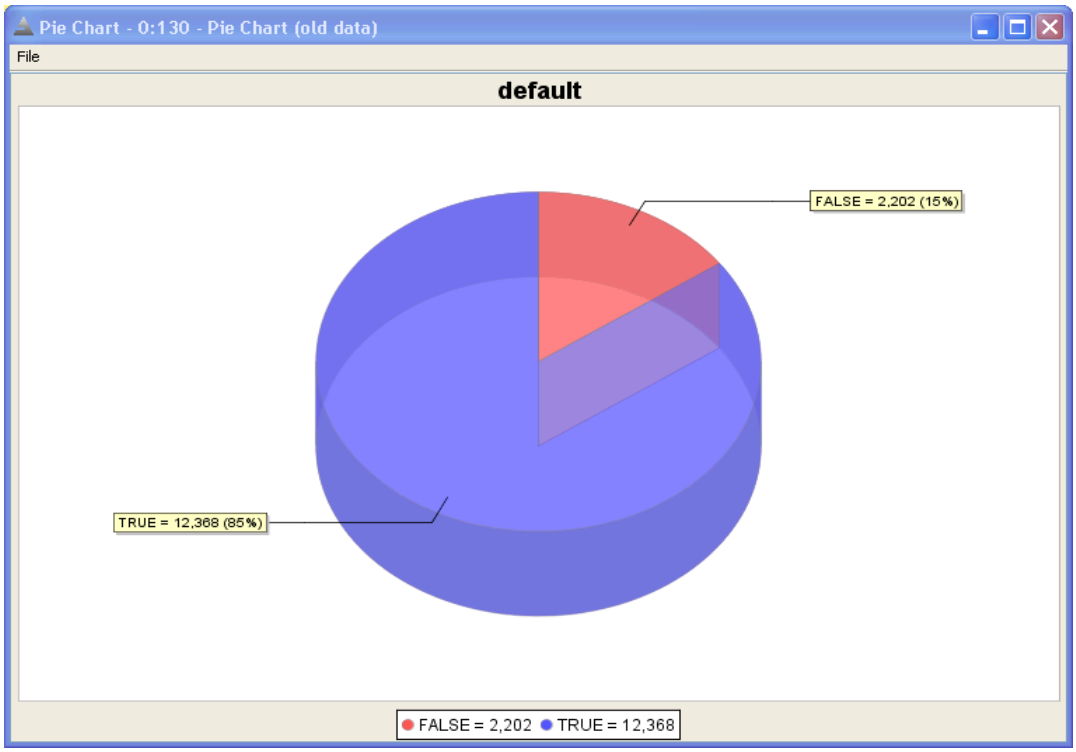


Figure 3.4-15:Taux de confiance de la prédiction des catégories de crime de San-Francisco pour l'année 2009

3.4.3. Arbre de décision spatial : construction d'un arbre de décision pour la prédiction des catégories de crimes de San-Francisco

Les données de criminalité sur la ville de San-Francisco ont été également mises à profit pour tester la prédiction basée sur les arbres de décision spatiaux. Pour rappel, l'objectif avec les arbres de décision est de générer un ensemble de règles qui puisse permettre de prédire la valeur d'une variable cible. Cet ensemble de règles est présenté sous forme de graphe ou d'arbre.

Pour la construction de l'arbre proprement dite, nous avons utilisé un sous ensemble des crimes répertoriés pour l'année 2010 notamment les crimes liés :

- au vandalisme
- au vol de voiture

- aux drogues et narcotiques

À cela s'ajoutent les appels au 911 n'ayant pas de rapport avec quelques crimes que ce soit, regroupés dans la catégorie NON-CRIMINEL.

La combinaison des différents nœuds de fonctionnalités ainsi que la configuration du nœud de génération de l'arbre de décision spatial sont données respectivement par les Figure 3.4-16 et Figure 3.4-17

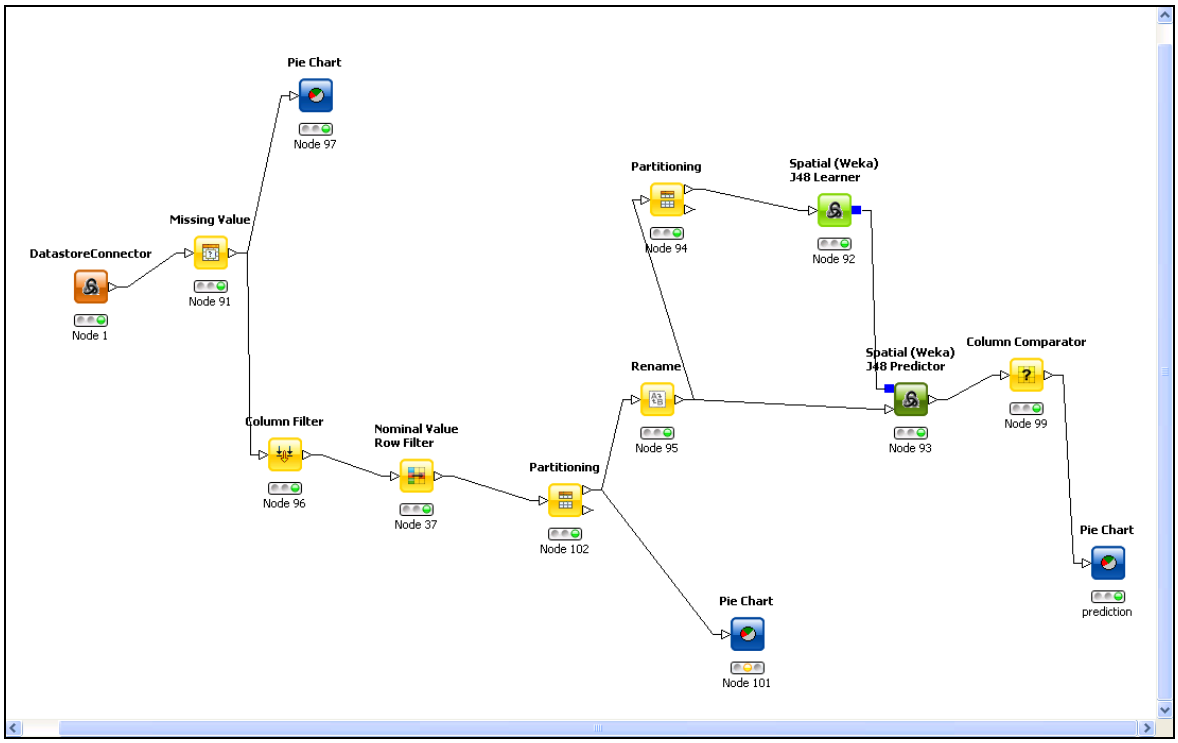


Figure 3.4-16 : Structure de workflow pour une fouille basée sur les arbres de décision spatiaux

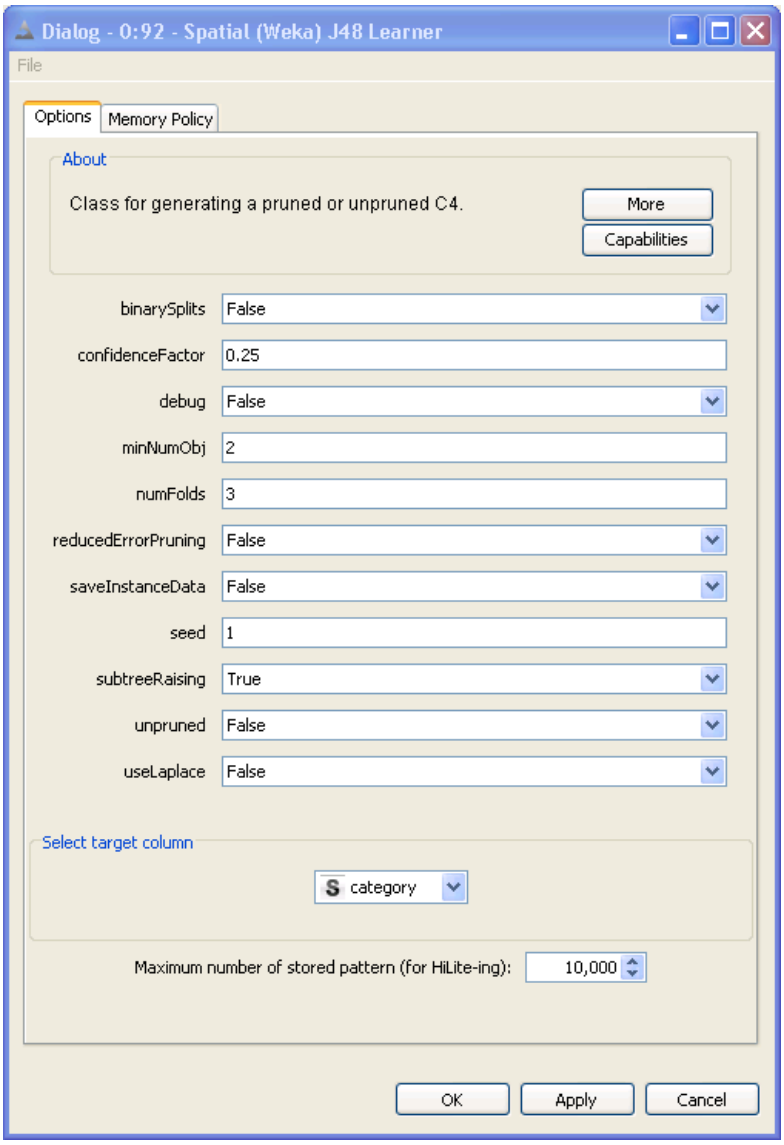


Figure 3.4-17: Configuration du nœud de génération de l'arbre de décision

Le temps mis pour la construction de l'arbre de décision spatiale est de moins d'une minute (1mn12s) et de 9mn42s pour la prédiction.

En analysant les valeurs de la prédiction vis-à-vis des valeurs réelles, on note une augmentation des crimes des catégories NON-CRIMINEL (33 à 43%). Dans le même temps, le pourcentage des crimes liés au vandalisme a fortement diminué (20 à 12%) (cf. Figure 3.4-18 et Figure 3.4-19).

Le taux de confiance dans la prédiction pour les arbres de décision est légèrement inférieur à celle basée sur le GeoKNN. En effet, en comparant les valeurs prédites à celles réelles, on note que 60% des valeurs prédites sont vraies (cf. Figure 3.4-20).

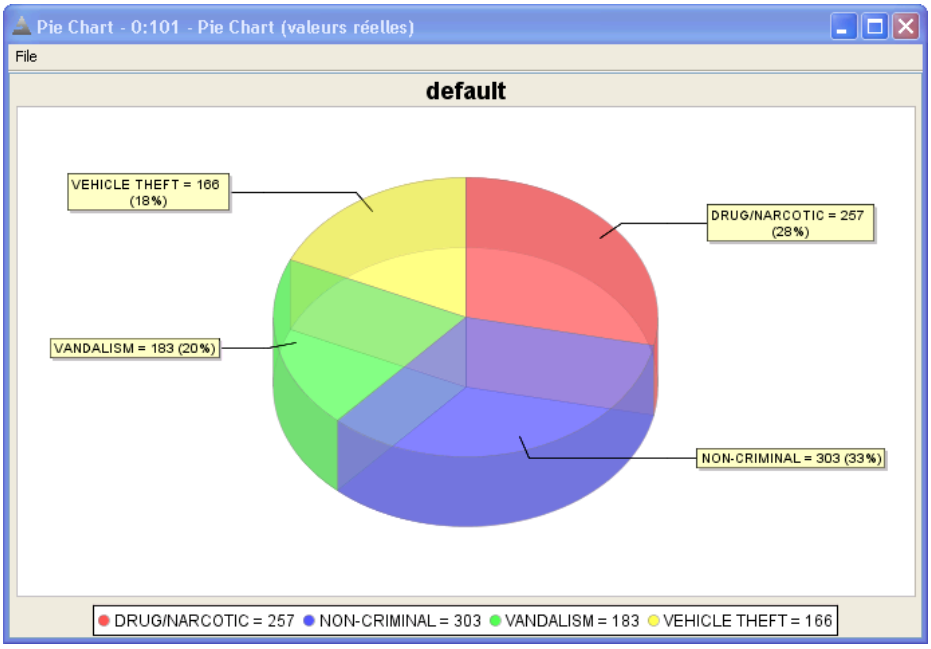


Figure 3.4-18: Pourcentage des crimes répertoriés pour 2010 par catégorie

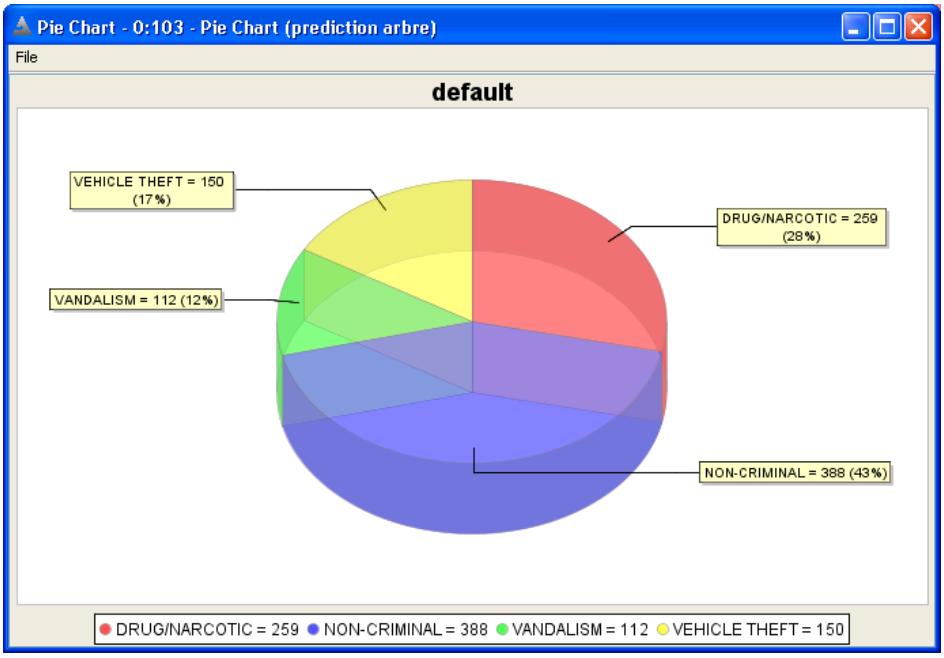


Figure 3.4-19: Pourcentage des crimes prédits pour 2010 par catégorie

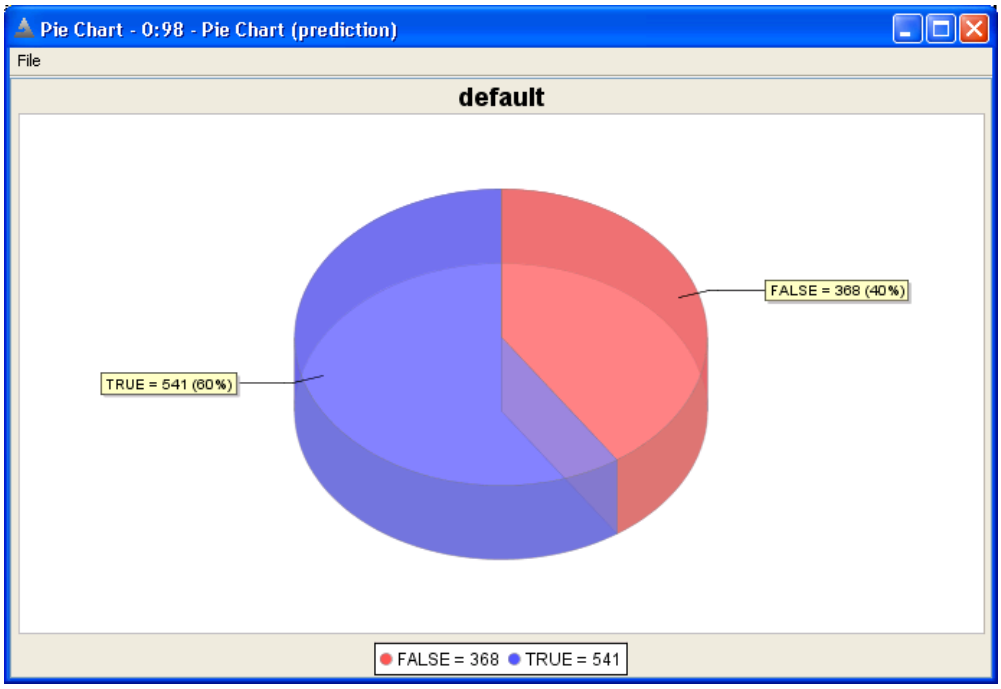


Figure 3.4-20: Taux de confiance de la prédiction basée sur les arbres de décision

3.5. Conclusion

Tout au long de ce chapitre, essentiellement technique, nous avons présenté un nouvel outil GeoKNIME issu de la « spatialisation » d'un outil de fouille de données « traditionnelle » afin que ce dernier puisse supporter la fouille sur des données géo-spatiales. Pour cela, il a fallu au préalable choisir un outil qui se prête aisément à cet exercice. C'est dans ce contexte que notre choix s'est porté sur KNIME ***qui de par sa modularité, son extensibilité et son support des différentes étapes du modèle CRISP-DM offre bien des avantages par rapport aux autres solutions***.

La démarche de « spatialisation » nécessitait au préalable la création d'un nouveau type de données en l'occurrence le type Geometry. Cette étape passée, divers algorithmes de fouille ont été enrichis spatialement. La bibliothèque KNIME s'est vue ainsi dotée d'algorithmes assurant des tâches de Géo-Classification, Géo-Clustering et de construction d'arbres de décision spatiaux.

En termes d'efficience, il est clair que ces algorithmes gagneraient à être améliorés au vu du temps de calcul relativement important entre géométries. Mais en aucun cas, on ne saurait remettre en cause leur utilité. En effet, à travers les différents tests effectués en se

basant sur les données de criminalité de la ville de San-Francisco, on a pu se rendre compte de la justesse des résultats issus de l'utilisation de ces algorithmes notamment au niveau du clustering, de la prédiction de valeurs basée sur les k plus proches voisins ou les arbres de décision.

Toutefois, nous ne comptons pas nous arrêter en si bon chemin en ce qui concerne la spatialisation d'autres algorithmes. À termes, d'autres algorithmes se verront également enrichis. Il s'agira essentiellement d'algorithmes de génération de règles d'association et de régression linéaire. Également d'autres fonctionnalités tournant autour de la fouille elle-même se verront implémentées.

L'utilité des algorithmes implémentés, ainsi que la « *relative simplicité* »[37] dans leur mise en œuvre témoignent de la validité et de l'aspect pratique de cette nouvelle approche intégrée de fouille de données spatiales. En effet, avec cette composante intégrée, des fonctionnalités supplémentaires peuvent aisément être mises en œuvre.

[37] La relative simplicité dans ce contexte signifie que la spatialisation d'algorithmes de fouille de données - en l'occurrence ceux auxquels nous avons touchés dans la réalisation du prototype – n'est pas d'une si grande complexité, donc peut se voir appliquée à tout autre algorithme implanté dans d'autres bibliothèques de fouille de données.

Chapitre 4 – Conclusions et perspectives

4.1. Conclusion

La « spatialisation », entendons par là l'utilisation de plus en plus accrue de données géo-spatiales, est un phénomène qui gagne en ampleur au sein des organisations. Cela combiné au potentiel de ces données en termes de connaissances a suscité le besoin d'apprendre d'elles. C'est de ce contexte qu'est née la fouille de données géo-spatiales dont la finalité est de servir de support au processus de décision en mettant à contribution les connaissances extraites des données géo-spatiales.

Dérivée de la fouille de données « traditionnelle », la fouille géo-spatiale s'est vite érigée en un domaine à part entière afin de mieux cerner la spécificité de l'information géo-spatiale. En effet, du fait des caractéristiques particulières de ce type d'information, la fouille « traditionnelle » a montré ces limites.

Dès lors, plusieurs approches de fouille géo-spatiales ont été mis en œuvre. Ces approches peuvent être caractérisées selon plusieurs points de vue dont celui des approches de prétraitement versus celui de traitement dynamique de la composante spatiale.

Pour rappel, les approches de prétraitement, outre le désavantage majeur qui consiste en l'extraction préalable des corrélations spatiales, prône la réutilisation des outils de fouille de données existants qui par ailleurs sont éprouvés. D'un autre côté, les approches de traitement dynamique de la composante spatiale traitent nativement des corrélations spatiales sans passer par une phase d'extraction. Mais ont comme inconvénient majeur la non prise en compte des attributs descriptifs dans la fouille, l'impossibilité de supporter toutes les étapes d'une fouille données.

Au vue des inconvénients respectifs de ces différentes approches, il a paru important de réfléchir à une nouvelle approche intégrée de fouille tirant parti des avantages des approches citées précédemment et surtout assurant le support de l'information spatiale à toutes les étapes du processus de fouille de données. Fort de cela, nous avons proposé et décrit une approche qui consiste à l'intégration de la composante géo-spatiale dans un outil de fouille de données qui prend en considération divers types de relations spatiales.

Différentes relations sont ainsi prises en compte dans notre approche. Ces relations vont des relations topologiques, à celles directionnelles en passant par les relations de similitude de formes. Aussi, l'innovation majeure au niveau de l'approche que nous proposons a été de prendre en compte la possibilité d'effectuer une fouille de données basée sur des relations qualitatives et quantitatives principalement pour les relations topologiques.

En termes d'avantages pour cette approche, on note qu'elle:

- Permet le support de la composante spatiale au niveau de toutes les phases du processus de fouille de données : principalement les phases de préparation des données, d'analyses exploratoire, d'évaluation et de mise en production.
- permet une réutilisation des outils de fouille de données existants : l'approche consiste à enrichir une bibliothèque de fouille existant. De ce point de vue, on tire avantage de la maturité des outils existants et de la diversité des algorithmes de fouille.
- traite dynamiquement les corrélations entre entités géo-spatiales : la composante géo-spatiale est intégrée comme un type à part entière. Aussi, on ne passe plus par une phase de prétraitement des relations spatiales mais les corrélations géo-spatiales sont extraites dynamiquement lors de l'opération de fouille de données. Afin de traiter ces corrélations spatiales, nous mettons à contribution différentes bibliothèques de traitement de données géo-spatiales existantes notamment JTS, GeoTools, GeOxygene.

Suivant les outils et les algorithmes de fouille, la démarche de « spatialisation » d'un outil à l'autre varie. D'où la nécessité d'avoir un cadre général qui décrit les différentes étapes nécessaire à la mise en œuvre de l'approche. C'est dans ce sens que nous avons proposé un cadriciel qui décrit comment procéder à l'intégration de la composante géo-spatiale et des différents éléments à avoir pour une intégration réussie.

En suivant le cadre général proposé, nous avons enrichi spatialement la bibliothèque open source de fouille de données KNIME. Cet enrichissement a d'abord consisté en l'intégration d'un nouveau type de données permettant ainsi le support transparent, et cohérent de la composante géométrique des objets spatiaux. Le support des géométries en

tant que type offre divers avantages dont principalement la banalisation de l'information géo-spatiale qui est alors vue comme un type à part entière qui à l'instar des autres disposent de ses propres opérations de manipulation. Par extension, la banalisation de l'information géo-spatiale facilite du même coup la fouille de données sur ce type de données.

Par la suite, différents algorithmes ont été enrichis spatialement. GEOKNIME – le nouvel outil issu de la spatialisation de KNIME - supporte actuellement des fonctionnalités de prédiction basée sur la Géo-Classification et les arbres de décision spatiales ainsi que celles de Géo-Clustering.

À travers les tests effectués sur les données de criminalité de la ville de San-Francisco, on a pu se rendre compte de l'efficience de l'approche ainsi que de la robustesse des algorithmes enrichis spatialement. Cependant, pour s'assurer de la validité et de l'approche et des algorithmes spatialement enrichi, d'autres tests devront venir renforcer la justesse de l'approche proposée.

Bien de choses restent à faire en ce qui concerne l'optimisation de ces algorithmes. Il est important de noter que comparativement à la fouille « traditionnelle », celle géo-spatiale exigera toujours plus de temps et de ressources au regard de la complexité même des objets spatiaux et des relations diverses qu'ils peuvent entretenir mutuellement. L'idée, comme nous le verrons dans la section suivante est de mettre en œuvre des méthodes qui permettront la réduction du temps de calcul.

4.2. Perspectives

Beaucoup de choses sont et restent à faire en ce qui concerne l'approche que nous avons proposée et de façon plus générale la fouille de données géo-spatiales. À court terme, bien des fonctionnalités sont à implémenter au sein de la bibliothèque open source que nous avons spatialement enrichie. Au nombre de ces fonctionnalités, on note :

- **La lecture de données géo-spatiales**

Il s'agit d'ajouter des fonctionnalités de lecture de données depuis d'autres sources de données autre que PostGIS. Il s'agit plus précisément des fonctionnalités de lecture de

fichiers de différents formats Shapefile, GML, etc. ainsi que la connexion à des bases de données géo-spatiales telles Oracle Spatial.

- **La réduction de données**

Il s'agit de mettre en œuvre des fonctionnalités qui permettront d'effectuer la fouille sur un sous ensemble de données en lieu et place de l'ensemble des données. Cela permettra de réduire sensiblement le temps que requiert actuellement la fouille de données géo-spatiales. La réduction du temps de calcul pourrait également se faire en implémentant une solution autre que spatiale. Il s'agit en effet des mettre en œuvre des fonctionnalités d'exécution de tâches de façon distribuée[38]; i.e. avoir la possibilité d'exécuter différentes tâches d'une fouille de données sur des machines différentes.

- **L'implémentation d'autres algorithmes de fouille**

Avec la variété d'algorithmes de fouille de données, il est impossible le temps d'un mémoire de les prendre tous en compte. Afin de compléter les algorithmes déjà enrichis, il serait intéressant d'implanter des algorithmes de génération de règles d'association spatiales et de régression linéaire spatiale.

De façon générale, la fouille de données géo-spatiales est un domaine passionnant où nombre de défis restent encore à relever. En termes de perspectives à long terme, les points suivants pourraient faire l'objet d'étude :

- **Couplage Fouille de données géo-spatiales/SOLAP**

Il serait intéressant de combiner notre approche avec l'analyse géo-spatiale notamment le SOLAP. Cette combinaison ne peut être qu'au bénéfice de l'utilisateur final dans la mesure où ces deux(2) branches sont complémentaires et concourent toutes au même but : le support de la prise de décision dans le monde de l'entreprise. Notons qu'un rapprochement OLAP-Datamining a déjà fait l'objet d'intenses recherches *(Sarawagi, et al., 1998) (Palpanas, et al., 2005) (Messaoud, 2006) (Ramakrishnan, et al., 2007)*. L'idée est de

[38] La fonctionnalité d'exécution parallèle est déjà implémentée dans la version commerciale de KNIME

voir dans quelle mesure on pourrait adapter les solutions existantes au couple Datamining spatial/SOLAP.

Réussir un tel couplage pourrait un temps soit peu aider à l'optimisation du temps consacré à la fouille de données en permettant de se focaliser sur un sous ensemble intéressant de données. D'un autre point de vue, ce couplage pourrait aider à la construction de cubes de données intéressants dans la mesure où grâce à la fouille de données principalement aux règles d'association voir quels sont les attributs qui vont ensemble.

Il est évident que l'implémentation d'un tel couplage ne sera pas sans difficultés et devra être effectuée sous le respect de certaines contraintes. En effet, contrairement à la fouille sur des données provenant de source autre que des cubes de données, la fouille basée sur OLAP devrait tenir compte du contexte. À un instant *t* donné, l'utilisateur d'un cube visualise des faits avec différents niveau de granularité. L'idée est de se servir de cette « photographie » instantanée des données pour effectuer la fouille de données.

- **GPMML ou Géo-Spatial PMML**

Pour rappel, le PMML est un langage basé sur XML qui permet d'assurer l'interopérabilité entre les différents outils de fouille de données[39]. Comme toujours, le KDD a une avance sur le GKD dans la mesure où même basique, cette fonctionnalité existe au niveau de certains outils de fouille. Il serait intéressant d'assurer une telle fonctionnalité pour la fouille de données géo-spatiales. En sommes, il s'agira de créer un nouveau format d'échange G/PMML basée sur PMML qui décrira sous forme XML les résultats obtenus d'une fouille de données. Avec ce nouveau format d'échange supportant les données géo-spatiales, tout outil de fouille de données géo-spatiales pourra utiliser les résultats provenant d'un autre outil.

- **Géo-Datamining visuel**

La visualisation cartographique est un élément essentiel en matière de fouille de données géo-spatiales dans la mesure où elle permet de voir sur une carte les résultats de la

[39] Cf. http://www.wikipedia.org

fouille. On pourrait pousser plus loin cette idée en permettant par exemple d'effectuer une fouille visuelle.

L'idée du Géo-Datamining visuel est de concevoir un outil du type SOLAP dans lequel le visuel prend toute sa place. L'utilisateur pourrait choisir depuis une carte, les entités géo-spatiales qu'il désire voir impliquées dans une fouille et grâce à des opérateurs appropriés ajuster certains paramètres et exécuter la fouille. Le Géo-Datamining visuel devrait à termes fournir des outils hautement interactifs et intuitifs de façon à permettre aux utilisateurs non-experts de s'affranchir de la complexité de la fouille de données.

Il est évident que l'atteinte de cet objectif exige beaucoup d'efforts en termes de R&D dans la mesure où beaucoup de choses restent à faire en ce qui concerne la fouille de données géo-spatiales.

Bibliographie

A Reuse-based Spatial Data Preparation Framework for Data Mining. **Bogorny, Vania, Martins, Engel, Paulo and Alvares, Luis Otavio. 2005.** 2005.

A survey of data mining and knowledge discovery software tools. **Goebel, Michael and Le, Gruenwald. 1999.** 1999.

ADAM. 2009. ADAM. [Online] Septembre 2009. http://datamining.itsc.uah.edu/adam/index.html.

ADaM: a data miningtoolkit for scientists and engineers. **Rushing, John, et al. 2004.** 2004.

Adoption of open source software: Is it the matter of quality . **Kamseu, Flora and Habra, Naji. 2004.** 2004.

Agrawal, Rakesh and Ramakrishnan, Srikant. 1994. Fast algorithms for mining association rules. [ed.] Morgan Kaufmann. *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases,.* September 12-15, 1994, pp. 487-499.

Agrawal, Rakesh, Imielinski, Tomasz and Swami, Arun. 1993. Database Mining: A Performance Perspective. *IEEE Transactions on Knowledge and Data Engineering.* 1993.

—. 1993. Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD international conference on Management of data.* 1993, pp. 207--216.

Agrawal, Rakesh, Johannes Gehrke, Dimitrios Gunopulos and Prahhakar Raghavan. 1998. Automatic Subspace Clustering of High Dimensional Data for Data Mining Application. *Proceedings of the 1998 ACM-SIGMOD International Conference on Management of Data.* Juin 1998.

Alsabti, Khaled, Ranka, Sanjay and Singh, Vineet. 1998. An Efficient K-Means Clustering Algorithm. 1998.

Amber, Scott. W. 2005. [Online] 2005. http://www.ambysoft.com/unifiedprocess/agileUP.html.

Appice, A., Ceci, M. and Malerba, D. 2000. KDB2000: An integrated knowledge discovery tool. *Management Information Systems.* 2000, Vol. 6, pp. 531-540.

Appice, Annalisa, et al. 2003. Discovery of spatial association rules in geo-referenced census data: A relational mining approach. *Intelligent Data Analysis 7.* 2003, pp. 541–566.

Bel Hadj Ali, Atef. 2001. *Qualité géométrique des entités géographies surfaciques. Application à l'appariement et définition d'une typologie des écarts géométriques (Thèse).* 2001.

Bel Hadj ali, Atel. 2001. Mesures entre objets surfaciques. Application à la qualification des liens d'appariement. *Bulletin d'Information Scientifique et Technique de l'IGN,.* 2001, 71, pp. 33-54 .

—. 2001. Mesures entre objets surfaciques. Application à la qualification des liens d'appariement. 2001.

Berthold, Michael R., et al. 2006. Knime: The Konstanz Information Miner. 2006.

Bogorny, V. and Engel, P. M. and Alvares, L.O. 2005. Towards the Reduction of Spatial Joins for Knowledge Discovery in Geographic Databases using Geo-Ontologies and Spatial Integrity Constraints. *In 2nd KDO ECML/PKDD Workshop.* 2005, pp. 51-58.

Bogorny, Vania, Engel, Paulo Martins and Luis O. Alvares. 2007. Enhancing Spatial Association Rule Mining in Geographic Databases. [ed.] Hector Oscar Nigro and Sandra Gonzalez Cizaro and Daniel Xodo. *Data Mining with Ontologies: Implementations, Findings and Frameworks.* 2007, pp. 160-181.

Bogorny, Vania, et al. 2006. Weka-GDPM: Integrating Classical Data Mining Toolkit to Geographic Information Systems. *In SBBD Workshop on Data Mining Algorithms and Aplications(WAAMD'06).* 2006, pp. 9-16.

Bogorny, Vania, Martins, Engel, Paulo and Alvares, Luis Otavio. 2005. A Reuse-based Spatial Data Preparation Framework for Data Mining. *In Proceedings of the 17th International Conference on Software Engineering and Knowledge Engineering.* 2005, pp. 649-652. In Proceedings of the 17th International Conference on Software Engineering and Knowledge Engineering.

—. 2005. A Reuse-based Spatial Data Preparation Framework for Data Mining. 2005.

Brachman, Ronald J. and Anand, Tej. 1994. The Process of Knowledge Discovery in Databases: A first sketch. *AAAI-94 Workshop.* 1994.

Brachman, Ronald, et al. 1993. Integrated Support For Data Archaeology. *International Journal of Intelligent and Cooperative Information Systems.* 1993.

Buchin, Kevin, Buchin, Maike and Wenk, Carola. 2006. Computing the Fréchet Distance between Simple Polygons in Polynomial Time. *Proceedings of the twenty-second annual symposium on Computational geometry.* 2006, pp. 80 - 87 .

—. 2006. Computing the Fréchet Distance between Simple Polygons in Polynomial Time. 2006.

Buttenfield, Barbara, et al. 2004. Geospatial Data Mining and Knowledge Discovery. *A research agenda for geographic information science.* 2004.

Chapman, Pete, et al. 1999. *CRISP-DM 1.0: Step-by-step data mining guide.* 1999.

Chawla, sanjay, shashi shekhar, weili, wu and uygar, ozesmi. 2001. Modeling spatial dependencies for mining geospatial data: an introduction. *Geographic data mining and Knowledge Discovery.* 2001.

—. 2001. Modeling spatial dependencies for mining geospatial data: an introduction. 2001.

Chelghoum, Nadjim, Zeitouni, Karine and Boulmakoul, Azedine. 2002. Fouille de données spatiales par arbre de décision multi-thèmes. *2èmes Journées sur l'Extraction et la Gestion des Connaissances.* 2002, Janvier, pp. 281-286.

Chen, Ming-syan, et al. 1996. Data Mining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering.* 1996, Vol. 8, pp. 866-883.

Chen, Xiaojun, et al. 2007. A Survey of Open Source Data Mining Systems. 2007.

Computing the Fréchet Distance between Simple Polygons in Polynomial Time. **Buchin, Kevin, Buchin, Maike and Wenk, Carola. 2006.** 2006.

Data Mining Desktop Survival Guide. **Williams, Graham. 2009.** 2009.

Devogele, Thomas. 1997. *Processus d'intégration et d'appariement de bases de donnése géographiques – Application à une base de données routières multiéchelles (Thèse).* 1997.

—. 1997. Processus d'intégration et d'appariement de bases de donnése géographiques – Application à une base de données routières multiéchelles (Thèse). 1997.

Discovering Geographic Knowledge: The INGENS System. **Malerba, Donato, et al. 2000.** 2000.

Discrete Fréchet Distance. **Eiter, Thomas and Mannila, Heikki. 1994.** 1994.

Egenhofer, Max. J and Shariff, A. Rashid B. M. 1998. Metric Details for Natural-Language Spatial Relations. *ACM Transactions on Information Systems.* 1998, Vol. 16, pp. 295--321.

Egenhofer, Max. J. 1998. Metric Details for Natural-Language Spatial Relations. 1998.

Egenhofer, Max.J. and Herring, John. 1991. Categorizing Binary Topological Relationships Between Regions, Lines, and Points in Geographic Databases . 1991.

Eiter, Thomas and Mannila, Heikki. 1994. Computing Discrete Fréchet Distance. 1994.

—. 1994. Discrete Fréchet Distance. 1994.

Empirical study of the effects of open source adoption on software development economics . **Ajila, Samuel A. , Di, Wu. 2007.** 2007.

Ester, Martin, et al. 1998. *Incremental Clustering for Mining in a Data Warehousing Environment.* [ed.] Morgan Kaufmann Publishers Inc. San Francisco, CA, USA : s.n., 1998. pp. 323 - 333. Proceedings of the 24rd International Conference on Very Large Data Bases. ISBN:1-55860-566-5.

Ester, Martin, et al. 1999. Spatial Data Mining: Database Primitives, Algorithms and Efficient DBMS Support. *In Proc.ofInt.Conf.on Databases in Office, Engineering and Science.* 1999, Vol. 4, pp. 193--216.

Ester, Martin, Kriegel, Hans-Peter and Sander, Jörg. 2001. Algorithms and applications for spatial data mining. *Geographic Data Mining and Knowledge Discovery.* 2001.

—. 1999. Knowledge Discovery in Spatial Databases. *23rd German Conf. on Artificial Intelligence.* 1999.

Ester, Martin, Kriegel, Hans-Peter and Sander, Jtirg. 1997. Spatial Data Mining A Database Approach. *Advances in Spatial Databases.* 1997, Vol. 1262, pp. 47-66.

—. 1997. Spatial Data Mining: A Database Approach. 1997.

Ester, Martin, Kriegel, Hans-Peter and Xu, Xiaowei. 1995. A Database Interface for Clustering in Large Spatial Databases. *Proceedings of 1st International Conference on Knowledge Discovery and Data Mining.* 1995.

Faber, Vance. 1994. Clustering and the Continuous k-Means Algorithm. *Los Alamos Science.* 1994, Vol. 22, pp. 138--144.

Fayyad, Usama. 1998. Data Mining and Knowledge Discovery. [ed.] Kluwer Academic Publishers. Janvier 1998, Vol. Volume 2 , Issue 1, pp. Pages: 5 - 7 .

Fayyad, Usama M. 1996. Data Mining and Knowledge Discovery: Making Sense Out of Data. IEEE Intelligent Systems, Octobre 1996, Vols. vol. 11,, no. 5, pp. pp. 20-25.

Fayyad, Usama, Gregory, Piatetsky-Shapiro and Padhraic, Smyth. 1996. From Data Mining to Knowledge Discovery in Databases. 1996.

Fayyad, Usama, Piatetsky-Shapiro, Gregory and Smyth, Padhraic. 1996. The KDD process for extracting useful knowledge from volumes of data. *ACM New York, NY, USA* . 1996, Vol. Volume 39, 11.

Frank, Andrew U. 1996. Qualitative Spatial Reasoning: Cardinal Directions as an Example. 1996.

Frawley, William J., Piatetsky-Shapiro, Gregory and Matheus, Christopher J. 1992. Knowledge Discovery in Databases: An Overview. *AI Magazine* . 1992, Vol. Volume 13, 3.

Friedman, Jerôme. 1997. Data mining and Statistics: What's the connection? 1997.

From Experimental Machine Learning to Interactive Data Mining. **Demsar, JANEZ and Zupan, Blaz. 2009.** 2009.

Gardarin, Georges. 2006. Cours Clustering. *Georges Gardarin.* [Online] 2006. http://georges.gardarin.free.fr/Cours_Total/DM3-Clustering.ppt.

Geographic data mining and knowledge discovery: an overview . **Miller, Harvey. J and Jiawei, Han. 2001.** 2001.

Geominer: A system prototype for spatial Data Mining: . **Han, Jiawei, Koperski, Krzysztof and Stefanovic, Nebojsa. 1997.** 1997.

Giudici, Paolo and Figini, Silvia. 2009. *Applied datamining for business and industry.* [ed.] NJ : John Wiley Hoboken. 2009. 9780470058862 (cloth).

Goyal, Roop. K et Egenhofer, Max. J. 2001. Similarity of cardinal directions. [éd.] Springer-Verlag. *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases.* 2001, Vol. 2121, pp. 36-58.

—. 2001. Similarity of cardinal directions. 2001.

Goyal, Roop. K. 2000. Similarity assessment for cardinal directions between extended spatial objects. 2000.

—. 2000. *Similarity assessment for cardinal directions between extended spatial objects (Thèse).* 2000.

Guha, Sudipto, Rajeev, Rastogib and Kyuseok, Shim. 1998. CURE: an efficient clustering algorithm for large databases. *In Proceedings of ACM SIGMOD International Conference on Management of Data.* 1998, pp. 73-84.

Hamerly, Greg and Elkan, Charles. 2003. Learning the k in k-means. [ed.] MIT Press. *Neural Information Processing Systems.* 2003.

Han, Jiawei. 1997. OLAP Mining: An Integration of OLAP with Data Mining. *In Proceedings of the 7th IFIP 2.6 Working* . 1997.

Han, Jiawei, Koperski, Krzysztof and Stefanovic, Nebojsa. 1997. Geominer: A system prototype for spatial Data Mining. [ed.] ACM. *ACM SIGMOD Record.* Juin 1997, Vols. 26 , Issue 2, 2, pp. 553 - 556 .

—. 1997. Geominer: A system prototype for spatial Data Mining:. 1997.

Han, Jiawei, Miceline, Kamber and Anthony, K.H. Tung. 2001. Spatial clustering methods in data mining: a survey. [ed.] Taylor and Francis. *Geographic Data Mining and Knowledge Discovery.* 2001, pp. 1–29.

Han, YC Jiawei and Cerconet, N. 1992. Knowledge Discovery in Databases: An Attribute-oriented approach. *Proceedings of the 18th VLDB Conference* . 1992.

Hand, David J., Mannila, Heikki and Smyth, Padhraic. 2001. *Principles of Datamining.* 2001.

Hernandez, Daniel, Clementini, Eliseo and Di Felice, Paolino. 1995. Qualitative distance. 1995.

—. 1995. Qualitative distance. 1995.

Hornick, Mark F., Marcade, Erik and Sunil, Venkayala. 2007. *Java data mining : strategy, standard, and practice : a practical guide for architecture, design, and implementation.* s.l. : Morgan Kaufmann, 2007. isbn13: 9780123704528.

Introduction to Xelopes version 4.0. **PrudSys AG. 2009.** 2009.

Jambu, Michel. 1999. *Introduction au data mining – analyse intelligente des données.* s.l. : Eyrolles, 1999.

Kantardzic, Mehmed. 2003. *Data mining : concepts, models, methods and algorithms.* s.l. : Wiley-IEEE Press, 2003.

Kanungo, Tapas, et al. 2000. The Analysis of a Simple k-Means Clustering Algorithm . 2000.

Karypis, George, Eui-Hong Han, and Vipin Kumar. 1999. Hierarchical Clustering Algorithm Using Dynamic Modeling. *IEEE Computer.* Août 1999, Vol. 8, 32, pp. 68-75.

Klösgen, Willi and MayFraunhofer, Michael. 2002. Spatial Subgroup Mining Integrated in an Object-Relational Spatial Database. [ed.] Springer. *Lecture notes in computer science.* 2002. PKDD 2002 : principles of data mining and knowledge discovery.

—. 2002. Spatial Subgroup Mining Integrated in an Object-Relational Spatial Database. 2002.

Klösgen, Willi and Zytkow, Jan M. 2002. Knowledge discovery in Databases: The purpose, necessity, and challenges. *Handbook of Data Mining and Knowledge Discovery.* Oxford University Press, Inc, 2002.

KNIME. 2009. [Online] 2009. http://www.knime.org/.

Kolatch, Erica. 2001. Clustering Algorithms for Spatial Databases: A Survey. 2001.

Koperski, Krzysztof. 1999. A progressive refinement approch to spatial data mining. *Ph,D Thesis.* 1999.

Koperski, Krzysztof, Adhikary, junas and Han, Jiawei. 1996. Progress and Challenges Survey Paper. SIGMOD Workshop on Research Issues on data Mining and Knowledge Discovery, 1996, pp. 1--10.

—. 1996. Progress and Challenges Survey Paper. 1996.

Koperski, Krzysztof, Han, Jiawei and Adhikary, Junas. 1998. Mining knowledge in geographical data. *Communications of the ACM.* 1998, Vol. 26.

Kurgan, Lukasz A. and Musilek, Petr. 2006. A survey of Knowledge Discovery and Data Mining process models. The Knowledge Engineering Review, 2006, Vol. Vol. 21, 1, pp. 1–24.

La gestion de projets: concepts, problématiques, méthodes et exercices. **D'Avignon, Gilles R. 2008.** 2008.

Larose, Daniel T. 2005. *Discovering Knowledge in Data: An introduction to data mining.* Hoboken, New Jersey : John Wiley & Sons, Inc.,, 2005.

Likas, Aristidis, Vlassis, Nikos and Verbeek, Jakob J. 2001. The global k-means clustering algorithm. 2001.

2008. Linux Online. [Online] 2008. http://www.linux.org/apps/AppId_8521.html.

Lu, Wei, Han, Jiawei and Ooi, Beng Chin. 1993. Discovery of General Knowledge in Large Spatial Databases. *Proc. Far East Workshop on Geographic Information Systems* . 1993.

Malerba, Donato, Annalisa, Appice and Vacca, Nicola. 2002. SDMOQL: An OQL-based Data Mining Query Language for Map Interpretation Tasks. *Proceedings of the EDBT 2002 Workshop on Database Technologies for Data Mining.* 2002, pp. 3--18.

—. 2002. SDMOQL: An OQL-based Data Mining Query Language for Map Interpretation Tasks. 2002.

Malerba, Donato, et al. 2000. Discovering Geographic Knowledge: The INGENS System. *Foundations of intelligent systems.* 2000. in 12th international symposium, ISMIS 2000.

—. 2000. Discovering Geographic Knowledge: The INGENS System. 2000.

Matteo, Matteucci. 2008. A Tutorial on Clustering Algorithms K-Means Clustering. [Online] 2008. http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html .

May, Michael and Savinov, Alexandr. 2003. SPIN!—An Enterprise Architecture for Spatial Data Mining. [ed.] Springer Berlin / Heidelberg. *Knowledge-Based Intelligent Information and Engineering Systems.* 2003, Vol. 2773, pp. 510-517.

—. 2003. SPIN!—An Enterprise Architecture for Spatial Data Mining. 2003.

Messaoud, Riadh BEN. 2006. *Couplage de l'analyse en ligne et de la fouille de donnees pour l'exploration, l'agregation et l'explication des donnees complexes.* 2006.

Messaoud, Riadh, et al. 2008. OLEMAR: An online environment for mining association rules in multidimensional data. [ed.] David Taniar. *Data mining and Knowledge discovery technologies.* 2008, pp. 1-35.

Mesures entre objets surfaciques. Application à la qualification des liens d'appariement. **Bel Hadj ali, Atel. 2001.** 2001.

Metric Details for Natural-Language Spatial Relations. **Egenhofer, Max. J. 1998.** 1998.

Metric details of topological line–line relations. **Nedas, K. A., Egenhofer, Max J and Wilmsen, D. 2005.** 2005.

Mierswa, Ingo, et al. 2006. Proceedings of the 12th ACM SIGKDD International PONZETTO & STRUBE Conference on Knowledge Discovery and Data Mining. [ed.] ACM Press. *YALE: Rapid Prototyping for Complex Data Mining Tasks.* 2006, pp. 935--940.

Miller, Harvey J. 2004. Tobler's First Law and Spatial Analysis. *Annals of the Association of American Geographers.* Juin 2004, Vol. 94, 2, pp. 284 - 289.

Miller, Harvey. J and Jiawei, Han. 2001. *Geographic data mining and knowledge discovery: an overview.* New York : Taylor & Francis, 2001., 2001. 0415233690.

—. 2001. Geographic data mining and knowledge discovery: an overview. 2001.

Mirkin, Boris. 2005. *Clustering for data mining: a data recovery approach.* 2005.

MLC++: a machine learning library in C++. **Kovahi, Ron, et al. 1994.** 1994.

Modèle statistiques des imprécisions géométriques des objets géographiques linéaires. **Vauglin, François. 1997.** 1997.

Modeling spatial dependencies for mining geospatial data: an introduction . **Chawla, sanjay, shashi shekhar, weili, wu and uygar, ozesmi. 2001.** 2001.

Mosig, Axel and Clausen, Michael. 2005. Approximately matching polygonal curves with respect to the Fréchet distance. *on the 19th European workshop on computational geometry - EuroCG 03.* 2005, pp. 113-127.

Natural Language Spatial Relations Between linear and areal objects: the topologie and metric of English language terms. **Shariff, A. Rashid. B.M and Egenhofer, Max. J. 1998.** 1998.

Nedas, K. A., Egenhofer, Max J and Wilmsen, D. 2007. Metric details of topological line–line relations. *International Journal of Geographical Information Science.* 2007, Vol. 21, 1, pp. 21 - 48 .

—. 2005. Metric details of topological line–line relations. 2005.

Ng, Raymond T. and Jiawei, Han. 2002. CLARANS: A Method for Clustering Objects for Spatial Data Mining. *IEEE Transactions on Knowledge and Data Engineering.* IEEE Educational Activities Department, Septembre 2002, Vol. 14, 5, pp. 1003 - 1016 .

Open Source Software Adoption: A Status Report. **Huaiqing, Wang and Wang, Chen. 2001.** 2001.

Openshaw, Stan. 1999. *Geographical data mining: key design issues.* 1999.

Openshaw, Stan, et al. 1987. *A Mark 1 Geographical Analysis Machine for the automated analysis of point data sets.* 1987. pp. 335 — 358.

2009. Orange. [Online] 2009. http://www.ailab.si/orange/.

Palpanas, Themis, Koudas, Nick and Mendelzon, Alberto. 2005 . Using Datacube Aggregates for Approximate Querying and Deviation Detection. *IEEE Transactions on Knowledge and Data Engineering.* 2005 , pp. 1465 - 1477.

Perens, Bruce. 1999. the open source definition in open source: voice of the open source revolution. 1999.

Piatestsky-Shapiro, Gregory. 2002. Data Mining coming of Age. *Handbook of Data Mining and Knowledge Discovery.* Oxford University Press, 2002.

Piatetsky-Shapiro, Gregory and Matheus, Christopher J. 1992. Knowledge Discovery Workbench for Exploring Business Databases. [ed.] Inc John Wiley & Sons. *INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS.* 1992, Vol. VOL. 7, pp. 675-686 .

Popelinsky, Lubos. 1998. Knowledge discovery in spatial data by. *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery.* 1998, Vol. 1510, pp. 185 - 193.

Processus d'intégration et d'appariement de bases de donnése géographiques – Application à une base de données routières multiéchelles (Thèse). **Devogele, Thomas. 1997.** 1997.

Progress and Challenges Survey Paper. **Koperski, Krzysztof, Adhikary, junas and Han, Jiawei. 1996.** 1996.

Qualitative distance. **Hernandez, Daniel, Clementini, Eliseo and Di Felice, Paolino. 1995.** 1995.

Ramakrishnan, Raghu and Chen, Bee-Chung. 2007. Exploratory mining in cube space. [ed.] Kluwer Academic Publisher. Août 2007, Vol. 15, 1, pp. 29 - 54.

Reinartz, Thomas. 2002. Stages of Discovery Process. *Handbook of Data Mining and Knowledge Discovery.* Oxford University Press, inc, 2002.

Rinzivillo, S., et al. 2008. Knowledge Discovery from Geographical Data. *Mobility, Data Mining and Privacy.* 2008, pp. 243-265.

Rushing, John, et al. 2005. ADaM: a data miningtoolkit for scientists and engineers. *Computers & Geosciences.* 2005, Vol. 31, 5, pp. 607-618 .

Sarawagi, Sunita, Agrawal, Rakesh and Megiddo, Nimrod. 1998. Discovery-Driven Exploration of OLAP Data Cubes. *Extending Database Technology.* 1998, Vol. 1377, pp. 168 - 182 . Proceedings of the 6th International Conference on Extending Database Technology: Advances in Database Technology.

SDMOQL: An OQL-based Data Mining Query Language for Map Interpretation Tasks . **Malerba, Donato, Annalisa, Appice and Vacca, Nicola. 2002.** 2002.

Seidl, Thomas and Kriegel, Hans-Peter. 1998. Optimal Multi-Step k-Nearest Neighbor Search. *ACM SIGMOD Record* . Juin 1998, Vol. 27, 2, pp. 154 - 165.

Shariff, A. Rashid. B.M and Egenhofer, Max. J. 1998. Natural Language Spatial Relations Between linear and areal objects: the topologie and metric of English language terms. 1998.

Shariff, A. Rashid. B.M, Egenhofer, Max. J. and Mark, D. 1998. Natural Language Spatial Relations Between linear and areal objects: the topologie and metric of English language terms. *International Journal of Geographical Information Science.* 1998, Vol. 12, pp. 215--246.

shekhar, shashi, et al. 2003. Trends in spatial data mining. 2003.

Shekhar, Shashi, et al. 2003. Trends in spatial data mining. 2003.

Shekhar, Shashi, et al. 2001. *What special about spatial data mining: Three case studies.* s.l. : R. Grossman, C. Kamath, V. Kumar, R. Namburu (eds.), 2001.

Similarity assessment for cardinal directions between extended spatial objects. **Goyal, Roop. K. 2000.** 2000.

Similarity of cardinal directions. **Goyal, Roop. K et Egenhofer, Max. J. 2001.** 2001.

Spatial Data Mining: A Database Approach. **Ester, Martin, Kriegel, Hans-Peter and Sander, Jtirg. 1997.** 1997.

Spatial Subgroup Mining Integrated in an Object-Relational Spatial Database. **Klösgen, Willi and MayFraunhofer, Michael. 2002.** 2002.

SPIN!—An Enterprise Architecture for Spatial Data Mining. **May, Michael and Savinov, Alexandr. 2003.** 2003.

Tan, Pang-Ning, Steinbach, Michael and Kumar, Vipin. 2006. *Introduction to data mining.* s.l. : Addison Wesley; US ed edition, 2006. ISBN-13: 978-0321321367.

TANAGRA. 2004. [Online] 2004. http://chirouble.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html.

Tekmono, Kardi. 2006. K-Means Clustering Tutorials. *Kardi Teknomo's Tutorial* . [Online] 2006. http://people.revoledu.com/kardi/tutorial/kMean/WhatIs.htm.

Teknomo, Kardi. 2009. Tutorial on Decision Tree. [Online] 2009. http://people.revoledu.com/kardi/tutorial/DecisionTree/index.html.

The MiningMart Approach to Knowledge Discovery in Databases. **MORIK, Katharina and SCHOLZ, Martin. 2004.** 2004.

Trends in spatial data mining. **shekhar, shashi, et al. 2003.** 2003.

Un logiciel open source pour l'enseignement et la recherche . **Ricco, RAKOTOMALALA. 2006.** 2006.

Vauglin, François. 1997. *Modèle statistiques des imprécisions géométriques des objets géographiques linéaires.* 1997.

—. 1997. Modèle statistiques des imprécisions géométriques des objets géographiques linéaires. 1997.

WEKA. 2008. [Online] 2008. http://weka.sourceforge.net/wekadoc/index.php/en:Weka_3.4.13.

WEKA: A Machine Learning Workbench. **Holmes, Geoffrey, Donkin, Andrew and Witten, Ian H. 1994.** 1994.

Widgets and Visual Programming. **Zupan, Blaz, et al. 2009.** 2009.

Wijsen, Jef. 2001. Data Mining et Data Warehousing. 2001.

Xu, Xiaowei, Martin Ester, Hans-Peter Kriegel and Jörg Sander. 1998. A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases. *Proceedings of the 14th International Conference on Data Engineering (ICDE 98).* 1998.

Zeitouni, Karine. 2002. *A Survey of Spatial Data Mining Methods Databases and Statistics Point of Views.* s.l. : IRM Press Hershey, PA, United States, 2002. pp. 229 - 242. ISBN:1-931777-02-0.

—. 2006. *Analyse et extraction de connaissances des bases de données spatio-temporelles.* 2006.

Zhang, Bin, Hsu, Meichun and Dayal, Umeshwar. 1999. K-Harmonic Means - A Data Clustering Algorithm. 1999.

Zytkow, Jan M. and Klösgen, Willi. 2002. Multidisciplinary contributions to Knowledge Discovery. *Handbook of Data Mining and Knowledge Discovery.* Oxford University Press, Inc, 2002.

Annexe A – Étude des outils open-source de fouille de données

A.1 Introduction

Le but ultime que nous nous sommes fixés dans la réalisation de la présente étude est d'arriver à une intégration efficace et efficiente de l'aspect spatial dans une bibliothèque open source de datamining. Seulement atteindre un tel objectif passe nécessairement par le choix d'un outil qui répond à un certain nombre de caractéristiques. Le choix de l'outil de datamining constitue, de ce fait, une étape majeure dans l'atteinte de l'objectif

Au regard du nombre impressionnant d'outils de datamining et des diverses caractéristiques à considérer lors de l'évaluation, effectuer un tel choix est loin d'être une tâche facile. Il faut en effet s'assurer que l'outil choisi répond parfaitement à nos besoins et s'intègre efficacement autant dans l'environnement logiciel que dans la philosophie du milieu.

Dans la présente, nous nous focaliserons d'emblée sur les outils de datamining du monde open source. Cela permet de réduire – à notre bonheur et dans une moindre mesure – le nombre d'outils à prendre en considération lors de l'étude. Mais plus important, le choix d'un outil open source, primo, offre au-delà de la disponibilité du code source, certains avantages tel que décrits dans *(Kamseu, et al., 2004) (Ajila, 2007)*:

- Disponibilité à moindre frais de l'outil,
- Participation d'une communauté plus ou moins active,
- Réutilisation et ajout d'autres fonctionnalités,
- Qualité du produit étroitement en rapport avec la qualité de la communauté participante (qualité dans la gestion du projet et dans le support)

Deuxio, l'esprit de ce choix s'inscrit dans la philosophie du groupe de recherche GeoSOA qui se veut un moteur en matière de technologies géo-Décisionnelles libres[40].

[40] http://geosoa.scg.ulaval.ca/

Dans la suite de cet écrit, nous étudierons premièrement les outils open source de datamining. Pour cela, nous décrirons les caractéristiques qui seront considérées lors de l'évaluation des différentes bibliothèques de datamining. Dans un deuxième temps, nous décriront et évalueront à proprement parler les outils de datamining à la lueur des caractéristiques alors définies. Il faut noter que ces sections seront essentiellement basées sur les études effectuées par *(Goebel, et al., 1999)* (Chen, et al., 2007) qui ont travaillé sur l'évaluation des outils de datamining et datamining open source – respectivement. En troisième lieu, nous choisirons un outil sur lequel nous allons procéder à une intégration de la composante spatiale.

A.2 Étude des outils open source de fouille de données

Le KDD[41] est un domaine qui de nos jours connait un engouement certain au regard du nombre assez impressionnant d'outils de datamining. Au vue des avantages qu'offre le KDD – extraction de connaissances potentiellement utiles - les entreprises ont vite adopté ces outils. Mais face à leur coût d'adoption, à l'incertitude qui plane quant à l'utilité du datamining au sein de l'entreprise et aux avantages qu'offre le monde open source (Chen, et al., 2007) , le pas a été vite franchi : bon nombre de méthodes et techniques de datamining se sont vu implémentées dans le monde open source. Bien des outils ont ainsi vu le jour (ADAM[42], AlphaMiner[43], Knime[44], etc.) et davantage, s'incrustent dans les organisations.

Dans les sections qui suivent, nous aborderons les caractéristiques à considérer, l'évaluation et le choix de ces outils. Mais avant d'entrér dans le vif du sujet, nous aborderons, sommairement un temps soit peu, les caractéristiques générales qui doivent guider le choix d'un logiciel libre.

A.2.1 Caractéristiques à prendre en considération lors du choix

Adopter un outil de datamining qu'il soit libre ou propriétaire n'est pas chose facile. En effet plusieurs paramètres rentrent en considération. Ces paramètres vont des critères de

[41] Knowledge Discovery in Database

[42] Voir http://datamining.itsc.uah.edu/adam/

[43] Voir http://www.eti.hku.hk/alphaminer/

[44] Voir http://www.knime.org/

sélection d'un logiciel de façon générale à ceux propres à un outil de datamining en passant par la spécificité des logiciels libres.

Parlant de spécificité des logiciels libres, le choix d'un tel outil répond à un certain nombre d'exigences que l'on peut scinder en deux grandes catégories *(Wang, et al., 2001)* : les exigences techniques et celles de gestion.

Les exigences techniques renvoient aux caractéristiques d'ordre architectural, opérationnel et de développement (entendu dans le sens implémentation). Plus spécifiquement, il s'agit de la:

- *Disponibilité d'une documentation* : il est nécessaire qu'il y'ait un support afin de permettre une adoption facile du logiciel.
- *Evolutivité* : il s'agit de voir dans quelle mesure l'outil est évolutif et quelle est la quantité de travail nécessaire pour ajouter un module. Naturellement cet effort devrait être moindre
- *Compatibilité par rapport aux standards* : il s'agit de voir si le logiciel suit certaines normes établies autant dans ses versions actuelles que futures.
- *Personnalisation et extensibilité* : un outil candidat devrait être extensible. Peut-on intégrer un module propriétaire ? il faut également évaluer les dépendances de l'outil avec le système d'exploitation afin de juger d'une possibilité d'extension.
- *Haute fiabilité* : il s'agit de la capacité de l'outil à produire des résultats conformes à ceux attendus.

Les exigences de gestion quant à elle, réfèrent à la licence, à la maintenabilité et à l'allocation de ressources. Il s'agit d'avoir un œil sur :

- *Les questions d'ordre budgétaires* : il faut noter que même si les logiciels libres sont plus ou moins gratuits, il nécessite d'autres frais (développement, maintenance, support) qui peuvent progressivement s'accroitre.
- *L'expertise de l'équipe de développement* : il est important que l'équipe soit familière avec les langages et outils du monde libre (perl, Unix,…). Cela permet de réduire les coûts

- *La Licence*: divers types de licences existent (GPL[45], LGPL[46], BSD[47], CPL[48]) chacune avec ses caractéristiques.
- La maintenabilité: il s'agit de s'assurer de la maintenabilité à long terme du logiciel.

Caractéristiques générales

Sous ce terme sont regroupés les critères liés au choix d'une licence appropriée, les questions liées à l'utilisabilité c'est-à-dire l'interactivité, l'interopérabilité et l'extensibilité *(Chen, et al., 2007)* .

Au niveau de ces critères, le choix de la licence, pour nous, correspond à une problématique de premier ordre. En effet, il faudra veiller lors de la sélection de l'outil de datamining, au choix d'une licence qui s'inscrit dans la philosophie du groupe de recherche d'une part et qui d'autre part, permet une participation active de la communauté quant à son amélioration. On note dans le monde open source, comme principaux licences *(Perens, 1999)* :

- General Public Licence (GPL) : cette licence ne permet pas de rendre les modifications privées. Toute modification devrait être distribuée sous licence GPL. L'auteur reçoit de ce fait toute modification même propriétaire. En plus, cette licence ne permet pas l'incorporation à un outil propriétaire.
- Library GPL (LGPL) : il s'agit d'une déclinaison de GPL pour des « parties » de logiciel. A la différence de GPL, les modules sous licence LGPL peuvent être incorporés dans des logiciels propriétaires.
- X, Berkeley Software Development (BSD), licence apache : ces licences diffèrent des licences GPL et LGPL en ce sens qu'elles permettent d'effectuer presque toutes modifications sur le logiciel. En plus les modifications peuvent être rendues privées.

---

[45] General Public License

[46] Library GPL

[47] Berkeley Software Development

[48] Community Public License

- Netscape Public License (NPL) et Mozilla Public License(MPL) : il s'agit de licences particulières développées respectivement pour Netscape et Mozilla. NPL stipule que Netscape se donne le droit de changer la licence du produit et de ne faire suite à toutes modifications apportées au logiciel. MPL quand à elle est une version plus open source de NPL où les droits de Netscape ont été retirés.
- licence Artistic: il s'agit d'une licence développée à l'origine pour PERL et qui s'est peu à peu étendue à d'autres langages et outils. Elle contient volontairement des lacunes qui permettent d'outrepasser les exigences. Cette licence interdit la vente d'un logiciel mais toute modification apportée à un logiciel peut être commercialisée.

L'utilisabilité renvoie à la facilité de prise en main de l'outil dans la réalisation des tâches de datamining. Cette notion renvoie notamment à *(Chen, et al., 2007)*:

- Interactivité qui décrit le niveau d'intervention de l'homme dans la réalisation des tâches. Quel est le niveau d'interactivité du processus de découverte de connaissances : autonome, guidé, ou complètement manuel.
- L'interopérabilité renvoie à l'interaction de l'outil avec d'autres. Echange de données ou de modèles grâce au support de PMML[49] ou CWM[50]
- Extensibilité qui décrit dans quelle mesure l'outil est extensible.

Caractéristiques techniques

Ces caractéristiques renvoient d'une part à l'aspect base de données et d'autre part à l'aspect fonctionnalité qui réfère aux algorithmes et méthodes implémentées.

L'aspect base de données renvoie principalement à deux notions :

---

[49] *Predictive Model Markup Language* est un langage de balise basé sur XML conçu pour définir des modèles de données et visant à rendre interopérables les systèmes de datamining. (wikipedia.org)

[50] *Common Warehouse Metamodel* est une spécification qui décrit un langage d'échange de métadonnées à travers un Entrepôt de données, un système décisionnel, un système d'ingénierie des connaissances (wikipedia.org)

- Premièrement à la notion de connectivité qui décrit la capacité de l'outil en considération d'accéder à diverses sources de données et de traiter différents types de fichier. Il s'agit, plus explicitement, du support des SGBD (systèmes de gestion de bases de données) courants (Oracle, Access, MySQL) en passant par des pilotes natifs ou par ODBC (Open DataBase Connectivity) ou JDBC (Java DataBase Connectivity). Mais aussi de la possibilité de traiter les formats aussi divers que CSV (Comma-Separed Values), Excel, ou plus propriétaire tel que ARFF (Attribute-Relation File Format).
- Deuxièmement, la connectivité renvoie à la capacité de traitement de volumes d'informations. Est-ce que l'outil étudié peut traiter une quantité moyenne ou élevée de données ?

L'aspect fonctionnalité réfère lui à la disponibilité des différentes tâches de datamining disponibles au sein de ces outils. Ces tâches peuvent être catégorisées comme suit *(Goebel, et al., 1999)* :

- Prétraitement des données : il s'agit de la sélection, du filtrage, de la transformation des données afin de les intégrer souplement dans le processus de datamining. Disposer d'une telle fonctionnalité permet de s'affranchir de la mise en œuvre de routines complexes et permet de gagner en temps en termes de productivité ;
- Prédiction : il s'agit de la prédiction de la valeur d'un attribut ; peut être utilisée pour valider ou infirmer une hypothèse ;
- Régression : consiste en l'analyse de la dépendance de plusieurs attributs et de prédire les valeurs de nouveaux enregistrements.
- Classification : il s'agit de déterminer à quelle classe prédéfinie, appartient un enregistrement donné ;
- Clustering : partitionnement en groupe de caractéristiques communes
- Association : il s'agit de déterminer les relations entre attributs de sorte à déterminer si la présence de l'un implique l'autre ;
- Visualisation de modèles : permet de comprendre aisément les résultats de la fouille.

A.2.2 Quelques logiciels de fouille de données

ADAM (Algortihm Development and Mining system)

(Rushing, et al., 2005) (ADAM, 2009) ADAM est un logiciel multiplateforme (Windows, Linux, Mac) qui comporte une double fonctionnalité : fonction de traitement d'images d'une part et de fouille de données d'autre part. En ce qui concerne la fonctionnalité de fouille de données, ADAM implémente bon nombre de méthodes et techniques de fouille de données : clustering, classification, règles d'associations. Il dispose en plus de fonctions de prétraitement, de filtrage et de réduction de la taille des données à fouiller. Les fonctions de traitement d'image sont toutes aussi variées que celles de fouille.

ADAM se compose d'un ensemble de composants hautement interopérables et fournissant des interfaces dans des langages aussi divers que C, C++, Perl, Python et même sous forme de webservices. En plus de l'interopérabilité, la nature distribuée de l'architecture d'ADAM constituent ses principaux atouts. Ces atouts font d'ADAM un logiciel hautement flexible et personnalisable. On peut en effet, combiner les différents composants disponibles pour ADAM afin d'obtenir un logiciel spécifique à nos besoins.

Concernant la licence, ADAM n'est pas explicitement sous une licence open source connu. Il est surtout connu comme étant propriété de l'Université d'Alabama à Huntsville. Il est seulement utilisable à des fins éducationnelles et de recherches aux États Unis ; et juste en mode évaluation.

ORANGE

*(Demsar, et al., 2004) (Zupan, et al., 2009) (Chen, et al., 2009) (Orange, 2009)*Orange est un logiciel de fouille de données implémenté en C++ et disponible sous licence GPL. Tout comme ADAM, il est hautement personnalisable en ce sens qu'il s'agit d'un framework qui se compose de plusieurs composants indépendants. Il dispose également d'un langage de Scripting en l'occurrence Python qui permet d'implémenter assez aisément des tâches de fouille de données. De plus, l'adaptateur pour les différents composants sont fait dans ce langage.

En termes de fonctionnalités, Orange, en plus de disposer d'un ensemble de fonction d'accès, manipulation, prétraitement des données, supporte un grand nombre d'algorithmes de fouille de données. On note cependant que l'une des grandes forces de cet outil réside dans l'aspect interface utilisateur et particulièrement la visualisation de données. En effet, Orange est constituée d'un ensemble de composants dénommés Orange widgets. Ces composants offrent différentes fonctionnalités allant de la possibilité de

personnalisation, au goût de l'utilisateur, du modèle à visualiser, au choix des données à manipuler en passant par les différents algorithmes de fouilles.

L'interactivité au niveau d'Orange est assurée par la mise en connexion de ces widgets au sein d'un framework, le canevas orange. Ces objets communiquent entre eux à travers des canaux de communication typés réalisé grâce au langage Python.

Du point de vue connectivité et quantité de données traitables, il ne peut traiter qu'un volume moyen de données disponibles sous MySQL.

TANAGRA

TANAGRA est un logiciel open source (avec toutefois une licence autre que celles classiques) de fouille de données qui dérive de SIPINA[51], également un logiciel open source qui implémente principalement des algorithmes de classification. Il poursuit principalement trois(3) objectifs *(Rakotomalala, 2006)*:

- Plateforme pour l'enseignement ;
- Logiciel de recherche ;
- Outil pédagogique d'apprentissage de la programmation ;

(Chen, et al., 2009) (TANAGRA, 2004) Développé avec le langage C++, TANAGRA offre bon nombre de fonctionnalités reparties sur différents composants. On note les composants de :

- accès aux données ;
- visualisation ;
- réalisation d'opérations de fouille de données (analyse factorielle, régression, statistiques descriptives, clustering, association,…).

Ces différents composants peuvent être assemblés à la manière d'une structure en arbre afin d'implémenter l'ordre de réalisation des tâches de fouille de données.

L'inconvénient majeur de cet outil est sa non-disponibilité dans les autres plateformes autre que Windows. Également, en termes de connectivité aux sources de données, il ne permet pas d'accéder aux systèmes de gestion de base de données. Seulement les sources de données fichiers sont accessibles notamment fichiers tabulaires et sous format Excel.

---

[51] http://eric.univ-lyon2.fr/~ricco/sipina.html

KNIME (Konstanz Information Miner)

*(KNIME, 2009)*KNIME est un logiciel de fouille de données disponible sous double licence : une licence open source, GPL en l'occurrence et une propriétaire. Construit grâce à l'API d'Eclipse, KNIME bénéficie d'une grande modularité et d'une interactivité certaine grâce à ses multiples composants qui peuvent être structurés sous forme de flow de données (tout comme Orange).

L'architecture de KNIME a été construite autour de trois (3) principes majeurs *(Berthold, et al., 2006)* :

- Framework interactif et visuel : possibilité de combiner divers composants en vue de mettre en œuvre ses opérations de fouilles de données.
- Modularité : indépendance totale ou moindre entre les composants et flexibilité des types de données. En effet aucun type n'est prédéfini, de nouveaux types peuvent être ajoutés facilement et déclarés compatibles avec ceux existants.
- Extensibilité : l'outil devrait être extensible sans grande modification.

Les fonctionnalités sous KNIME sont distribuées sous forme de composants représentées par des nœuds qui peuvent être inter reliés par des arcs pour former un flot de données. Les segments propagent les données entre les nœuds qui disposent chacun d'un état à un instant donné (configurer, exécuté,…). Comme composants on note :

6. Les composants d'accès aux sources de données : il permet d'accéder à des sources aussi diverses que les fichiers, les bases de données grâce à JDBC mais aussi d'y écrire. La nouveauté dans KNIME réside dans la possibilité pour les composants d'accès aux données, la lecture/écriture des fichiers au format PMML (beta).
7. Les composants de manipulation et de transformation de données : ils permettent d'assurer les opérations de filtrage de colonnes, trie, jointure, fusion, échantillonnage de données
8. Les composants de fouille de données : ils comportent des algorithmes de clustering, d'association, d'induction de règles, de régression

9. Les composants permettant de réaliser des opérations statistiques telles que les corrélations linéaires, le comptage de valeurs, de régression linéaire et polynomiale.
10. Les composants de visualisation

Comme nous l'avons noté ci haut, KNIME est hautement extensible en ce sens qu'on peut y ajouter des extensions provenant d'autres outils de fouille (Weka), ou des composants provenant d'outils statistiques et graphiques. Également, il est possible d'ajouter ses propres extensions en redéfinissant sans grande difficulté quelques classes.

MINING MART

*(Morik, et al., 2003)*MINING MART est un logiciel de fouille de données qui est spécialisé dans le traitement de gros volumes d'informations. La particularité de l'approche de MINING MART réside dans l'optimisation du processus de découverte de connaissances. Cela passe nécessairement par la réduction de la charge et du temps de travail consacré à l'étape de prétraitement des données.

Afin d'atteindre cet objectif, l'architecture du logiciel est composé d'un méta modèle dénommé M4 qui décrit toutes les étapes du prétraitement ainsi que les données elles mêmes ; les données pouvant provenir de base ou d'entrepôt de données. Disposer d'un langage de description des données (métadonnées) comporte bien d'avantages:

- Abstraction : les métadonnées sont fournies à deux(2) niveaux d'abstraction : conceptuel (abstrait) et relationnel (exécutable). Cela permet au modèle d'être compréhensible et réutilisable.
- Documentation des données : les sources de données ainsi que les attributs impliqués dans la chaine de prétraitement sont décrits conformément aux deux niveaux d'abstraction.
- Documentation des cas : il s'agit de documenter les différents opérateurs de la chaine de prétraitement.
- Facilité d'adaptation des cas : les cas ainsi décrit peuvent être appliqués dans des situations aussi diverses, moyennant une certaine adaptation des modèles relationnels et conceptuels, pour peu qu'il y'ait une certaine ressemblance.

Pour revenir aux deux(2) niveaux d'abstraction de M4, on peut noter que par analogie avec le modèle relationnel, le niveau abstrait (conceptuel) au niveau de M4 correspond à un modèle conceptuel de données qui décrit les sujets d'intérêt sous forme d'entités et de relations entre ces entités. L'avantage avec M4 étant la possibilité de déclarer un arbre d'hiérarchies entre les entités. Le modèle relationnel (exécutable) correspond à une instance particulière du modèle conceptuel. C'est à ce stade que l'instance de base de données est définie, de même que les différentes tables qui participent au traitement. Cela grâce à un compilateur chargé de traduire le modèle conceptuel en exécutable.

En réalité, MINING MART est beaucoup plus axé dans le prétraitement des données que dans la réalisation d'opérations de fouille

MLC ++

(Kovahi, et al., 1994) MLC++ est une bibliothèque open source de fouille de données développée par l'Université de Stanford. Il s'agit d'une implémentation C++ d'une variété d'algorithmes d'apprentissage supervisé. L'objectif majeur qui a sous-tendu la mise en œuvre de cette bibliothèque est la réutilisation et l'extensibilité à moindre coût. Les fonctionnalités au niveau de MLC++ sont organisées autour des principales classes allant des classes d'utilité générale, à celles servant à la visualisation en passant par les classes implémentant les algorithmes courant d'apprentissage.

En termes d'accès aux sources de données, MLC++ ne permet pas l'accès aux systèmes de gestion de base de données ; les données sont fournies par l'intermédiaire d'un fichier.

ALPHAMINER

*(Alphaminer, 2005)*ALPHAMINER est un logiciel open source disponible sous licence GPL. Implémenté en java et disponible sur diverses plateforme (Windows, Mac, Linux), il offre plusieurs fonctionnalités organisées autour des catégories suivantes :

- Accès des données : le logiciel permet d'accéder à des données disponibles dans des fichiers ou stockées dans une base de données via ODBC.
- Exploration de données : il s'agit de récupérer certaines informations statistiques sur les données en entrée.

- Transformation de données : ces transformations regroupent l'échantillonnage, la normalisation, le traitement des valeurs manquantes
- Modélisation : cette fonctionnalité regroupe les différentes classes d'algorithmes de fouille de données : association, clustering, régression, arbre de décision.
- Evaluation : il s'agit de l'évaluation, par la production d'une matrice de confusion ou d'un graphe d'évaluation, de la précision et de la performance des modèles résultant de la fouille.
- Déploiement : Alphaminer supporte la fonction score qui donne une prédiction ou une classification d'un nouvel ensemble de données sur la base d'un arbre de décision ou un modèle de régression linéaire préalablement construit.

L'ensemble des fonctionnalités décrites ci haut, peuvent se combiner grâce au système de workflow supporté par Alphaminer.

Du point de vue de l'extensibilité, des modules additionnels peuvent facilement être ajoutés au logiciel sans grandes modifications. En effet, des fonctionnalités supplémentaires peuvent être ajoutées grâce à la mise en œuvre de composants basés XML. En plus il a été préalablement intégré à deux autres logiciels de fouille open source non moins connus (Weka, Xelopes[52]).

XELOPES (eXtEnded Library fOr Prudsys Embedded Solutions)

*(Linux-Online, 2008) (PrudSys AG, 2009)*XELOPES est une bibliothèque open source de fouille de données embarquée implémentée dans diverses langages (Java, C++, C#). Produit de la société PrudSys AG, il est disponible sous licence GPL. XELOPES a été construit sur la base du standard CWM et offre par ailleurs plusieurs avantages dont :

- Support du standard CWM : possibilité d'échanger des données avec d'autres applications décisionnelles.

[52] http://www.prudsys.com/Produkte/Algorithmen/Xelopes

- Support des standards décisionnels : des standards comme PMML, JOLAP[53], JMI[54] sont supportés.
- Indépendance à la plateforme : peut être exécuté sur diverse plateformes grâce aux différentes implémentations disponibles.
- Indépendance à la source de données : le logiciel peut accéder à divers types de sources de données incluant les fichiers plats, les bases de données. Aussi, en plus de la possibilité de traiter des données changeantes, il est capable de traiter de larges volumes d'informations.

XELOPES est un outil complet et tout aussi riche. Son architecture complète, robuste et complexe témoigne de sa particularité qui réside essentiellement en la capacité, contrairement à d'autres, d'accéder à des données situées dans un entrepôt.

RATTLE

(Graham, 2006) RATTLE est une application graphique de fouille construit autour de R qui est un outil statistique sous licence libre (GPL) et également un langage de programmation. Les avantages de RATTLE sont intrinsèquement liés à ceux de R. on note que R possède les avantages suivants :

- R possède une maturité certaine car résultant de l'implication de statisticiens et chercheurs expérimentés lors de son implémentation
- R dispose de bibliothèques dans des domaines aussi divers que l'économétrie, l'analyse spatiale, la bio informatique,…
- R permet d'accéder à diverses sources de données,
- R peut produire des graphiques dans divers formats : PDF, SVG, JPG, PNG

Les fonctionnalités de RATTLE sont organisées autour des catégories suivantes :

- Le chargement des données : chargement des données depuis diverses sources : fichier (CSV, ARFF, Rdata[55]) et base de données via ODBC

[53] Java OLAP interface

[54] Java Metadata Interface : spécification java permettant l'implémentation d'une plateforme Indépendante pour la création, la gestion, le stockage, l'échange des métadonnées

[55] Extension propre à R (et à Rattle)

(Access, Oracle, MySQL, SQLite). Cette fonctionnalité inclut l'échantillonnage des données et l'attribution de rôles particuliers à certaines variables.

- Exploration et transformation des données : il s'agit de comprendre les données. Cela passe par l'obtention d'informations statistiques sur les données ; l'étude des variables et des liens qu'elles entretiennent entre elles ; le traitement des valeurs manquantes. En plus de disposer de ses propres fonctions d'exploration de données, RATTLE peut également intégrer GGOBI qui est une bibliothèque open source très interactive et performante dédiée à l'exploration des données. La transformation des données renvoie à leur nettoyage, suppression des valeurs doubles, normalisation, etc.
- Modélisation : cela ramène à l'utilisation des différentes classes d'algorithmes de fouille : analyse de clusters, analyse d'association, classification et régression. Plusieurs algorithmes de ces classes de fonctions se retrouvent implémentés dans RATTLE. On note par exemple que pour la régression, le logiciel supporte les régressions généralisées, logistiques, multinomiales.
- Evaluation et déploiement : RATTLE offre plusieurs options pour l'évaluation de la performance ainsi que les taux d'erreurs associés à un modèle.

RATTLE est certes un puissant outil de fouille et possède une interface graphique attrayante mais à la différence d'autres outils comme ORANGE, KNIME, il ne dispose pas d'une structure de workflow où les différentes tâches peuvent être mises en relation et par conséquent assurer une certaine interactivité.

WEKA (Waikato Environment for Knowledge Analysis)

*(Holmes, et al., 2005)*WEKA est une collection d'algorithmes dédié à la réalisation de tâches de fouille de données. Développé à l'université de Waikato, WEKA est un projet open source disponible sous licence GPL. Implémenté en java, et de ce fait portable, WEKA dispose de plusieurs fonctionnalités disponibles au niveau des différents modules le

composant. Au-delà des fonctionnalités classiques de fouilles de données, WEKA intègre des fonctions de prétraitement et de post-traitement.

WEKA peut être utilisé de deux(2) façons : ou bien en tant que logiciel autonome ou bien en tant que bibliothèque attachée à une application dans laquelle les fonctions peuvent être appelées dans le code source. En tant qu'application autonome, et dans le but de faciliter la prise en main et l'interactivité, les fonctionnalités de WEKA ont étés rendues disponibles dans divers composants. On note comme composants :

- Simple CLI : il s'agit d'une interface ligne de commandes qui permet d'accéder à l'ensemble des classes disponibles dans WEKA.
- Experimenter : cette interface donne la possibilité à un utilisateur donné de mener des tests sur les données en vue de déterminer quel modèle fournit le meilleur résultat. Ces tests ou expériences peuvent être menés localement sur une seule machine ou de façon distribuée. Bien qu'il existe une interface graphique qui permet d'effectuer facilement ces manipulations, le module peut également être lancé en mode ligne de commandes.
- Explorer : cette interface graphique exploite les fonctionnalités offertes par les différents packages de WEKA allant du prétraitement à la visualisation des résultats en passant par la modélisation (datamining).
- Knowledge flow : il s'agit de l'interface permettant de combiner les différentes tâches de découverte de connaissances à la manière d'un workflow. Il offre comme avantages entre autre la possibilité de traiter des données en mode batch et de façon incrémentale ; traiter parallèlement des tâches batch et visualiser les résultats pendant le traitement.

En termes de connectivité aux sources de données, WEKA permet d'accéder à des sources diverses. En effet, il permet d'accéder aux bases de données courantes par l'intermédiaire de JDBC. Également divers formats de fichier sont accessibles : CSV, ARFF. En réalité, quel que soit le type de source de données accédée, WEKA converti les données dans un format ARFF avant d'effectuer les différents traitements.

RAPID MINER

(Mierswa, et al., 2006) RAPID MINER anciennement YALE est un logiciel open source (GPL) de fouille de données qui implémente une variété d'algorithmes. La particularité de RAPID MINER est sa capacité à permettre un prototypage rapide d'application à moindre coût.Dire de RAPID MINER qu'il permet le prototypage, revient à dire qu'il répond aux exigences suivantes :

- Flexibilité en ce qui concerne les fonctionnalités de prétraitement et de fouilles de données ;
- Accessibilité à plusieurs sources de données ;
- Facilité d'utilisation.

RAPID MINER répond en effet à l'ensemble de ces exigences. D'un point de vue utilisabilité, le logiciel est facile à prendre en main au sens où il fournit une interface utilisateur riche. Les fonctionnalités sont reparties sous forme d'opérateurs (composants) ; lesquels peuvent être combinés pour obtenir un graphe à une structure arborescente qui modélise le workflow. Cette structure est également représentée sous forme XML. Ce qui permet sa lisibilité autant par l'homme que par une machine. Notons que cette structure - sous forme d'arbre - est également riche en ce sens qu'elle permet d'appliquer sur les opérateurs des boucles et des conditions.

Les fonctionnalités disponibles sous RAPID MINER sont aussi riches que variées. On note les :

- Fonctions de prétraitement : il s'agit des opérateurs de discrétisations, de réduction de dimensions, d'échantillonnage, de normalisation ;
- Fonctions de transformation : traitement des valeurs manquantes, extraction, pondération de variables ;
- Fonctions d'analyse : il s'agit des différents algorithmes de fouille : arbres de décision, processus gaussiens, réseaux de bayes, régression, clustering, association ;
- Fonctions d'évaluation et d'optimisation ;
- Fonctions de visualisation.

Pour les fonctionnalités non disponibles, RAPID MINER fournit des API[56] qui permettent d'en effectuer facilement une implémentation. La description possible des

[56] Application Programming Interface

opérateurs sous forme XML permet d'intégrer aisément les nouveaux opérateurs dans l'interface graphique de l'outil. L'extensibilité de l'outil ne fait donc pas de doute.

En ce qui concerne la connectivité aux sources de données, RAPID MINER peut accéder nativement aux bases de données Oracle, MySQL, SQL server, Sybase. Les autres types de base de données peuvent être accéder grâce au connecteur JDBC. En ce qui concerne les fichiers, le logiciel peut accéder aux fichiers de format ARFF, CSV, Excel. Quant à la quantité de données traitable, tout comme WEKA, RAPID MINER ne peut traiter qu'un volume moyen de données.

A.3 Synthèse et choix d'un outil de fouille de données

Les tableaux suivant résument les différentes analyses comparées des outils de fouille de données en fonction des différents critères évoqués ci-dessus. Ces synthèses s'inspirent des travaux de *(Chen, et al., 2007)* sur la comparaison d'outils open-source de fouille de données.

A.3.1.Synthèse selon le critère caractéristiques générales

Les figures suivantes (cf. Figure A.3-1, **Erreur ! Source du renvoi introuvable.**) ésument les comparaisons selon le critère « caractéristiques générales »

Outils	Licence	Langage	Windows	Linux	Mac
ADAM	inconnu	C, Perl, C++, Python	x	x	x
AlphaMiner	GPL	Java	x	x	x
KNIME	GPL	java	x	x	x
Mining Mart	Inconnu	Java	x	x	x
MLC++	Autre	C++	x	x	
Orange	GPL	C++, Python	x	x	x
Rapid Miner	GPL	Java	x	x	x
Rattle	GPL	R	x	x	x
Tanagra	Autre	C++	x		
Weka	GPL	java	x	x	x
Xelopes	GPL	Java, C++, C#	x	x	x

Figure A.3-1 : Synthèse selon la licence d'utilisation et la plate-forme

Outils	Interaction	Interopérabilité	Extensibilité
ADAM	Autonome	-	Simple
AlphaMiner	Manuel	PMML	Excellent
KNIME	Manuel	PMML	Excellent
Mining Mart	Manuel	-	Simple
MLC++	Guidé	-	Simple
Orange	Manuel	-	Excellent
Rapid Miner	Manuel	-	Excellent
Rattle	Guidé	PMML	Simple
Tanagra	Manuel	-	Simple
Weka	Manuel	-	Excellent
Xelopes	-	PMML,JMI,JOLAP	-

Figure A.3-2 : Comparaison des outils de fouille de données selon le critère utilisabilité

A.3.2.Synthèse selon le critère caractéristique technique

Les figures ci-dessous (cf. Figure, Figure A.3-4, Figure A.3-5) résument une comparaison des outils de fouille de données selon les caractéristiques techniques à savoir les différentes bases de données auxquelles ces outils accèdent ainsi que les fonctionnalités de fouille de données offertes.

Outils	Oracle	Sysbase	sqlServer	Mysql	Access	ODBC	JDBC	ARFF	CSV	Excel	OLAP
ADAM								x			
AlphaMiner					x	x		x	x	x	
KNIME				x	x	x	x	x	x	x	
Mining Mart	x				x						x
MLC++											
Orange				x							
Rapid Miner	x	x	x	x			x	x	x	x	
Rattle				x	x	x			x	x	
Tanagra								x		x	
Weka							x	x	x		
Xelopes				x			x		x	x	x

Figure A.3-3: Synthèse selon le type de bases de données accédées

Outils	Volume de données
ADAM	Large
AlphaMiner	Moyen
KNIME	Moyen
Mining Mart	Inconnu
MLC++	Large
Orange	Moyen
Rapid Miner	Moyen
Rattle	Large
Tanagra	Moyen
Weka	Moyen
Xelopes	Inconnu

Figure A.3-4: Synthèse des outils de fouille selon le volume de données traitées

Outils	Prétraitement	Clustering	Prédiction	Règles d'association	Évaluation	visualisation
ADAM	Excellent	x	x	x	-	Excellent
AlphaMiner	Excellent	x	x	x	x	Excellent
KNIME	Excellent	x	x	x	x	Excellent
Mining Mart	Excellent					Excellent
MLC++	Excellent					Excellent
Orange	Excellent	x	x	x	x	Excellent
Rapid Miner	Excellent	x	x	x	x	Excellent
Rattle	Bien	x	x	x	x	Excellent
Tanagra	Excellent	x	x	x	x	Moyen
Weka	Excellent	x	x	x	x	Excellent
Xelopes	-	x	x	x	-	-

Figure A.3-5: Synthèse des outils selon les fonctionnalités de fouille offertes

A.4 Conclusion

Dans le présent document, nous avons fait un tour d'horizon des différents logiciels open source de fouille de données existants. Comme on peut le noter, une pléthore de

bibliothèques existe chacune offrant des fonctionnalités aussi diverses que variées. Mais au-delà de cette variété dans la fonctionnalité, on peut noter que des bibliothèques passées en revue, implémentent pour la plupart les principaux algorithmes de fouille. De ce fait, lors du choix d'un outil, l'accent devrait être mis sur des caractéristiques telles la licence, les fonctionnalités de fouille offerte et l'utilisabilité.

Au regard de ces caractéristiques, notre choix s'est porté sur la bibliothèque KNIME qui comparativement aux autres, constitue le meilleur compromis notamment en termes d'algorithmes implémentés, d'interactivité, de flexibilité quant à la prise en compte d'un nouveau type de données *(Berthold, et al., 2006)* ; ce qui est assurément, un de nos objectifs.

Annexe B – Techniques de clustering et de classification

B.1. Introduction

Le KDD est un domaine qui rappelons le, permet la découverte, la mise à nues de connaissances non explicites et potentiellement utiles au sein d'un gros volume d'information. Pour ce faire, le KDD à travers l'étape de modélisation (fouille proprement dite) offre plusieurs techniques et méthodes à même d'atteindre cet objectif. Chacune de ces techniques et méthodes, aussi séduisantes les unes que les autres, et fort utiles sont à utiliser dépendamment du contexte et de l'objectif sous tendu par la fouille de données.

Dans la présente étude, l'objectif à terme est d'arriver à une intégration efficace, efficiente de l'aspect spatial au sein d'une bibliothèque open-source de fouille de données. Atteindre cet objectif, passe nécessairement par le choix – parmi la pléthore de techniques de fouille existantes - d'un certain nombre d'algorithmes à implémenter.

Ce choix, loin d'être basé sur des considérations subjectives, trouve sa justification d'une part dans l'impossibilité d'implémenter, dans le temps imparti pour la présente étude, toutes les techniques de fouille. D'autre part, toutes les techniques de fouille ne sont pas

adaptées pour le domaine spatial du fait de interdépendance des entités géo-spatiales *(Koperski, et al., 1996) (Ester, et al., 1997) (Shekhar, et al., 2003)*.

Le présent document se donne pour objectif de choisir et décrire des techniques de fouille de données qui s'adaptent mieux à la prise en compte de la composante géo-spatiale. Notre choix s'est porté sur deux techniques notamment le K-Nearest Neighbors et le K-Means qui d'une part dans leur philosophie, sont proche de celle du domaine spatial ; principalement dans la considération de l'influence du voisinage dans les tâches de prédiction/classification. D'autre part, le choix de ces deux techniques s'explique aussi par la simplicité et la facilité d'implémentation de leurs algorithmes.

Dans la suite du document, nous décriront ces deux (2) algorithmes de fouille de données qui accomplissent deux fonctions distinctes (la classification et le clustering). Ainsi dans un premier temps, nous parlerons du K-Nearest Neighbors en décrivant ses principes, ses caractéristiques principales, son algorithme, ses avantages et ses inconvénients. Par la suite, suivant la même démarche, nous aborderons l'algorithme de clustering K-Means.

B.2. K Nearest Neighbors

Principes

K-Nearest Neighbors est un algorithme de classification supervisée qui permet de déterminer les caractéristiques d'une observation donnée sur la base d'un nombre prédéterminé de voisins *(Tekmono, 2006)* **.** Qualifié d' « Instance based learning[57] », du fait qu'il n'utilise pas de modèle pour la prédiction, le K-Nearest Neighbors est un des algorithmes de classification, voire de fouille de données le plus utilisé *(Hand, et al., 2001) (Larose, 2005)*. Sa force vient principalement de sa simplicité et de sa facilité de mise œuvre.

L'algorithme KNN, fonctionne sur la base de deux (2) jeux de données passés en entrée. Le premier de ces jeux de données constitue le « training set », c'est-à-dire les données dont on va se servir pour prédire la valeur d'une variable donnée. Notons que ce jeu contient les valeurs de l'ensemble des variables y compris celles dont on veut prédire la

[57] Aussi qualifié de « Lazy Learning » ou « Example Based Learning »

valeur. Le deuxième jeu de données en entrée de l'algorithme, constitue les données dont on veut prédire la valeur d'un attribut (d'une variable). Ce jeu contient les valeurs de toutes les variables sauf celles de l'attribut dont on veut prévoir la valeur.

Les observations, qu'il s'agisse du « training set » ou du jeu de données dont on veut prédire les valeurs, sont considérées comme des points d'un espace p-dimensionnel. La tâche majeure de l'algorithme K-Nearest Neighbors, comme on l'a noté plus haut, consiste à prédire les différentes valeurs d'un attribut sur la base d'un nombre K de voisins dans cet espace (p-dimensionnel). À la lecture de cela, on note que l'application de l'algorithme soulève quelques questions :

- Comment évaluer la notion de voisinage ?
- Quel est le nombre de voisins à considérer. En d'autres termes, comment fixer la valeur de K ?
- Est-ce que toutes les variables ont le même niveau d'importance ?
- Comment combiner les différentes valeurs d'observations afin de déterminer la valeur de la variable cible ?

Evaluer la notion de voisinage

Évaluer le voisinage au niveau du Nearest Neighbors revient à mesurer la proximité entre deux points de notre espace. Appliquer une technique d'évaluation de la proximité revient tout simplement en l'utilisation d'une fonction plus ou moins complexes de distance *(Seidl, et al., 1998)*.

La mesure de la distance ou de la similarité/dissimilarité[58] prend toute son importance au niveau de l'algorithme. En effet, en fonction de la nature des variables et du contexte, une méthode sera privilégiée plus qu'une autre. De ce fait, *(Giudici, et al., 2009)* notent que la distance euclidienne est plus adaptée aux variables quantitatives. Tandis que les mesures de similarité peuvent être autant appliquées aux variables quantitatives que qualitatives.

Autant il existe plusieurs méthodes d'évaluation de la distance dans un espace à n dimensions, autant il existe plusieurs méthodes de mesure de la similarité. On note comme techniques de mesure de la distance *(Tekmono, 2006)*:

---

[58] Voir http://people.revoledu.com/kardi/tutorial/Similarity/WhatIsSimilarity.html#Distance pour de plus amples connaissances sur la définition mathématique de la distance et des mesures de similarité.

- La distance euclidienne
- La distance de Manhattan
- La distance de Tchebychev
- La distance de Minkowski

L'utilisation d'une méthode à la place d'une autre dépendra du contexte, de l'objectif sous tendu par l'utilisation de la méthode et des avantages et inconvénients qu'offre ladite méthode.

Au nombre des différentes techniques de mesures de similarité, on note :

- La méthode Russel-Rao,
- La méthode Jaccard,
- La méthode Sokal-Michener.

Bien que le choix d'une fonction d'évaluation de la distance soit d'une importance capitale dans la détermination des voisins, il est important de procéder à une normalisation des différentes valeurs avant de procéder à l'évaluation de la distance ou de la similarité entre deux (2) points de l'espace. Cela permet d'une part de mettre les différentes valeurs sous la même échelle ; et d'autre part permet d'éviter de biaiser le calcul de la distance (similarité) *(Larose, 2005)*. Il existe différentes méthodes de normalisation notamment Z-score et Min-Max, etc.

Fixation de la valeur de k

Le choix du nombre de voisins à considérer ou tout simplement de la valeur de K reste tout aussi important que le choix de la technique de mesure de la proximité. Selon *(Larose, 2005)* il faut faire un compromis entre un modèle qui minimise et le taux d'erreur et la variance. En d'autres termes, il s'agit de choisir K de telle sorte que la valeur de celui-ci ne soit ni trop grande ni trop petite.

Bien que l'on note qu'une grande valeur de k minimise le taux d'erreur (un taux d'erreur proche de zéro), la variance reste assez instable et peut de ce fait, être préjudiciable à la classification. A l'inverse, une petite valeur de K rend la variable de la prédiction assez stable, avec un taux d'erreur plus ou moins élevé. De même, il faut noter que plus le nombre de voisins choisi est élevé, plus la classification résultante n'est pas généralisable. Ainsi pour trouver la bonne valeur de K qui optimise la qualité de la

classification, il est conseillé de procéder à des expérimentations successives, au cours desquelles on fait évoluer la valeur de K.

Attribuer un poids aux variables

Le talon d'Achille de l'algorithme K-Nearest Neighbors est la pondération des variables (ou dimensions). En effet, les dimensions n'ont pas le même niveau d'importance ; et on pourrait penser que les points plus proches de l'espace ont une plus grande influence que ceux éloignés. De ce fait, les points influenceront à des degrés divers le résultat de la classification. D'où la nécessité d'attribuer un coefficient de pondération ou poids aux différentes dimensions. Une technique courante dans l'attribution des poids est de prendre l'inverse du carré de la distance.

Prédiction/ classification des valeurs de la variable cible

La finalité lorsqu'on utilise l'algorithme du K-Nearest Neighbors est d'arriver à une classification ou une prédiction des valeurs de la variable cible. Selon la nature de la variable à prédire, la méthode de prédiction/classification diffèrera. En effet, pour les variables quantitatives, la valeur à prédire s'obtient en faisant la somme des différentes valeurs pondérées par leurs coefficients respectifs, le tout divisé par la somme des coefficients de pondération. Par contre pour les variables qualitatives, généralement, la valeur de la variable à prédire prend la valeur de l'attribut présentant le plus grand nombre d'observations (le mode statistique).

Algorithme

De façon générale, l'algorithme peut se résumer comme suit :

i. Fixer le nombre de voisins à considérer,
ii. Choisir la fonction de similarité à appliquer
iii. Passer en paramètre les deux jeux de données : celui des données tests et celui des données dont on veut effectuer une prédiction/classification,
iv. Evaluer les distances entre chaque point à prédire/classer et les données test,
v. Classer les distances obtenues par ordre croissant,
vi. Considérer alors uniquement les k plus petites distances,
vii. Attribuer un coefficient de pondération aux distances
viii. Prédire ou classer selon que la variable soit quantitative ou qualitative

Le pseudo algorithme est le suivant :

```
algorithme K_Nearest_neighors(k: entier, fonction_distance: entier,donnee_test:tableau,
donnee_a_predire: tableau)
Debut
        //définition de  deux compteurs qui serviront à parcourir les deux tableau de données
        compteur_1, compteur_2: entier
        //définition du tableau compteur_2 * compteur_1 qui contiendra les calculs de distance
        tableau_distance : tableau_donnees
        Pour compteur allant de 1 à longueur(donnee_test) Faire
                Pour compteur_2 allant de 1 à longueur (donnee_a_predire) Faire

        tableau_distance[compteur_2,compteur_1]:=fonction_distance(donnee_test[compteur_1],d
onnee_a_predire[compteur 2])
                FinPour
        FinPour
        //on classse les éléments du tableau par ordre croissant
        classer_element_tableau(tableau_distance, ordre_croissant)
        //on ne considère alors qu'un nombre k de voisins
        recuperer_nombre_fixe_voisin(tableau_distance, k)
        //on attribut un poids aux distances obtenues afin de tenir compte de l'influence de
certaines variables
        attribuer_coefficient_ponderation(tableau_distance,poids)
        //la classification ou la prédictiction est fonction du type de variable
        determiner_valeur_a_predire(donnee_test,tableau_distance, type_variable)
Fin
```

Forces et faiblesses

A l'instar d'autres algorithmes de fouille de données, K-Nearest Neighbors comporte des forces et faiblesses. Comme mentionné plus haut, l'avantage majeur de l'algorithme réside dans sa simplicité et sa facilité d'implémentation. Également, l'algorithme est robuste quant au traitement des données anormales surtout lorsque la classification est basée sur une pondération de la distance *(Tekmono, 2006)*. Malheureusement, l'algorithme est sujet à certains inconvénients dont :

- la dégradation des performances lorsque la dimension s'accroit ;
- un temps de calcul plus élevé à cause de l'évaluation des indices de similarité/dissimilarité entre tous les points de l'espace p-dimensionnel ;
- l'impossibilité de produire un modèle pour décrire les futures données. Cela est intrinsèquement lié à la nature « Instance Based Learning » de l'algorithme.

B.3. K-Means

Principe

K-Means est une technique de clustering développée par J.MacQueen et mise œuvre dans sa forme actuelle par E.Forgy. Pour rappel, le clustering est une technique de fouille de données qui permet de classer un ensemble d'objets en groupes de sorte que la similarité intra-groupe soit maximale et celle inter-groupe minimale *(Larose, 2005).*

K-Means est un algorithme simple et efficace qui permet de partitionner un ensemble d'objets d'un espace à p dimensions en un nombre fini K de clusters sur la base de la minimisation d'une fonction-objectif *(Matteo, 2008)*. Cette fonction-objectif, bien que fonction de la variante de l'algorithme K-Means, est plus souvent définie comme le carré des distances séparant les objets des centroïdes. En effet les données au niveau de K-Means sont considérées comme des points d'un espace p-dimensionnel dans lequel le centroïde est considéré comme le point le plus centralement localisé – par rapport à un groupe d'objets.

Le clustering à l'aide de K-Means requiert de l'utilisateur, la fixation du nombre de clusters avant la mise en route ; ce qui sonne comme un inconvénient à cet algorithme *(Hamerly, et al., 2003)*. De ce point de vue, on peut dire que K-Means et K-Nearest Neighbors se ressemblent (le choix de k au départ). Mais la comparaison s'arrête là. Autant K-Nearest Neighbors ne permet pas de partitionner des données en groupes, autant K-Means ne permet pas de faire de la prédiction/classification.

Mais à la différence de K-Nearest Neighbors, K-Means fonctionne sur la base d'un seul jeu de données en entrée avec pour objectif la production d'un nombre défini de groupes de sorte à ce que les données à l'intérieur d'un groupe soient similaires. Tout comme, au niveau de la technique des proches voisins, la mesure de similarité peut varier selon le contexte dans lequel s'effectue la fouille de données ; qu'il s'agisse de données quantitatives ou qualitatives ou de la préférence d'une mesure de calcul de distance par rapport à un autre. Le choix de la mesure de similarité permet ainsi donc, contrairement à certaines idées reçues, l'application de K-Means sur des données qualitatives.

À l'image de K Nearest Neighbors, l'application de K-Means soulève quelques questions :

- Comment choisir le nombre de clusters à former,

- Quelle mesure de similarité choisir,
- Comment réduire l'arbitraire dans le choix des clusters de départ,
- Comment évaluer la qualité des clusters formés.

Fixer le nombre de cluster

L'algorithme de K-Means, nécessite comme paramètre de départ, le choix du nombre de clusters à former. Le choix, loin d'être arbitraire, devrait se faire sur la base d'une connaissance des données. D'où la nécessité qu'il y'ait un expert lors de la fouille afin de servir de guide dans la fixation du nombre de groupes. Ce nombre est déterminant dans la qualité des résultats fournis par l'algorithme et son choix peut ne pas paraître évident même pour un expert du domaine. Des variantes de K-Means, ont été mises en œuvre afin de palier à cet état de fait (voir G-Means[59]).

Choisir des clusters de départ

K-Means est un algorithme itératif qui au fil de chaque itération permet de regrouper les données autours d'un élément central, le centroïde. Le problème majeur dans cet algorithme réside dans la première itération, au cours de laquelle, l'utilisateur est obligé de définir de façon arbitraire les K premiers clusters. Cela est d'autant plus problématique que, de ce choix, dépend le résultat de la fouille de données. En effet, comme le note *(Matteo, 2008)*, K-Means est sensible à l'ordre de définition des premiers clusters. De ce fait, le choix d'un point au détriment d'un autre peut avoir beaucoup d'impact sur la qualité des clusters formés.

Pour remédier à cette situation, *(Larose, 2005)* propose de faire plusieurs expérimentations basées sur le choix de centroïdes différents à chaque fois et ceci afin de déterminer de façon optimale, les meilleurs clusters. *(Likas, et al., 2001)* proposent plutôt un algorithme dénommé Global K-Means. Cet algorithme utilise K-Means pour la recherche des minimums locaux et effectue par la suite une itération afin de stabiliser la formation des groupes.

Evaluer la qualité des clusters formés

[59] Variante de K-Means utilisant une distribution gaussienne pour la détermination du nombre de clusters

L'évaluation de la qualité des clusters reste tout aussi problématique que la détermination du nombre de clusters. En effet, ce problème est inhérent à l'algorithme K-Means qui ne détermine que des minimums locaux et pas globaux. En effet, l'algorithme minimise la somme des carrés des distances séparant les points de l'espace aux centroïdes. L'algorithme s'arrête des que cette somme est stable. Ce qui ne garanti pas qu'on est atteint un minimum optimal. Parce qu'en relaçant l'algorithme et en initialisant avec des centroïdes différents, il est probable que le minimum obtenu soit plus petit que celui obtenu précédemment. Ceci dénote que la qualité des clusters formés est intimement liée à l'ordre d'initialisation des centroïdes.

Algorithme

L'algorithme de K-Means se résume comme suit :

i. Choisir le nombre k de clusters désirés
ii. Choisir arbitrairement dans l'ensemble des données un nombre k de centroïdes
iii. Evaluer la distance entre les autres points de l'espace et les centroïdes
iv. Assigner un point dans le groupe d'un centroïde si la distance les séparant est minimal comparée aux autres.
v. Calculer les nouveaux centroïdes des groupes formés
vi. Aller a iii et répéter les opérations jusqu'à ce que les mouvements des points soient stables (i.e. que les points ne changent plus d'appartenance).

```
algorithme simple_K_Means(k: entier, fonction_objectif: fonction,donnee_test:tableau)
Debut
        //définition de  deux compteurs qui serviront à parcourir les deux tableau de données

        compteur_1, compteur_2: entier
        //définition du tableau compteur_2 * compteur_1 qui contiendra les calculs de
distance
        tableau_distance : tableau_donnees
        tableau_centroïde: tableau_données //tableau contenant les centroïdes
        tableau_cluster: tableau_donnees //tableau des clusters formés
        //fonction d'initialisation des k centroïdes
        tableau_centroïde:=initialiser_K_premier_centroide(k);
        //parcours de l'ensemble des données et computation de la distance
        label computation:pour compteur_1 allant de 1 à k faire
                Pour compteur_2 allant de 1 à longueur(donnee_test) Faire

        tableau_distance[compteur_2,compteur_1]:=fonction_objectif(tableau_centroïde[comp
teur_1],donnee_test[compteur_2])
                FinPour
        FinPour
        //on classse les éléments du tableau par ordre croissant et par centroïde
        classer_element_tableau(tableau_distance, ordre_croissant)
        //affection de chaque point à un cluster donné
        tableau_cluster:=regroupement_par_cluster(tableau_distance, tableau_centroïde)
        //computation des nouveaux cluster
        redefinition_cluster(tabeau_centroïde);
        //repartir au niveau du label computation si fonction objectif
        si evaluation_stabilite_cluster_former(tableau_cluster)=faux alors
                aller_a_label(computation)
        fin si
Fin
```

Forces et faiblesses

Comme le note *(Kanungo, et al., 2000) (Larose, 2005) (Matteo, 2008)*, K-Means est une technique de clustering très populaire dont la principale force réside dans sa simplicité. En effet, il est l'une des techniques de clustering les plus utilisées et son algorithme est simple d'implémentation. De plus, il a prouvé son efficacité dans la production de clusters de bonne qualité dans bien des domaines *(Alsabti, et al., 1998)*.

Malgré sa popularité, le K-Means souffre d'un certain nombre d'inconvénients que nous avons évoqués plus haut. Il s'agit notamment de:

- La dépendance à l'ordre d'initialisation des premiers centroïdes,
- La nécessité de définir un nombre de clusters au départ,

- La qualité des clusters obtenus,
- La consommation de ressource temps lors du clustering.

Des variantes de K-Means ont été mises en œuvre afin de remédier à ces inconvénients. Certains à l'image de KH-Means *(Zhang, et al., 1999)*, en lieu et place de minimiser la somme des carrés des distances, se base plutôt la moyenne des distances harmoniques. Ce qui lui permet d'être d'une part insensible à l'ordre d'initialisation des premiers centroïdes et d'autres parts, de produit des clusters de bonne qualité. La variante mise en œuvre par *(Faber, 1994)* dénommé Continuous K-Means permet une utilisation efficiente en n'évaluant pas la distance entre chaque point de l'espace et son centroïde à la différence du K-Means classique. Bien d'autres variantes existent ; chacune tentant à sa façon de remédier aux insuffisances de l'algorithme initiale (voir X-Means, K-Médian, K-Médoïds,…).

B.4. Conclusion

Le monde de la fouille de données dispose de plusieurs algorithmes appartenant à des diverses catégories chacune réalisant une fonction particulière. Dans ce document, nous avons décrit deux(2) algorithmes appartenant à des catégories différentes : classification et clustering. Le choix de ces algorithmes notamment K-Nearest Neighbors et K-Means s'est fondé sur plusieurs critères déterminants dont :

- L'adaptabilité de ces algorithmes dans un contexte spatial : en effet, ces algorithmes utilisent la notion de voisinage dans leur mode opératoire (tâche de classification/prédiction ou clustering) ;
- La simplicité et la facilité de mise en œuvre ;
- Popularité dans le domaine de la fouille de données : ces algorithmes ce sont vus implémentés dans plusieurs outils dédiés à la fouille de données.

Certes, ces algorithmes comportent leur part de forces et de faiblesses, mais possèdent un potentiel d'adaptabilité facile quant à l'intégration de l'aspect spatial.

Annexe C : Mesures des similarités descriptives

Les mesures de similarité permettent de décrire dans quelle mesure deux objets quelconques peuvent être similaires selon un certain point de vue. Il faut noter que ce point de vue peut être assez varié. À titre d'exemple, on peut vouloir établir une similarité entre deux(2) objets selon la couleur, la taille, la forme, l'orientation, la proximité. Selon le point de vue, on pourra être amené à utiliser différentes de mesures de similarité.

Les mesures de similarité sont des fonctions mathématiques permettent de capturer le degré de ressemblance entre deux(2) objets quelconques. Toutefois, la similarité est une fonction difficile à évaluer d'où l'utilisation de mesures de dissimilarité communément appelées distance. Néanmoins, sous réserve de certaine normalisation, on peut passer aisément d'une mesure de dissimilarité à une mesure de similarité et inversement (voir Équations ci-dessous).

Soient i et j deux(2) objets quelconques de l'espace. Si on note par $\delta_{i,j}$ et $s_{i,j}$ les valeurs de dissimilarité normalisées (i.e. ayant une valeur comprise entre [0,1]) et similarité - respectivement entre les deux objets ; la relation entre la similarité et la dissimilarité est donnée par :

$$s_{i,j} = 1 - \delta_{i,j}$$

Pour une similarité comprise entre [0,1]. Ainsi, lorsque les deux objets sont exactement similaire (similarité à 1), la dissimilarité est à zéro

Lorsque l'on désire une similarité comprise entre [-1,1], la formule devient :

$$s_{i,j} = 1 - 2\delta_{i,j}$$

Ainsi, si les objets sont complètement différents (dissimilarité à 1), la similarité est à -1. Et si la dissimilarité est à 0, la similarité est de 1.

La dissimilarité ou distance, à l'inverse de la mesure de similarité, est une fonction permettant de capturer le degré de différence entre deux objets. D'un point de vue mathématique, une mesure de dissimilarité est une application à valeur dans R+ qui possède certaines propriétés. On note les caractéristiques suivantes[60] :

1. Positivité : la distance est toujours positive. Il s'agit d'une application donnée par :

[60] http://fr.wikipedia.org/wiki/Distance_%28math%C3%A9matiques%29

$$d: E \times E \rightarrow \mathbb{R}^{+} \quad \text{(B.4-1)}$$

2. Séparation : une distance zéro entre deux points signifie qu'il s'agit des mêmes points. Soient x et y deux éléments d'un espace vectoriel tels que

$$\forall x, y \in E, d(x, y) = 0 \Rightarrow x = y \quad \text{(B.4-2)}$$

3. Symétrie : la distance entre deux points est toujours pareille qu'on parte de l'un ou l'autre des points.

$$\forall x, y \in E, d(x, y) = d(y, x) \quad \text{(B.4-3)}$$

4. Inégalité triangulaire : on ne réduit pas la distance entre deux points en passant par un troisième. Mais cette dernière caractéristique n'est valable que pour les distances métriques.

$$\forall x, y, z \in E, d(x, z \leq d(x, y) + d(y, z) \quad \text{(B.4-4)}$$

Comparées aux données classiques, les données géo-spatiales sont d'une certaine complexité en ce sens qu'elles sont porteuses de deux composantes d'information : une descriptive et l'autre géométrique. Sur ces deux types de composantes, comme on peut s'en douter, on ne peut appliquer les mêmes mesures de similarité. D'où la nécessité pour chacune de ces composantes, d'appliquer la mesure de similarité qui sied le mieux.

5. Mesures de similarité applicable à la composante descriptive

Nous nommons données descriptives, les données de types classiques : données quantitatives, chaines de caractères, booléens, données ordinales. Selon le type de données traitées, on utilise différentes mesures de dissimilarité. Comme on pourra le noter par la suite, certaines mesures peuvent être utilisées sur différents types de données.

o Mesures de similarité applicables aux variables quantitatives

Par attribut de type quantitatif, il faut entendre les attributs ayant une valeur numérique. À titre d'exemple, le nombre de personnes d'un ménage, la longueur d'un fleuve, le chiffre d'affaire annuel d'une entreprise sont autant de valeurs quantitatives. On note plusieurs mesures de similarité applicable aux valeurs de ce type.

- **Distance de Minkowski**

Il s'agit d'une généralisation de plusieurs distances dont les distances euclidienne, de Manhattan de Tchebychev (voir sections suivantes). Il s'agit d'une distance applicable autant sur les variables ordinales que celle quantitatives. Sa formule est donnée par:

$$d_{ij} = \sqrt[\lambda]{\sum_{k}^{n} |x_{ik} - x_{jk}|^{\lambda}}$$

Ainsi lorsque

- $\lambda = 1$ on obtient une distance de Manhattan
- $\lambda = 2$ on obtient une distance euclidienne
- $\lambda = \infty$ on obtient une distance de Tchebychev

- **Distance Euclidienne**

Il s'agit de la mesure de distance la plus couramment utilisée. Elle est déterminée en calculant la racine carrée de la somme du carré de la différence des coordonnées des points d'un espace à N-dimensions. Comparée à la distance de Manhattan, la distance euclidienne est la ligne droite qui relie deux(2) points de l'espace. Dans la majorité des opérations de clustering, pour des raisons d'efficacité, on préfère utiliser le carré de la distance euclidienne plutôt que la racine carrée. Sa formule est donnée par :

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$$

- **Distance de Manhattan ou city block distance**

Cette mesure considère l'espace comme une grille quadrillée. En lieu et place d'évaluer la diagonale entre une paire de points, elle évalue le nombre de pas de grille à parcourir pour aller d'un point à l'autre. Il s'agit d'un cas particulier de la distance euclidienne avec un paramètre lambda égale à 1(voir section précédente – distance de Minkowski). On calcule cette distance en sommant la valeur absolue de la différence des coordonnées entre deux pairs d'objets. Sa formule est donnée par :

$$d_{ij} = \sum_{k=1}^{n} |x_{ik} - x_{jk}|$$

- **Distance de Bray Curtis**

Il s'agit d'une mesure de similarité utilisée principalement dans le domaine des sciences environnementales. Tout comme la distance de Manhattan, l'espace est considéré comme une grille. Sa formule es donnée par :

$$d_{ij} = \frac{\sum_{k=1}^{n} |x_{ik} - x_{jk}|}{\sum_{k=1}^{n} x_{ik} + \sum_{k=1}^{n} x_{jk}}$$

- **Distance de Tchebychev**

Cette mesure évalue la distance maximale séparant une paire de points. Elle peut s'appliquer autant sur des données quantitatives qu'ordinales. Elle est déterminée en prenant le maximum de la valeur absolue de la différence des coordonnées de la paire de points en considération. Sa formule est donnée par :

$$d = Max|x_i - x_j|$$

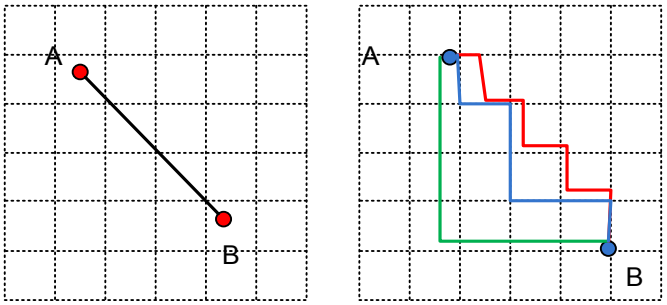


Figure 1: Distance euclidienne (gauche) versus distance de Manhattan (droite)

o Mesures de similarité applicables aux variables binaires

Les variables binaires réfèrent à des données qui prennent pour valeur vrai/faux, oui/non, 0/1 ou positif/négatif. La mesure de la similarité entre données binaires prend tout son intérêt lorsque ces données sont des vecteurs, c'est-à-dire des données comportant plusieurs éléments de type binaire. On note que la mesure de similarité entre des vecteurs binaires revient tout simplement en une mesure de fréquence.

Pour illustrer les mesures de distances dont nous parlerons dans les paragraphes suivants, prenons l'exemple de deux vecteurs V_1 et V_2 binaires tels que : V_1 (1, 1, 1,1) et V_2 (0, 1, 0,0)

Soient :

- p : le nombre d'éléments positifs (1 ou oui ou vrai) dans les deux vecteurs
- q : le nombre d'éléments positifs pour le $i^{ième}$ élément du vecteur V1 et négatif pour V2
- r : le nombre d'éléments négatifs pour le $i^{ième}$ élément du vecteur V1 et positifs pour le V2
- s: le nombre d'éléments négatifs pour les deux vecteurs

p, q, r, s peuvent être représentés schématiquement par :

	Positif	**Négatif**
Positif	p	q
Négatif	r	s

Pour faire le parallèle avec nos deux vecteurs, on a : p=1, q=3, r=0, s=0.

- **La distance de Hamming**

Il s'agit d'une mesure utilisée beaucoup dans le monde informatique ou des télécommunications – surtout pour la correction d'erreurs. Cette distance évalue le nombre de bits ou d'éléments différents entre deux séquences de vecteurs selon une comparaison bit à bit. Considérant que nos deux vecteurs ci-dessus (V_1 et V_2), représente deux objets i et j, la distance de Hamming est donnée par :

$$d_{ij} = q + r$$

- **Le simple Matching distance**

Il s'agit d'une variante de la distance de Hamming. Elle se calcule en divisant le nombre de bits différents (distance de Hamming) par le nombre total de bits. Sa formule est donnée par :

$$d_{ij} = \frac{q + r}{p + q + r + s}$$

- **La distance de Jaccard**

La distance de Jaccard est donnée par la formule suivante:

$$d_{ij} = \frac{q + r}{p + q + r}$$

- **Mesures de similarité applicables aux variables de type nominale**

Pour les données de ce type, on peut procéder à un prétraitement des données avant de leur appliquer une mesure de dissimilarité du genre celles utilisables sur les données binaires.

○ Mesures de similarité applicables aux variables de type ordinale

Les variables ordinales réfèrent à des données de type qualitatif avec cependant une hiérarchie entre elles. À titre d'exemple, supposons une variable décrivant l'état d'une chaussée. Cette variable pourra éventuellement prendre les valeurs : très bon, bon, moyen, dégradé. On constate que même si ces valeurs sont de types chaines de caractère (nominal), on peut déduire un certain ordre hiérarchique entre elles. On note également pour ces types de données, diverses mesures de dissimilarité. Pour cette catégorie de données, on peut également appliquer des mesures de dissimilarité telles celle de Tchebychev, de Minkowski ou de Hamming. On note comme mesures de distances :

- **La distance de Kendall**

La distance de Kendall permet d'évaluer le « désordre » entre un vecteur quelconque vis-à-vis d'un vecteur de référence (il s'agit de vecteurs de données ordinales). De façon pratique, elle consiste à évaluer le nombre de permutation minimum qu'il faut effectuer pour passer d'un vecteur non ordonné B à un vecteur de référence A en considérant toutefois (dans le vecteur B) des paires d'éléments adjacents. Pour illustration, considérons deux vecteurs de données ordinales A et B représentant les moyens de locomotion de deux(2) individus quelconques. A et B sont tels que : A = [Taxi, Metro, Bus], B = [Metro, Taxi, Bus]. Si on considère A, le vecteur de référence, la distance de Kendall entre A et B est d'une unité (1). En effet, pour transformer B en A, il faut juste inter changé « Metro » et « Taxi ». Ce qui revient à effectuer une opération (voir figure ci-après).

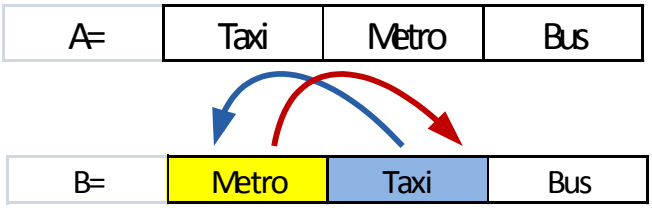


Figure 2: Distance de Kendal entre deux vecteurs ordinaux

Une variante de cette mesure dénommée la distance de Cayley, permet elle de considérer toute paire d'éléments non nécessairement adjacents (voir Figure).

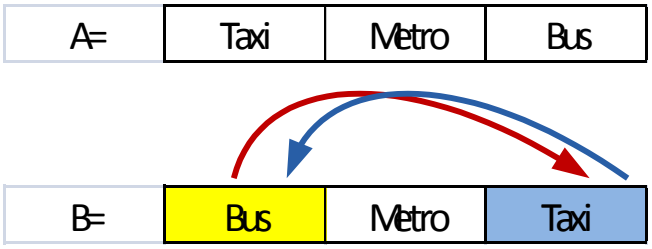


Figure 3: Distance de Cayley entre deux vecteurs ordinaux

On note qu'avec la distance de Kendall, il n'aurait pas été possible de permuter directement les éléments « Bus » et « Taxi » parce qu'il faudrait alors que ces deux(2) éléments soient adjacents.

- **La distance de Spearman**

Il s'agit du carré de la distance euclidienne. Sa formule est donnée par :

$$\mathbf{d_{ij}} = \sum_{\mathbf{k=1}}^{\mathbf{n}} \left(\mathbf{x_{ik}} - \mathbf{x_{jk}}\right)^{\mathbf{2}}$$

Annexe D – Article scientifique : vers une nouvelle approche de prise en compte de la composante spatiale dans le processus de prise de fouille de données

Mamadou OUATTARA, Thierry BADARD
Université Laval - Faculté de Foresterie, de Géographie et de Géomatique - Département des Sciences Géomatiques – Groupe de recherche GEOSOA
E-mail : mamadar.ouattara.1@ulaval.ca , thierry.badard@scg.ulaval.ca

Résumé : *le stockage de plus en plus massif de l'information géo-spatiale, a suscité le besoin d'apprendre de ces données. C'est en cela qu'est née la fouille de données spatiales qui consiste en l'extraction de connaissances utiles au sein de ces données afin de servir de support au processus de prise de décision. À ce jour, différentes approches existent pour guider le processus de fouille ; mais elles ont toutefois montré certaines limites. Dans ce document, nous proposons une nouvelle approche consistant en une intégration efficiente de la composante spatiale dans des outils de fouille de données traditionnelles. La particularité de cette approche réside dans le traitement dynamique de la composante spatiale à travers la prise en compte de tous types de géométries et de relations; et cela au niveau de toutes les étapes du processus de fouille de données. Cette nouvelle approche permet d'effectuer des tâches de géo-clustering ou géo-classification basée sur l'utilisation de mesures de similarité entre entités géo-spatiales fondées sur les relations métriques, topologiques, directionnelles et de reconnaissance de formes. Cette approche a été pratiquement implémentée en procédant à l'enrichissement d'une bibliothèque open source de fouille de données, KNIME.*

Mots-clés : fouille de données spatiale, datamining spatiale, Geographic Knowledge Discovery, mesure de similarité, relations spatiales, Geo-BI.

Introduction

À la faveur de l'évolution des technologies liées à l'acquisition, au traitement, à la production et au stockage de l'information spatiale, l'intérêt de la société pour ce type d'information est allé de plus en plus croissant au fil des ans (en témoigne les nombreux usages). Avec l'énorme quantité de données géo-spatiales stockées est apparu un besoin nouveau, celui d'apprendre de ces données afin de prendre des décisions utiles pour l'entreprise. C'est de cela qu'est née une branche nouvelle, la fouille de données géo-spatiales également dénommé datamining spatial ou GKD[61].

Le GKD est une sous branche du KDD[62] qui a pour objet la découverte de connaissances implicites et potentiellement utiles au sein de vaste ensemble de données géo-spatiales (Miller, et al., 2001). Comparé au KDD, le GKD est un domaine relativement jeune qui est née d'une part du besoin de trouver des connaissances

[61] Geographic Knowledge Discovery

[62] Knowledge Discovery in Databases

dissimulées dans les vastes référentiels de données géo-spatiales et d'autre part de l'inadéquation des techniques de fouille de données « traditionnelles » dans le traitement des données géo-spatiales.

L'inadéquation des techniques traditionnelles de fouille de données a conduit à la mise en œuvre principalement de deux approches permettant de prendre en compte la dimension spatiale de l'information :

- une première dite de prétraitement qui consiste en l'extraction explicite des relations spatiales existantes entre objets ;
- une seconde dite spatial-centric consistant au développement d'algorithmes qui tiennent « nativement » compte de la spécificité des données géo-spatiales.

Ces approches ayant toutefois montré leurs limites, il est nécessaire de voir la fouille de données sous un angle nouveau; celui de l'intégration efficace et transparente de la composante spatiale au sein d'outils de fouille de données existants. Dans le présent document, nous nous attachons à décrire cette nouvelle approche de fouille de données spatiales d'exploitation de la composante géo-spatiale des données qui tient compte de la spécificité de l'information géographique. Le document est structuré comme suit, nous abordons premièrement les précédentes approches de fouille de données en exposant leurs avantages et inconvénients. Par la suite, nous décrivons notre approche. Enfin, nous décrivons, l'implantation des mesures de similarité géo-spatiales ainsi que d'autres fonctionnalités au sein d'un outil open source de fouille de données.

De l'inadéquation des approches précédentes de fouille de données spatiales

Pour l'heure, il existe deux principales approches de fouille de données spatiales (Rinzivillo, et al., 2008). Ces approches sont nées de la difficulté d'utiliser les outils de fouille de données traditionnelle afin de traiter convenablement l'information géo-spatiale. Cette incapacité résulte des caractéristiques propres à cette donnée (Koperski, et al., 1996) (shekhar, et al., 2003) (Ester, et al., 1997) (Miller, et al., 2001):

- dépendance spatiale et hétérogénéité,
- diversité des types de données,
- complexité des objets spatio-temporels

Approche préconisant le prétraitement des données spatiales

La première de ces approches préconise une réutilisation des outils existants de fouille de données. Mais avant l'utilisation effective de ces outils (algorithmes et techniques), il faut en premier procéder à un prétraitement de la donnée géo-spatiale (voir Figure B.4-1). Ce prétraitement vise principalement l'extraction des corrélations spatiales et leur stockage sous la forme d'attributs de type classiques (chaine de caractère, type numérique, type booléen, etc.) *(Bogorny, et al., 2005)* (Rinzivillo, et al., 2008).

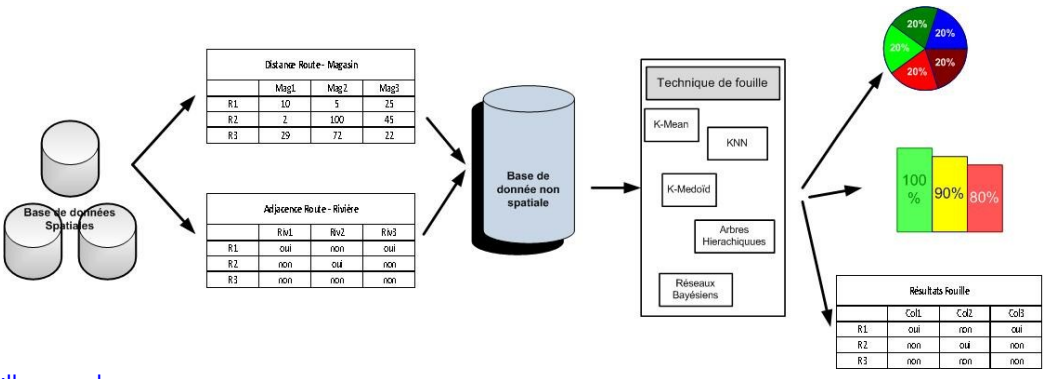


Figure B.4-1: Approche de prétraitement

Cette approche est assez pratique en ce sens qu'elle met à contribution les outils existants de fouille de données « traditionnelle ». On note toutefois qu'elle comporte un certain nombre de désavantages dont :

- un temps de computation énorme,
- une redondance dans le stockage,
- une mise à jour difficile en cas d'ajout, modification ou suppression,
- une impossibilité d'utiliser les données pour la visualisation cartographique,
- une extraction de relations spatiales trop évidentes

Approche préconisant le développement d'algorithmes spatiaux

Cette approche préconise en lieu et place du prétraitement de la composante spatiale, le traitement dynamique de la composante spatiale à travers la mise en œuvre d' algorithmes et/ou d'outils.

Au regard des limitations de la première approche (temps de computation élevé, extraction de relations non pertinentes), plusieurs méthodes et outils allant dans le sens de cette approche ont été mis en œuvre (Malerba, et al., 2002) (May, et al., 2003) (Klösgen, et al., 2002) (Han, et al., 1997) (Malerba, et al., 2000). En termes de comparaison, cette approche offre bien plus d'avantages que la première. On note comme avantages :

- la flexibilité ;
- la sélection dynamique des relations lors de la fouille ;
- la réduction des jointures spatiales et donc une diminution des ressources exigées pour la computation.

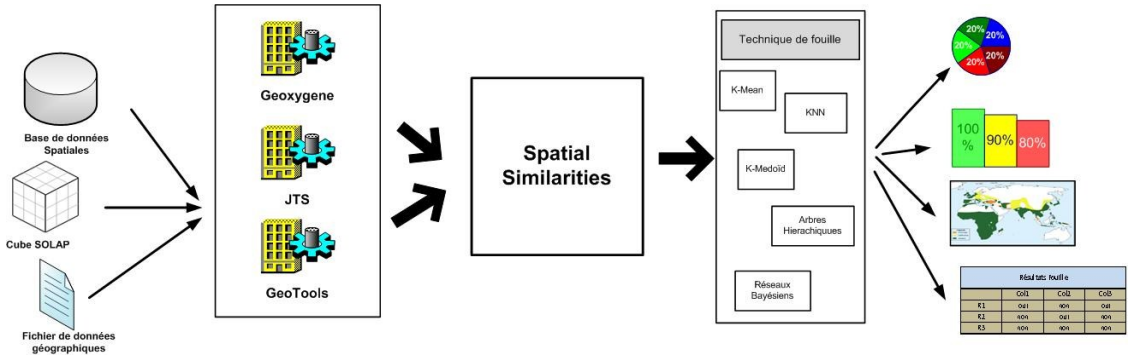


Figure B.4-2: Approche de fouille de données utilisant des mesures de similarité géo-spatiales

Vers une nouvelle approche

Face aux inconvénients des précédentes approches :non prise en compte de tous types de géométries, et de relations spatiales, non disponibilité de la composante spatiale à toutes les étapes de la fouille de données, il est utile et certainement pratique d'adopter une approche qui va au-delà des désavantages des approches précédentes d'une part, et d'autre part met à profit l'existant en matière de fouille de données « traditionnelle » dans la mesure où elle est assez mature. C'est dans ce sens que s'inscrit l'approche que nous proposons.

Loin de constituer, tout comme dans la première approche de fouille de données, en une simple extraction et exposition des relations spatiales entre objets sous forme d'attributs de type classique (numérique, chaine de caractères, booléen), notre approche sonne comme un compromis entre les deux (2) approches précédemment citées en ce sens que nous proposons une approche traitant dynamiquement la composante spatiale et mettant à profit les outils existant de datamining.

Comme avantages, l'approche proposée, à la différence des précédentes, permet le traitement de tous types de géométries (point, ligne, polygone), de relations à toutes les étapes du processus de fouille de données (cf. le modèle CRISDM). Notons que ces traitements sont effectués dans le respect de la première loi de la géographie

(Bogorny, et al., 2005) (Chawla, et al., 2001) qui stipule en substance que les entités géo-spatiales sont inter reliées et sont beaucoup plus influencées par leur voisinage..

Géo-clustering ou l'utilisation de mesures de similarité entre entités géo-spatiales

L'approche proposée, une fois implémentée permettra de réalisations diverses tâches de fouille de données spatiales dont la géo-classification, le géo-clustering ainsi que la construction d'arbres de décisions spatiaux; pour ne citer que celles là. Pour ce qui concerne les tâches de géo-clustering et de géo-

classification (basée sur les K plus proche voisins), il est nécessaire d'utiliser une mesure de similarité sur les données géo-spatiales. Ces mesures sont principalement basées sur les relations spatiales entretenues par les entités géographiques.

Les mesures de similarité[63] géo-spatiales constituent le cœur de notre approche et au-delà de toutes approches désirant offrir des fonctionnalités de géo-clustering et géo-classification en ce sens qu'elles permettent de capturer les relations de voisinage qui font toute la spécificité de l'information géographique. Toutefois, du fait de la complexité de ce type de données - à savoir l'existence d'une composante descriptive et géométrique - on ne peut appliquer sur ce type de données les mêmes mesures de similarité. D'où la nécessité pour chacune de ces composantes, d'appliquer la mesure de similarité qui sied le mieux. Dans le présent article, nous nous intéressons particulièrement aux mesures de similarité applicable à la composante géométrique (les mesures de similarité applicables sur la composante descriptive réfèrent aux mesures rencontrées le plus souvent dans la littérature ; distance euclidienne, distance de Manhattan, distance de Hamming, etc.).

Les mesures de similarité géo-spatiales sont basées sur les principales relations entretenues par les entités géo-spatiales. Ces relations sont généralement de trois (3) types (métriques, topologiques et directionnels) ; mais peuvent être étendues à d'autres types de relations dont celle en rapport avec la reconnaissance de formes.

Similarité basée sur les relations métriques

Les entités géo-spatiales, on le sait, sont beaucoup influencées par leur voisinage. L'idée au niveau des mesures de similarité ci-dessus nommées, est d'évaluer la proximité entre deux entités géo-spatiales sous l'angle de la distance « réelle » les séparant. Il pourra s'agir à titre d'exemple de la proximité entre divers foyers de propagation d'un phénomène. À ce niveau, on pourra utiliser une mesure quantitative ou qualitative *(Bogorny, et al., 2005) (Hernandez, et al., 1995)*. Les mesures quantitatives reviennent à l'utilisation des différentes déclinaisons de la fonction de distance (buffer, distance) en tenant compte du type d'entités géo-spatiales alors considérées.

Les mesures qualitatives consistent en l'utilisation du langage naturel à travers des expressions du type « très proche de », « proche de », « loin de ». Cela nécessite d'associer préalablement une distance quantitative à ces expressions.

Similarité basée sur les relations topologiques

Au nombre des relations entretenues par les entités géo-spatiales, la relation topologique occupe une place de choix. Pour rappel, il s'agit d'une relation binaire entre deux(2) entités géo-spatiales

[63] Voir la définition mathématique de mesure de similarité/dissimilarité dans (Tekmono, 2006)

lorsqu'on prend en considération leurs intérieurs, limites et extérieurs. Tout comme au niveau des mesures de similarité basées sur les relations métriques, celles basées sur les relations topologiques peuvent également être de nature quantitative ou qualitative.

L'utilisation de relations topologiques qualitatives comme mesures de similarité revient à l'exploitation de la matrice à 9-intersection *(Egenhofer, 1998)*.

Afin de tirer parti des mesures de similarité basées sur les relations topologiques qualitative, il est important de leur associer d'autres mesures de similarité en l'occurrence celles applicables aux variables de type binaires. D'un point de vue pratique, il s'agit dans un premier temps d'évaluer la relation topologique entre les entités considérées et par la suite d'utiliser une mesure de similarité descriptive (distance de Kendall, distance de Hamming, distance de Cayley, etc.) sur le résultat obtenu.

La réponse qualitative, dépendamment du contexte, peut être suffisante ou non. Toutefois dans certains cas, il est utile d'avoir un ordre de grandeur en ce qui concerne une relation topologique entre entités géo-spatiales. C'est en ce sens que les relations topologiques quantitatives introduites par *(Nedas, et al., 2005) (Shariff, et al., 1998)* prennent tout leur sens. En réalité, il s'agit d'une extension de la matrice à 9-intersection en vue de délivrer des mesures topologiques plus détaillées. Toutefois ces « détails » topologiques ne s'appliquent pas à tous les cas de configurations topologiques ni à tous les types d'entités géo-spatiales (dimension >1).

Similarité basée sur la similitude de formes

En plus de procéder à une fouille de données sur la base des relations usuelles – relations topologiques, métriques, directionnelles - on peut effectuer une fouille de données sur la base d'une similarité de formes. Ces mesures de similarité sont pour la plupart utilisées dans le domaine du traitement et de l'analyse d'images, celui de l'évaluation de la qualité géométrique et par extension le domaine de l'appariement de données géométriques.

Il existe diverses mesures permettant d'évaluer le degré de similarité entre deux formes géo-spatiales selon l'espace de représentation des éléments géo-spatiaux. On note comme mesures : la distance de Hausdorff, la distance de Fréchet, la distance surfacique, etc.

La distance de Hausdorff et celle de Fréchet sont des mesures de dissimilarité utilisées pour quantifier la dissemblance entre des contours d'entités géo-spatiales. En d'autres termes, ces mesures donnent de meilleures résultats lorsque appliquées sur des entités linéaires. On note toutefois que la distance de Fréchet donne de meilleurs résultats comparée à celle de Hausdorff car elle capture la dissemblance entre deux contours orientés *(Eiter, et al., 1994) (Devogele, 1997) (Bel Hadj ali, 2001) (Buchin, et al., 2006)*.

Pour pallier à la limitation des deux mesures de distance décrites plus haut (distance de Hausdorff, distance de Fréchet), *(Vauglin, 1997)* propose la distance surfacique. En effet, cette distance consiste en une évaluation de la proportion de surface commune entre deux(2) entités géo-spatiales.

Similarité basée sur les relations directionnelles

Les relations directionnelles peuvent jouer un rôle prépondérant dans la fouille de données en tant que mesure de. À ce titre, prenons l'exemple d'entités géo-spatiales « zones de tension » mutuellement disjointes au sein desquelles on s'intéresse à la progression d'un phénomène quelconque afin de pouvoir regrouper (clustering) ces dites entités selon une ou des caractéristiques communes. En se contentant de caractériser les rapports topologiques que ces entités entretiennent entre elles, on notera qu'elles sont similaires. De ce fait, on risque de retrouver ces objets au sein d'un seul groupe ou cluster. Par contre, en associant une autre mesure de similarité en l'occurrence celle basée sur les relations directionnelles, le résultat sera différent. On caractérise mieux la similarité qui parait entre ces entités.

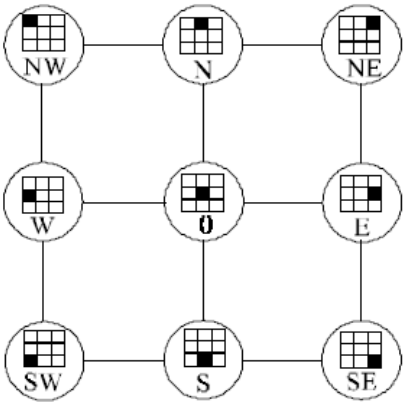


Figure B.4-3 : Graphe de voisinage conceptuel (tiré de *(Goyal, 2000)*)

Cependant, construire une mesure de similarité sur la base des relations directionnelles n'est pas aussi évident que celle basée sur les relations métriques. En effet, considérant deux entités géo-spatiales respectivement positionnées au Sud et Nord-Est d'une entité de référence, sur quelle base peut on dire laquelle est plus proche de l'entité de référence ? Toutefois, on peut mettre à contribution certains travaux en matière de distances directionnelles *(Goyal, 2000) (Goyal, et al., 2001)*.Cette mesure de distance se fonde sur un graphe de voisinage conceptuel(voir Figure B.4-3) basé sur les neuf(9) « quadrants directionnels ». Ce graphe fourni la distance comme le coût de passage d'un quadrant directionnel à l'autre. Ainsi, pour une entité cible quittant un « quadrant » vers un autre, le chemin suit l'adjacence des « quadrants » (verticalement ou horizontalement) en privilégiant celui qui minimise la distance totale. À titre d'exemple, un objet quittant le Nord-Est pour le Sud-ouest passe soit par le Nord ou par l'Est.

Implantation de la composante spatiale dans une bibliothèque open source de fouille de données

Dans les paragraphes précédents, nous avons décrit quelques mesures de similarité basées sur les relations spatiales entre entités géo-spatiales. Nous avons mis à contribution ces mesures pour enrichir une bibliothèque de fouille de données, KNIME, afin que celui-ci puisse supporter une fouille dynamique des

données géo-spatiales. KNIME est un logiciel de fouille, sous licence GPL, construit autour de l'API d'Eclipse. Il bénéficie par ailleurs d'une grande modularité et d'une interactivité grâce à ses multiples composants organisés sous forme de nœuds et peuvent être combinés à la manière d'un flow de données.

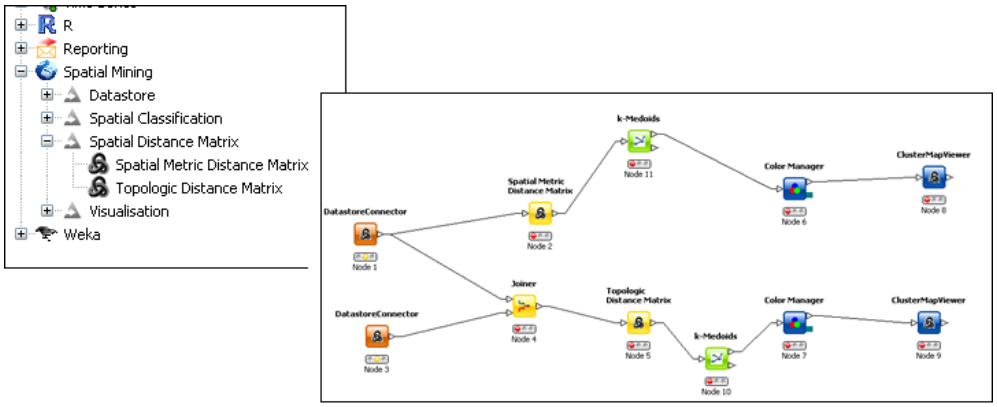


Figure B.4-4: Un plugin Géo-spatial pour KNIME

L'enrichissement de KNIME a consisté à l'implémentation d'un ensemble de nœuds assurant diverses fonctionnalités (voir Figure B.4-4).

On note : la lecture de base de données géo-spatiales, la construction d'une matrice de distance sur la base de mesures de similarité géo-spatiales, la visualisation cartographique du résultat de fouille de données.

Lecture de base de données géo-spatiale

Cette fonctionnalité permet la lecture des données stockées dans une base de données géo-spatiales. Elle constitue un préalable à toute opération de fouille de données. La lecture se limite pour le moment aux données contenues dans un système de gestion de base de données géo-spatiales, mais sera étendue pour prendre en compte la lecture de données provenant de fichiers. L'information de nature géographique de ces données est affichée sous la forme d'une chaine WKT (voir Figure B.4-5). Cette information est affichée dans une colonne de type géométrique – préalablement implémenté. La géométrie constitue de ce fait dans KNIME un type à part entière sur lequel, on peut – à l'instar des types de données classiques – effectuer diverses opérations.

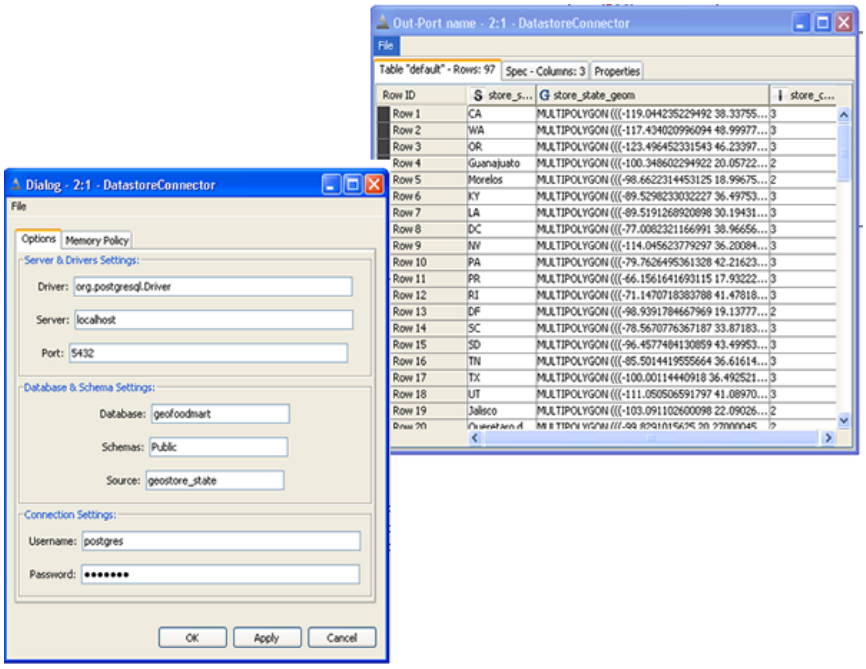


Figure B.4-5: Implémentation d'un type Géométrique dans KNIME

Matrice de distance basée sur des mesures de similarité géo-spatiale

Après la mise en œuvre de la fonctionnalité de lecture de données géo-spatiales dans KNIME, l'étape suivante à consister à l'implémentation des mesures de similarité géo-spatiales. Pour ce faire, nous avons développé deux(2) fonctionnalités de calcul de matrice de distance spatiale. La première de ces fonctionnalités (voir Figure B.4-7) utilise pour la construction de cette matrice, les mesures de similarité basées sur les relations métriques et de reconnaissance de formes. La deuxième (voir Figure B.4-6) quant à elle se fonde sur des mesures basées sur les relations topologiques.

Ces matrices de distances, en plus d'utiliser les mesures de similarité géo-spatiales, mettent à contribution les mesures de similarité « traditionnelles » de sorte à obtenir des matrices de distances agrégées ; i.e. tenant compte à la fois des composantes descriptives et géométriques de l'information géo-spatiale.

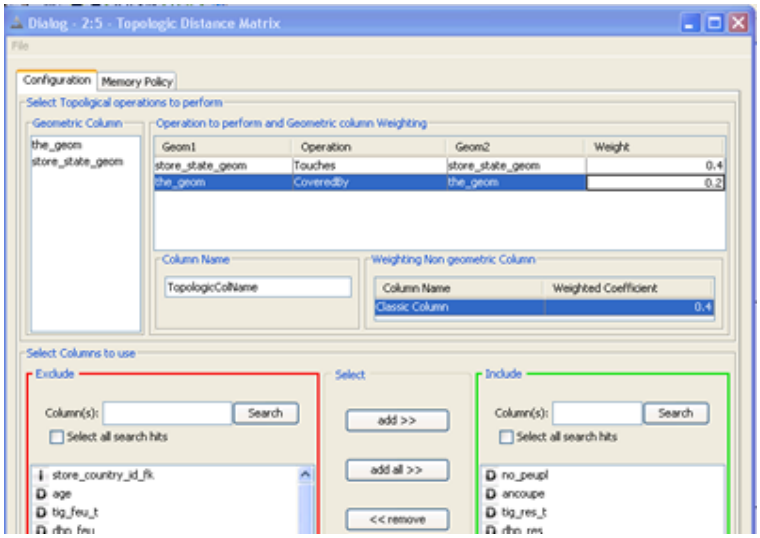
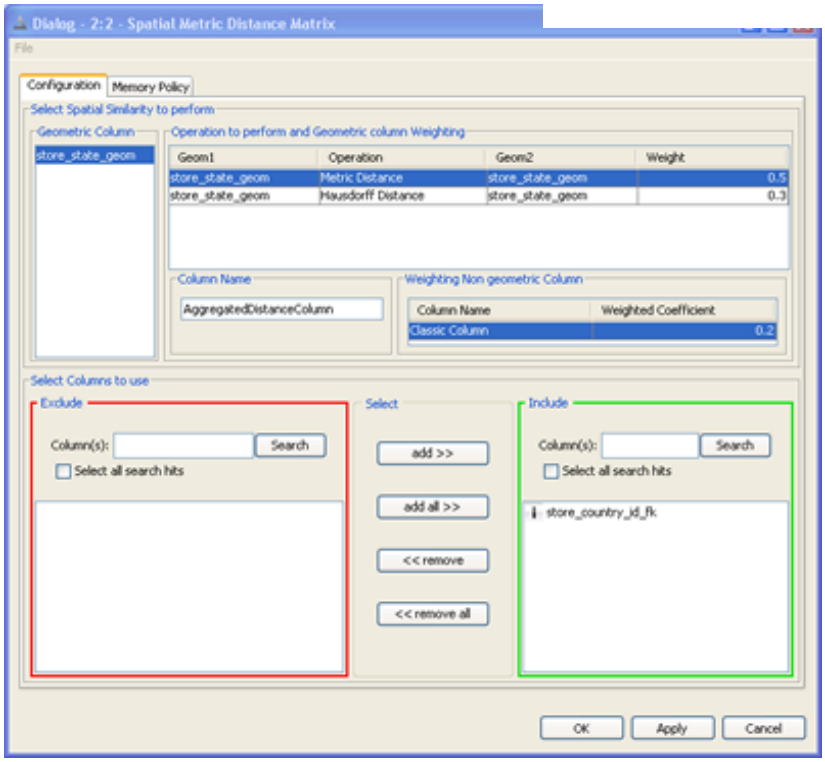


Figure B.4-6: Matrice de distance spatiale fondée sur des relations topologiques

En plus de la possibilité offerte pour l'agrégation, l'utilisateur a le choix d'attribuer un coefficient de pondération pour chaque opération.



Les différentes matrices ainsi obtenues peuvent servir à la réalisation de tâches de regroupement ou clustering. A titre d'illustration, nous avons mis à contribution ces matrices pour effectuer un clustering basé sur les K-Medoïds.

Visualisation

Figure B.4-7: Mesure de similarité basée sur les relations métriques et de reconnaissance de forme

cartographique des résultats de la fouille de données

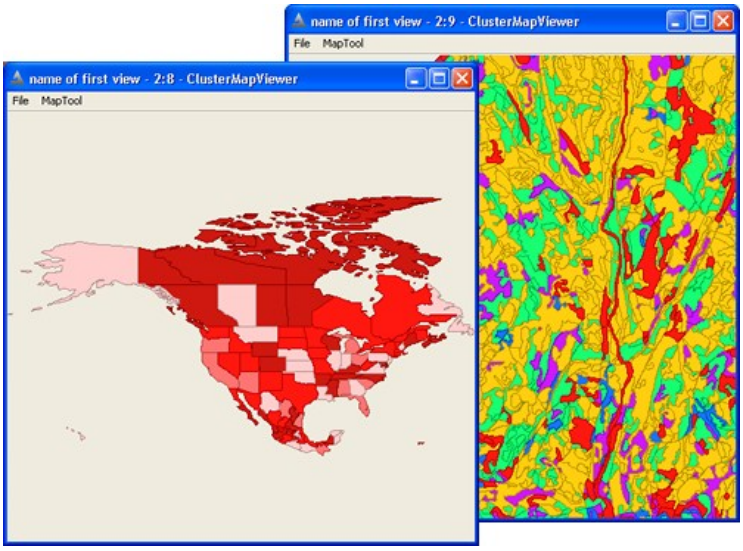


Figure B.4-8: Visualisation cartographique du résultat d'une fouille de données

De part la nature interactive de la fouille de données, la présentation ou visualisation des résultats tient une place importante. Fort de cela, nous avons implémenté une fonctionnalité de visualisation cartographique (voir Figure B.4-8).

Cet outil permettra à l'utilisateur d'apprécier de façon visuelle, le résultat de la fouille et s'il y'a lieu d'ajuster certains paramètres. Ce visualisateur supporte les fonctions classiques de zoom, déplacement, information lorsque l'on clique sur la carte.

Conclusion

La fouille de données spatiales, comparée à la fouille de données « traditionnelles », est un domaine relativement jeune qui est toujours à la recherche de ses repères. Des différentes approches de fouille qui existent, on note que celles-ci ont montré certaines limites. La nouvelle approche que nous avons proposée offre bien d'avantages en ce sens qu'elle tient compte de la spécificité des données géo-spatiales à savoir les différents types de relations que celles-ci peuvent entretenir. Aussi, cette approche met à contribution les outils existant de fouille de données en considérant dynamiquement et de façon transparente la composante spatiale de l'information. Une intégration réussie a été effectuée dans la bibliothèque open source KNIME. Cet outil supporte non seulement le type géo-spatial tout comme les types classiques de données ; mais également des fonctionnalités de clustering et de classification en tenant compte des mesures de similarité géo-spatiales.

Bien que l'intégration soit réussie, nombre de travaux reste à faire. Notamment l'implémentation de mesures de similarité fondées sur les relations directionnelles, la prise en compte des mesures de

similarité spatiales dans d'autres techniques de fouille de données comme les règles d'association, la classification bayésienne, les réseaux de neurones, etc. aussi, l'optimisation de ressource mémoire reste un défi à relever parce que le calcul impliquant les données géo-spatiales, à la différence des données classiques, coûte en temps. Avec l'intégration transparente de la composante géo-spatiale, des fonctionnalités gravitant autour de la tâche même de fouille de données sont à implémenter en l'occurrence celles consistant a l'analyse exploratoire des données.

Annexe E : Détails sur la topologie quantitative

E.1. La proximité ou closeness

E1.1 Distances entre limites

Il existe deux notions l' « outer-closeness » et l' « inner-closeness » permettant d'estimer la distance entre les limites d'une ligne et d'un polygone. Elles peuvent servir par exemple à détailler différentes configurations topologiques. En effet, l'outer-closeness peut servir de distance pour davantage quantifier les relations topologiques du type : intersection limite et intérieure ; et dans une certaine mesure l'adjacence. L'inner-closeness quant à elle peut détailler les relations topologiques suivantes : intersection intérieure et inclusion limite et intérieure.

Il peut arriver que ces deux(2) mesures de distance puissent chacune être utilisées pour décrire une configuration topologique donnée. Dans ce cas, il appartiendra à l'utilisateur effectuant la fouille de voir laquelle des distances convient la mieux. Au pire des cas, on pourra utiliser ces mesures de façon complémentaire. Bien entendu, cela n'est pas sans conséquence sur l'utilisation de ressources machines. En revanche, on obtient des résultats de fouille plus précis.

Outer-closeness

Cette notion traduit la distance séparant un point situé sur la limite extérieure d'une ligne et la limite d'un polygone dans la configuration représentée à la **Erreur ! Source du envoi introuvable.**. Si on note ∂L , la limite de la ligne et R^- l'extérieur du polygone, cette distance peut être évaluée si et seulement si $\partial L \cap R^- = \nexists \emptyset$

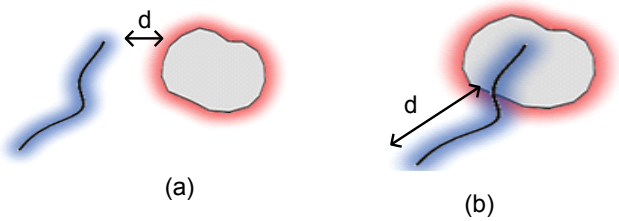


Figure E.1-1: configurations dans lesquelles on peut estimer l' «outer-closeness »

En clair et comme l'indique la figure ci-dessus, il faut qu'au moins une limite de la ligne soient situé à l'extérieur du polygone. La distance « d » illustrée sur les Figure E.1-1-a **Erreur ! Source du renvoi introuvable.**-b représente l' « outer –closeness » qui n'est utre que la plus courte distance séparant la limite extérieure de la ligne d'avec la frontière du polygone.

Inner-closeness

À la différence de l'outer-closeness, l'inner-closeness représente la distance séparant la limite intérieure d'une ligne d'avec la frontière d'un polygone (cf. **Erreur ! ource du renvoi introuvable.**). Si on note ∂L , la limite de la ligne et R^o l'intérieur du polygone, cette distance peut être évaluée si et seulement si $\partial L \cap R^o = \emptyset$. En claire il faut nécessairement que l'une des limites de la ligne soit à l'intérieur du polygone.

Figure E.1-2: configurations dans lesquelles on peut estimer l'inner-closeness

L'inner closeness représente la plus petite distance séparant la limite intérieure de la ligne d'avec la frontière du polygone (distance d sur la figure ci-dessus). Bien entendu, les deux limites de la ligne peuvent se retrouver à l'intérieur du polygone (Figure E.1-3-b), auquel cas, l'inner-closeness représente la plus petite des distances séparant les deux limites de la ligne d'avec la frontière du polygone.

E.1.2 Distance entre intérieur et limite

Les distances entre intérieur et limite respectivement de ligne et polygone sont capturées par « outer-nearness » et l' « inner-nearness » selon que l'intérieur de la ligne se situe à l'intérieur ou à l'extérieur du polygone.

Outer-nearness

L' « outer-nearness » représente la plus courte distance séparant un point situé à l'intérieur de la ligne et la limite du polygone lorsque la ligne (intérieur et limite) est à l'extérieur du polygone (cf. Figure E.1-4-a).

Si on note ∂L et L^o respectivement les limites et l'intérieur de la ligne et R^- l'intérieur du polygone, cette distance peut être évaluée si et seulement si $\partial L \cap R^- = \neg\emptyset$ et $L^o \cap R^- = \neg\emptyset$.

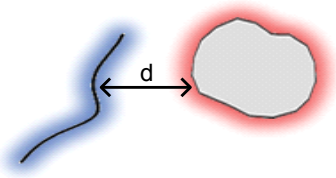


Figure E.1-4: Configuration dans laquelle on évalue un outer-nearness

Inner-nearness

L'inner-nearness décrit la distance séparant un point situé à l'intérieur de la ligne d'avec la limite du polygone. Contrairement à la distance précédente (outer-nearness), la ligne devrait être complètement située à l'intérieur du polygone (cf. Figure E.1-5et Figure E.1-6). Ainsi donc, cette distance ne peut être évaluée que si $\partial L \cap R^- = \emptyset$ et $L^o \cap R^- = \emptyset$.

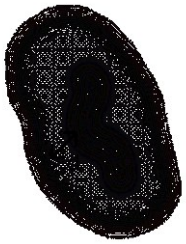


Figure E.1-6: Configuration dans laquelle on peut calculer un inner-nearness

E.2. Partitionnement ou splitting

E.2.1 Considération de la longueur de l'objet linéaire

On s'intéresse dans ce cas de figure à la longueur de ligne se trouvant à l'intérieur ou à l'extérieur de l'entité surfacique. La longueur obtenue est ensuite relativiser vis-à-vis de la longueur totale de l'entité linéaire. On note trois (3) ratios: *l'inner transversal splitting*, *l'outer transversal splitting* et la *line alongness*.

Inner et outer transversal splitting

Les configurations dans lesquelles sont évaluées l'inner et l'outer transversal splitting sont à quelques égards semblables aux conditions du inner et outer closeness. Il faut en effet, pour l'inner transversale splitting qu'il y'ait intersection entre les intérieurs des entités linéaires et surfaciques ($L^o \cap R^o = \neg\emptyset$) (voir Figure E.2-1-a). Réciproquement pour l'outer transversal splitting, il faut que l'intérieur de la ligne soit en contact avec l'extérieur du polygone ($L^o \cap R^- = \neg\emptyset$) (voir Figure E.2-2-b)

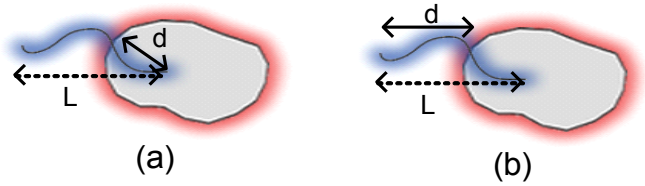


Figure E.2-3: configuration pour évaluer un inner (a) et un outer(b) transversal splitting

La longueur ***d*** mesurée, que ça soit pour l'inner ou l'outer transversal splitting est mise en rapport avec la longueur totale de l'objet linéaire. On obtient le ratio suivant :

$$r = \frac{d}{L}$$

Avec d étant la longueur de ligne à l'intérieur ou à l'extérieur du polygone et L, la longueur totale de la ligne.

On peut noter que ces deux mesures sont complémentaires. En effet, on peut passer aisément de l'un à l'autre. Si r_1 est la valeur de l'inner transversal splitting, on obtient la valeur de l'outer splitting, r_2 comme suit : $r_2 = 1 - r_1$.

L'utilisation de ces ratios est adaptée pour des configurations topologiques de type intersection intérieure et inclusion limite dans une certaine mesure à supposer que l'une des limites de la ligne touche la limite du polygone (voirFigure E.2-2).

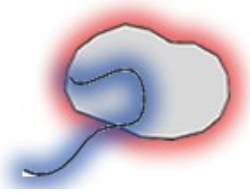


Figure E.2-4: Exemple de configuration d'intersection limite

Pour d'autres types de configuration, on peut bien évidemment utiliser ces ratios. On obtiendrait alors des résultats non nécessairement signifiants. Par exemple, on peut

s'amuser à utiliser inner transversal splitting dans un cas de disjonction. On obtiendrait alors une valeur nulle (0) ; étant donné qu'aucune partie de la ligne ne fait intersection avec l'intérieur du polygone. Également pour un cas d'inclusion totale, la valeur de ce ratio serait de 1.

Line alongness

Dans ce type de configuration, il est nécessaire que l'intérieur de l'objet linéaire soit en intersection avec les limites du polygone (cf. Figure E.2-3). Le ratio à ce niveau est obtenu est faisant le rapport entre la longueur de la ligne qui touche les limites de l'objet surfacique et la longueur totale de la ligne. Si on note di, les longueurs de la ligne en contact avec la limite du polygone, la formule de ce ratio est donnée par :

$$r = \frac{\sum di}{l}$$

Figure E.2-5 : configuration dans lesquelles on mesure une *line alongness*

On imagine ce type de ratio utile dans une configuration topologique de type adjacence afin de mesurer le degré d'adjacence entre deux entités géo-spatiales.