Détection et interprétation visuelle d'outliers dans les grands ensembles de données

Lydia BOUDJELOUD, François POULET

ESIEA – Pôle ECD
38, rue des docteurs Calmette et Guérin
Parc Universitaire de Laval-Changé, 53000 Laval
{boudjeloud | poulet}@esiea-ouest.fr

Résumé. Nous présentons un algorithme hybride de détection d'outliers (individus atypiques) dans de grands ensembles de données, utilisant un algorithme génétique pour la sélection des attributs et une approche basée sur la distance pour la détection de l'élément outlier (atypique) suivant ce sous-ensemble d'attributs. Une fois l'outlier trouvé, nous essayons de l'expliquer : est-ce une erreur, un bruit ou une valeur significativement différente des autres ? Pour ce faire, on utilise des méthodes visuelles telles que les coordonnées parallèles. Nous évaluons les performances de notre méthode sur différents ensembles de données de grandes dimensions et le comparons avec les algorithmes existants.
Mots-clefs. fouille de données, détection d'outliers, visualisation, algorithmes génétiques, coordonnées parallèles, grands ensembles de données.

1. Introduction

Le développement du réseau internet et la baisse des coûts du matériel informatique ont permis à de nombreux organismes de constituer de grandes masses de données trop volumineuses et complexes pour pouvoir être appréhendées par un utilisateur. L'Extraction de Connaissances à partir de Données (ECD) est née de ce besoin, on la définit comme étant l'extraction de nouvelles connaissances potentiellement utiles à partir de grandes quantités de données [Fayyad et al. 1996], le cœur du processus d'ECD est la fouille de données (Data Mining). Dans ce cadre précis (de fouille de données), nous nous intéressons à la recherche d'outliers (individus atypiques). La recherche d'outliers a de nombreuses applications telles que la détection de fraudes, la recherche pharmaceutique, les applications financières, le marketing, etc. Un outlier (individu atypique) est un petit ensemble de données, un point ou une observation qui est considérablement différent, divergent, dissemblable ou distinct du reste des données. Le problème est alors de définir cette dissimilitude entre objets, ce qui caractérise un outlier. Typiquement, celle-ci est estimée par une fonction calculant la distance entre objets, la tâche suivante consiste à déterminer les objets les plus éloignés de la masse. Certaines difficultés apparaissent lorsque l'on est face à des ensembles de données ayant un grand nombre de dimensions en terme d'attributs. En effet, dans les ensembles à grandes dimensions, les données sont rares et la notion de voisinage perd de son sens. La rareté dans les espaces de grandes dimensions implique que tout point est candidat pour être un bon outlier et donc la recherche d'outliers devient complexe et coûteuse en temps de calcul.

Nous allons donc essayer de détecter les outliers (individus atypiques) en n'utilisant qu'un sous-ensemble d'attributs. Ce type d'approche peut éliminer certains problèmes, on peut en citer au moins deux :

1) un outlier l'est rarement à cause de tout l'ensemble d'attributs.

2) certains attributs peuvent être discriminants pour un outlier donné et pas du tout pour un autre outlier.

En effet, il peut s'avérer que certains attributs composant les données ont plus ou moins d'importance dans la détection de certains outliers. La façon dont seront considérés ces attributs va donc déterminer la pertinence de leur prise en compte pour la détection d'outliers.

Nous proposons un algorithme hybride de détection d'outliers dans de grands ensembles de données utilisant un algorithme génétique pour la sélection des attributs et une approche basée sur la distance pour la détection de l'élément le plus éloigné de la masse suivant ce sous-ensemble d'attributs.

Nous présentons dans une première partie un tour d'horizon des méthodes de détection d'outliers en détaillant les avantages et les inconvénients de chacune. Nous expliquons le contexte de notre étude, les grands ensembles de données, nous détaillons notre algorithme puis nous commentons les résultats obtenus sur quelques ensembles de données, nous essayerons ensuite d'interpréter visuellement les résultats obtenus, une étape de post traitement des données, nous terminons enfin, par la conclusion et les travaux futurs.

2. Etat de l'art

2.1. La détection d'outliers

Ces dernières années de plus en plus de publications traitent de la détection d'outliers : [Barnett et Lewis 1994, Arning et al. 1996, Knorr et Ng 1998, Ramaswamy et al. 2000, Rocke et Woodruff 1999, Aggarwal et Yu 2001]. Les différentes méthodes existantes pour la détection d'outliers peuvent être classées selon les catégories suivantes :

2.1.1. Méthodes basées sur la densité

Dans ce type de méthodes, les clusters sont considérés comme des régions denses de l'espace des données séparées par des régions de faible densité (considérées souvent comme du bruit). Certaines de ces méthodes se basent sur un critère local appelé LOF : Local Outlier Factor, affecté à chaque élément de l'ensemble des données, il dépend de la densité locale de voisinage de ce dernier. Les éléments sont ensuite triés à partir de ce facteur, et les objets ayant un grand LOF sont considérés comme outliers, [Breunig et al. 2000]. [Papadimitriou et al., 2002] présentent LOCI, un algorithme qui pointe sur les éléments outliers de l'ensemble des données. Il calcule pour chaque élément la densité d'un voisinage (inférieur à un certain rayon). L'élément est un outlier, si cette densité tend vers 0.

2.1.2. Algorithmes de clustering (classification non supervisée)

Plusieurs algorithmes de clustering : DBSCAN [Ester et al. 1996], BIRCH [Zhang et al. 1996], GDBSCAN [Sander et al. 1998], CURE [Guha et al. 1998] et OPTICS [Ankerst et al.

1999] détectent les outliers. Sachant que leur objectif principal est de classifier les ensembles de données, ils ne sont pas optimisés pour la détection d'outliers. Dans la plupart des cas, la définition d'outlier ou le critère de détection sont implicites et ne peuvent pas être facilement référencés dans les procédures de clustering. Ces approches ne sont pas appropriées pour des ensembles de données de grandes dimensions [Jain et al. 1999].

2.1.3. Méthodes basées sur les distances

Dans cette catégorie de méthodes, un élément est un outlier si la distance entre cet élément et tous les éléments de l'ensemble des données est supérieure à un certain seuil. Ceci devient très coûteux en temps de calcul, si nous devons calculer la distance entre chaque élément de l'ensemble des données [Knorr et Ng 1998, Knorr et al. 2000].

2.1.4. Méthodes basées sur les distributions

Dans cette catégorie les méthodes proviennent généralement du domaine statistique, elles utilisent les modèles de distribution standards (Normal, Poisson, etc.). Ces modèles de distribution servent à ajuster les données en utilisant leurs caractéristiques statistiques. L'application de ces méthodes sur des ensembles de données de grande taille est très coûteuse, elles ne sont pas très performantes pour des ensembles de données de grandes dimensions [Barnett et Lewis 1994].

2.1.5. Approches basées sur la profondeur

A chaque individu est affecté un niveau de profondeur (la profondeur Pi d'un point P dans la dimension i, est le minimum entre le nombre de points à sa gauche et le nombre de points à sa droite dans la dimension i. La profondeur du point P dans l'espace en D dimensions est le minimum des Pi (i : 1 … D)). Les données sont donc représentées en niveaux dans l'espace des données, en respectant la valeur de profondeur de chaque point. Les éléments qui se trouvent dans le niveau extérieur sont considérés comme outliers. Les algorithmes de cette catégorie ne peuvent pas considérer un large D (>3) [Johnson et al. 1998].

2.1.6. Méthodes basées sur les réseaux de neurones

Ces méthodes utilisent un perceptron multicouches pour modéliser les données à partir d'un petit ensemble de données représentatif, un apprentissage est ensuite effectué sur le reste des données. Ce réseau de neurones tente de reproduire la forme d'entrée durant l'apprentissage, les outliers sont les moins reproduit par l'apprentissage à la sortie et donc ont une grande erreur de reconstruction, cette erreur est utilisée comme facteur de mesure pour l'élément outlier [Hawkins et al. 2002, Williams et al. 2002].

2.2. Problème des ensembles de données de grandes dimensions

Le problème de détection d'outliers dans des ensembles de données de grandes dimensions a été le sujet de plusieurs travaux [Hinneburg et al. 2000, Knorr et Ng 1998, Ramaswamy et al. 2000]. Les approches proposées par ces derniers (comme pour certains

autres algorithmes de fouille de données) considèrent des concepts de voisinage pour trouver des outliers en se basant sur leur relation avec le reste des données. Cependant, dans les ensembles à grandes dimensions, les données sont rares et les notions de distance ou de voisinage perdent leur sens. L'éparpillement, la rareté dans les espaces de grandes dimensions implique que tout point est candidat pour être un bon outlier selon les perspectives de voisinage et donc la recherche d'outliers devient complexe et très coûteuse en temps d'exécution. [Hinneburg, et al. 2000] introduisent le concept de recherche d'outliers dans un sous-ensemble d'attributs. En effet, certains attributs peuvent être discriminants pour un certain outlier alors que ces mêmes attributs peuvent s'avérer peu révélateurs pour un autre outlier. La façon dont seront considérés ces attributs va donc déterminer la pertinence de leur prise en compte pour la détection d'outliers. L'approche la plus intuitive est d'énumérer tous les sous-ensembles d'attributs possibles et de rechercher le sous-ensemble qui nous donne l'élément le plus atypique du reste des données. [Kohavi et John 1997] démontrent que cette recherche est exponentielle. Pour pallier ce problème, il arrive que nous ayons recours à de nouvelles techniques, généralement étudiées en recherche opérationnelle, des heuristiques ou méta-heuristiques telles que les algorithmes génétiques, la recherche tabou ou le recuit simulé. A ce sujet [Aggarwal, et Yu 2001] introduisent une de ces techniques pour la recherche d'outlier en s'axant surtout sur l'étude du comportement des projections en mesurant la densité de l'ensemble des données suivant ces projections.

L'utilisation des algorithmes génétiques pour la recherche de dimensions pertinentes semble naturelle, la principale raison est qu'on interagit directement avec les dimensions. Chaque individu de l'algorithme génétique doit représenter un sous-ensemble de dimensions, la qualité de chaque candidat est évaluée par la fonction d'évaluation (appelée aussi fitness), selon l'objectif que l'on veut atteindre (dans notre cas, la recherche d'outlier). La représentation la plus utilisée de l'individu est la représentation binaire de n bits [Michalewicz 1996, Goldberg 1989, Mitchell 1996], l'espace de recherche résultant correspond donc, à un espace booléen à n dimensions. Chaque bit peut avoir deux valeurs (0 ou 1) indiquant que l'attribut est présent (bit à 1) dans l'ensemble sélectionné ou pas (bit à 0). Cette représentation n'apporte aucune information sur la pertinence de l'attribut. Un attribut pertinent peut être changé lors d'une mutation par exemple. De plus, lorsque l'ensemble de données a un grand nombre de dimensions allant jusqu'à plusieurs milliers, cette représentation est inappropriée et l'exécution devient très coûteuse en temps de calcul. Une autre représentation a été proposée par [Cherkauer et Shavlik 1996], chaque gène d'un individu est représenté par deux valeurs : (0) pour l'absence d'attribut et (A_i) pour l'attribut lui-même et pour souligner la pertinence d'un attribut, la valeur (A_i) peut apparaître deux fois dans le gène.

Plusieurs comparaisons ont été faite [Kudo et Skalansky 2000, Yang et Honavar 1998], ces comparaisons indiquent que les algorithmes génétiques sont plus performants lorsque le nombre d'attributs est supérieur à 100 par rapport aux autres méthodes de sélection de dimensions. Ils ont aussi été utilisés dans plusieurs domaines de la fouille de données et l'extraction de connaissances à partir des données tels que la découverte des règles d'association, [Banzhaf et al 1998] en fait un tour d'horizon complet.

Nous nous intéressons aux grands ensembles de données, notre objectif est de retrouver des sous-ensembles de dimensions pertinentes en détection d'outlier. Le premier avantage de notre méthode c'est qu'elle nous fourni des sous-ensembles réduits et donc, des solutions plus facilement interprétables visuellement. Nous allons détailler dans ce qui suit, l'algorithme hybride pour la recherche d'outlier dans un sous-ensemble d'attributs.

3. Algorithme hybride pour la détection d'outlier

3.1. Algorithmes génétiques

Les algorithmes génétiques [Holland 1975, Goldberg 1989] sont des algorithmes de recherche probabilistes qui simulent l'évolution naturelle, ils sont basés sur les mécanismes de sélection naturelle et génétique de l'être vivant. Pour les AGs, l'espace de recherche du problème est représenté par une collection d'individus. Chaque individu est représenté par un tableau de caractères, chaque case est appelée chromosome. L'objectif est de trouver un individu avec la meilleure identité génétique de l'espace de recherche. La qualité de chaque individu est mesurée avec une fonction objectif. Une partie de l'espace de recherche sera examinée à chaque itération de l'algorithme, cette dernière est appelée population. L'algorithme génétique commence généralement par une population initiale choisie aléatoirement et la qualité de chaque individu est évaluée. A chaque itération on sélectionne deux parents, un croisement sera opéré sur les deux individus pour créer deux enfants, l'un d'eux va être remis dans la population. Pour quelques générations une mutation sur un chromosome est opérée sur un des enfants.

3.1.1. La population de départ

Les individus sont constitués à partir d'un tableau contenant toutes les dimensions (attributs) disponibles qui décrivent l'ensemble des données. A ce niveau là, la population de départ est prête, nous l'évaluons à l'aide d'une procédure de recherche de l'élément outlier basée sur les distances que nous ne détaillons pas ici. Cette procédure aura comme sortie l'élément le plus éloigné des autres en ne considérant que le sous-ensemble d'attributs. Une fois la population évaluée, triée selon la plus grande distance trouvée, nous opérons un croisement sur deux parents choisis aléatoirement. L'un des deux enfants est ensuite muté avec une probabilité de 1/10 et remis dans la population en remplaçant un individu dans la deuxième partie de la population, sous la médiane. Le choix de l'individu remplacé se fait aléatoirement. Cette technique a été initiée par Reeves [Reeves 1995].

3.1.2. Croisement

Nous avons opté pour deux types de croisements : un point de coupe déterminé aléatoirement et un point de coupe optimisé, on détermine dans ce cas le meilleur point avant d'opérer la coupe, ce qui implique une évaluation de l'individu issu de chaque coupe possible.

3.1.3. Mutation

Nous définissons une mutation comme étant l'inversion d'un chromosome dans un individu. Cela revient à modifier aléatoirement la valeur d'un des composants de l'individu. Dans notre algorithme, nous modifions un chromosome en le remplaçant par un attribut différent des autres chromosomes composant l'individu, car nous pourrions avoir un individu ayant deux attributs similaires, cas à éviter. La mutation permet d'assurer une recherche aussi

bien globale que locale, selon le nombre de gènes mutés. Les mutations garantissent mathématiquement que l'optimum global peut être atteint. De plus, une population trop petite peut s'homogénéiser à cause des erreurs stochastiques. Les gènes favorisés par le hasard peuvent se répandre au détriment des autres. La mutation permet de contrebalancer cet effet en introduisant constamment de nouveaux gènes dans la population. Traditionnellement la mutation est appliquée avec un faible taux aux enfants pour empêcher une convergence trop rapide de l'algorithme génétique, nous avons choisit d'opérer une mutation toutes les 10 générations.

3.1.4. Fonction d'évaluation

Pour notre algorithme génétique les individus sont des combinaisons d'attributs qui décrivent l'ensemble des données, nous évaluons la population à l'aide d'une procédure se basant sur les distances qui trouve l'élément le plus éloigné de l'ensemble de données pour cette combinaison d'attributs. L'évaluation basée sur les distances peut être faite de deux façons : pour un sous-ensemble d'attributs donné, on calcule la distance euclidienne entre chaque élément et tous les autres éléments de l'ensemble de données ou bien, on détermine le centre de gravité de l'ensemble des données, puis on calcule la distance euclidienne entre chaque élément et le centre de gravité. Nous recherchons dans les deux cas la plus grande distance générée par les différentes combinaisons de dimensions ainsi que l'élément qui en est la cause. Nous ne retenons que le premier élément trouvé, il sera considéré comme l'individu le plus atypique (outlier) par notre approche. Nous remarquerons, par la suite, que ce même élément est détecté par un algorithme récent de détection d'outliers comme étant outlier sur tout l'ensemble de données.

3.1.5. Réglage des paramètres

L'un des avantages des algorithmes génétiques est de varier les solutions même si l'algorithme commence par une population de petite taille, ceci grâce aux différents opérateurs génétiques utilisés. Pour le problème qui nous intéresse, nous avons fixé la taille de la population à 60 individus. Nous pouvons ainsi calculer de 10000 à 100000 générations en quelques minutes sur un Pentium IV à 1,7 GHz sous Linux. Pour un nombre de générations fixe, avec une population de 60 individus, nous retrouvons toujours la même solution (vérifié sur plusieurs ensembles de données).

Le choix du nombre de générations peut influer sur l'optimalité de la solution car il faut laisser l'algorithme converger tout en ayant un bon compromis entre le temps de calcul et la qualité du résultat. Un nombre de générations trop petit fera probablement évoluer l'algorithme vers un optimum local peu intéressant. Un nombre de générations trop grand sera inutile car le temps de convergence sera excessif. L'algorithme s'arrête après un certain nombre de mutations, de croisements et/ou de générations sans amélioration de la solution. Nous l'avons fixé à 1% du nombre maximal de générations. Pour le problème qui nous intéresse et avec les moyens de calcul dont nous disposons, nous avons constaté qu'un nombre de 100 000 générations constituait un bon compromis pour confirmer l'optimalité de la solution trouvée. Il arrive qu'à cause de problèmes de mémoire, certains ensembles de données (Segmentation et Ovarian) n'arrivent pas à atteindre les 100 000 générations et s'arrêtent autour de 50 000 générations, le résultat est donc affiché tel quel.

3.2. Complexité de l'algorithme

La complexité de l'algorithme dépend du nombre de fois où l'on fait appel aux deux procédures de calcul des distances : en calculant la distance entre tous les éléments (procédure P1) ou bien en déterminant en premier le centre de gravité (procédure P2). A chaque itération de l'algorithme génétique, pour chaque individu de la population (qui représente dans notre cas une combinaison d'attributs), nous déterminons l'élément le plus éloigné correspondant à cette combinaison en utilisant les deux procédures basées sur la distance. Ceci explique le temps d'exécution de l'algorithme qui peut être long face aux ensembles de données ayant un grand nombre d'éléments. Nous avons les variantes suivantes pour notre algorithme :
- AGP1 : l'algorithme génétique utilisant la procédure 1 comme fonction d'évaluation dont la complexité est de l'ordre de O (D * Nb_ind^2),
- AGP2 : l'algorithme génétique utilisant la procédure 2 comme fonction d'évaluation dont la complexité est de l'ordre de O (Nb_ind*D),
avec : Nb_ind = nombre d'individus et D = nombre d'attributs du sous-ensemble.

Pour chaque exécution, nous avons fixé la taille de la population de notre algorithme génétique à 60 individus que nous déroulons pour un certain nombre d'itérations fixé à 100 000 itérations. Les résultats de notre algorithme restent identiques lorsque l'on fait varier ces paramètres, la taille de la population n'influe pas sur le résultat car justement l'avantage des algorithmes génétiques est de varier les solutions même si on démarre d'une population initiale de petite taille.

Il peut arriver que dans certains cas (selon le type des données) la détection d'outliers basée sur les distances ne soit pas appropriée, rappelons que notre objectif est de retrouver des sous-ensembles (combinaisons) d'attributs pertinents et donc le moyen utilisé pour la détection dépend plus de la définition d'un outlier et du type des données ainsi que de l'algorithme de détection d'outliers.

Ens. de données	Attributs	Éléments
Segmentation	19	2310
Lung	12533	32
Breast	24481	78
Colon	2000	62
Ovarian	15154	253
Prostate	12600	102
AMLL-ALL	7129	38
MLL	12582	57
DLBLC	4026	47
CNS	7129	62

TAB 1 – *Description des ensembles de données*

4. Tests et résultats

4.1. Résultats numériques

Afin de tester les algorithmes et de s'assurer de la pertinence du modèle proposé, nous allons procéder à l'évaluation des performances de notre algorithme génétique [Boudjeloud et Poulet 2004] en comparant les différentes variantes avec les résultats obtenus par l'algorithme LOCI [Papadimitriou et al., 2003] un algorithme qui détecte les éléments outliers de l'ensemble des données. Ces expérimentations nous permettent d'analyser le comportement des algorithmes par rapport à la durée d'exécution et aux critères des différents ensembles de données. Les variantes de l'algorithme ainsi que l'algorithme LOCI ont été programmées en langage C/C++ et implémentées sur un PC pentium IV à 1,7 GHz sous Linux. Les différents algorithmes ont été testés avec les ensembles de données préalablement centrés réduits (tab. 1) : tous les ensembles de données de type numérique sont issus du Kent Ridge Biomedical Data Set Repository [Jinyan et Huiqing, 2002], sauf Image Segmentation, qui provient de l'UCI Machine Learning Repository [Blake et Merz, 1998].

	AGP1			**AGP2**		
	Outlier	**Ens att**	**Temps**	**Outlier**	**Ens att**	**Temps**
Segment. (19*2310)	**1683**	0-**6**-**8**-16	10h	**1683**	5-**6**-7-**8**	30min 50sec
Lung (12533*32)	**10**	**12307-5936-1430**-7466	6min 22sec	**10**	**12307-5936-1430**-394	19sec
Breast (24481*78)	**9**	10069-23383-16859-16082	14min 26sec	**9**	7042-6019-18456-22913	46sec
Colon (2000*62)	**56**	**118-305-877-**806	9min 54sec	**56**	**118-305-877-**267	32sec
Ovarian (15154*253)	160	5649-13847-9050-5645	14min 52sec	78	10975-7777-14536-10980	130sec
Prostate (12600*102)	**80**	**11692**-22-12266-10448	26min 4sec	**80**	11742-1148-**11692**-10836	54sec
AMLL-ALL (7129*38)	**28**	5198-**5228-5710-5709**	8min 47sec	**28**	18-**5228-5710-5709**	22sec
MLL (12582*57)	**56**	6732-**8408**-1041-7060	19min 40sec	**56**	1888-6375-2773-**8408**	32sec
DLBLC (4026*47)	**0**	**1205-1269-1366-1367**	12min 20sec	**0**	**1205-1269-1366-1367**	25sec
CNS (7129*60)	**36**	**18-6395-5198**-5613	21min 24sec	**36**	**18-6395-5198**-1069	32sec

TAB 2 – *Résultats obtenus pour un sous-ensemble d'attributs de taille 4*

Nous évaluons ces algorithmes pour D=4 et D=9 (taille des sous-ensembles d'attributs). Nous avons choisit de travailler sur des sous-ensembles réduits pour avoir des solutions plus facilement interprétables visuellement. En effet, la limite des méthodes visuelles existantes est de l'ordre de quelque dizaine d'attributs, la visualisation et l'interprétation des résultats deviennent impossible lorsque l'on a plus de quelques dizaines d'attributs. Les résultats de nos tests (pour D=4 et D=9) sont décrits dans les tableaux 2 et 3 respectivement. Cependant, notre méthode ne se limite pas à quelques attributs. Dans un souci d'interprétation visuelle,

nous fixons D à 4 attributs. Les tableaux 4 et 5 décrivent les résultats obtenus par notre méthode sur deux ensembles de données (Colon et Lung cancer) en faisant varier la taille du sous-ensemble d'attributs.

	AGP1			AGP2		
	Out.	**Ens att**	**Temps**	**Out.**	**Ens att**	**Temps**
Segment. (19*2310)	**1683**	0-1-**6-8**-9-10-11-13-16	10h	**1683**	2-3-4-5-**6**-7-**8**-17-18	35min 45sec
Lung (12533*32)	**10**	3507-**12307**-12256-**1430-5936-394**-5593-3287-1544	1min 58sec	**10**	**1430-5936-394**-**12307**-2330-6529-3370-5115-1950	1min 25sec
Breast (24481*78)	**9**	22630-10007-20884-19433-6697-**22913**-19749-16356-16137	21min 8sec	**9**	11271-2218-17133-7401-1505-**22913**-1875-8857-10847	3min 30sec
Colon (2000*62)	**56**	**877**-22-316-14-15-806-**118**-21-**305**	7min 7sec	**56**	1101-1954-249-1456-267-**118**-342-**877-305**	2min 42sec
Ovarian (15154*253)	**160**	13274-6537-7916-13342-**8285**-7086-8457-8358-9050	30min 21sec	**160**	8257-**8285**-8274-8969-9035-8956-9038-9034-8278	10min 59sec
Prostate (12600*102)	**80**	6907-4831-603-7303-**11692**-10755-10921-7066-10836	19min 21sec	**80**	9951-**11692**-2113-8990-11742-8496-648-10836-11578	4min 34sec
AMLL-ALL (7129*38)	**28**	**18**-531-1778-**5710-5228**-1764-**5709**-5198-1838	2min 43sec	**28**	**18**-5057-**5228**-1221-**5709-5710**-265-5189-2682	1min 34sec
MLL (12582*57)	**44**	11525-4-10733-7414-**4926-12153**-1041-7354-4214	9min 36sec	**44**	**4926-12153**-11629-9098-1513-2517-1270-7354-8612	66sec
DLBLC (4026*47)	**0**	**1205-2497-1269-1367-1366-1171**-2599-1341-**1170**	4min	**0**	**1171-1367-1366-1205-1170**-754-39-**1269-2497**	1min 51sec
CNS (7129*60)	**36**	**5198-6775-18-6395**-66-811-6753-6511-6878	6min 39sec	**36**	5933-5268-**6775-5198**-2387-**18**-5911-6753-**6395**	2min 34sec

TAB 3 – *Résultats obtenus pour un sous-ensemble d'attributs de taille 9*

Les éléments et les attributs en gras indiquent que les mêmes éléments outliers, ainsi que les attributs sont retrouvés dans le même tableau pour les mêmes paramètres de l'algorithme génétique. Les éléments et les attributs soulignés indiquent que ces attributs ainsi que les éléments outliers sont retrouvés selon les différents paramètres d'exécution des algorithmes (les deux tableaux 2 et 3).

Dans les tableaux 4 et 5, nous avons pris deux ensembles de données (Colon et Lung cancer) sur lesquels nous avons fait varier la taille du sous-ensemble d'attributs. Nous retrouvons toujours un sous-ensemble d'attributs commun considéré comme pertinent par notre méthode.

P2 Lung	D=1	D=3	D=4	D=6	D=6000	D=12533
Sous-ens. d'attributs	**5936**	**5936-394-1430**	**5936-394-1430-12307**	**5936-394-1430-12307**-2330-525	**5936**-….-**1430**	0-…-12532
Outlier	10	10	10	10	10	10

TAB 4– *Résultats obtenus pour Lung Cancer en variant D*

P2 Colon	D=3	D=4	D=6	D=60	D=2000
Sous-ens. d'attributs	**305-877-118**	**118-305-877-267**	**305-267**-1220-**877**-342-**118**	**877**-….-**118**	0-…-2000
Outlier	56	56	56	56	56

TAB 5– *Résultats obtenus pour Colon Cancer en variant D*

Selon les différents tests, nous retrouvons le même outlier pour tous les ensembles de données, quelque soit la taille du sous-ensemble d'attribut ou la variante de l'algorithme génétique, sauf pour l'ensemble de données MLL où nous trouvons deux outliers différents pour deux tailles différentes du sous-ensemble d'attributs (D=4 ou D=9) quelque soit la variante de l'AG et l'ensemble de données Ovarian où l'élément 78 est détecté seulement avec AGP2 (4 attributs). Ces résultats peuvent confirmer que certains attributs peuvent être discriminants pour un certain outlier alors que ces mêmes attributs peuvent s'avérer peu révélateurs pour un autre outlier, sachant que l'algorithme génétique n'a pas atteint les 100 000 générations pour les variantes P1 (4 et 9 attributs) et P2 (9 attributs). Cependant, on les retrouve comme outliers détectés par LOCI.

Pour confirmer nos résultats, nous avons voulu comparer les résultats obtenus par l'algorithme génétique avec ceux de l'algorithme LOCI (*Local Correlation Integral*), [Papadimitriou et al., 2003] un algorithme récent, qui détecte les éléments outliers de l'ensemble des données. Nous avons programmé l'algorithme en langage C/C++ et implémenté sur un PC pentium IV à 1,7 GHz sous Linux. Les outliers détectés par LOCI sont indiqués selon un ordre croissant.

LOCI	Outlier	Temps
Segment. CR	781, 1230, 1626, 1717	~2h
Segment.	936, 1796, 501, 10170	~2h
Lung	5, 10, 12, 13, 15	1min 30sec
Breast	9, 19, 27, 49	1min 45sec
Colon	5, 10, 23, 43, 56	2min 30sec
Prostate	13, 67, 69, 80, 94	3min 49sec
Ovarian	37, 52, 60, 78, 149, 160, 205	44min 49sec
AMLL-ALL	10, 16, 19, 28, 33	1min 10sec
MLL	10, 31, 44, 52, 56	65sec
DLBLC	0, 7, 21, 35, 40	2min
CNS	18, 25, 36, 42, 50	40sec

TAB 6– *Résultats obtenus par LOCI*

LOCI repère les éléments outliers en affectant à chaque individu un facteur MDEF (*Multi-granularity deviation factor*), ce facteur décrit la déviation relative de la densité locale du voisinage par rapport à la moyenne des densités du voisinage de ce point. Pour exécuter cet algorithme il faut fixer certains paramètres [Papadimitriou et al., 2003] dont par exemple, le rayon maximal R_{max} et un seuil de déviation k. Les résultats peuvent être différents selon que les données sont centrées réduites (CR) ou pas. Nous avons fait plusieurs tests avant d'avoir les résultats décrits dans le tableau 6, nous avons pu obtenir des résultats concernant Lung Cancer centré réduit uniquement pour un R_{max}= 1000 et k = 1.5 et dans tous les autres cas pas d'outliers (ensemble vide). Concernant Segmentation après plusieurs tests nous avons pu avoir des résultats pour k = 3 et un rayon R_{max} = 60 pour l'ensemble non centré réduit et un rayon R_{max} = 1 pour l'ensemble centré réduit. Idem pour les autres ensembles (centrés réduits) où nous avons fait plusieurs tests avec différents paramètres sur tous les ensembles de données avant d'obtenir les résultats présentés dans le tableau 6. En étudiant plus précisément les résultats obtenus sur Lung par exemple, nous remarquons que nous avons obtenu pour un sous-ensemble d'attributs le même outlier (élément 10) que LOCI. Cependant, notre méthode permet de retrouver les attributs les plus pertinents pour les outliers détectés, tel que décrit dans les tableaux 4 et 5 où l'attribut 5936 est pertinent pour l'élément 10 (outlier) de l'ensemble de données Lung et la combinaison d'attributs (305-118-877) l'est pour l'élément 65 (outlier) de l'ensemble de données Colon.

4.2. Interprétation visuelle des résultats

Nous allons maintenant essayer d'expliquer les résultats de l'algorithme de détection d'outlier en utilisant des méthodes visuelles : les coordonnées parallèles [Inselberg 1985, Inselberg et Dimsdale 1990] et les matrices de scatter-plot [Becker et al., 1987, Carr et al., 1987]. Nous pouvons utiliser d'autres méthodes de visualisation, cependant les coordonnées parallèles ainsi que les matrices scatter-plot offrent une interactivité avec les attributs que les autres méthodes n'ont pas.

En effet, les coordonnées parallèles sont considérées comme l'un des outils les plus efficaces pour la visualisation de données multidimensionnelles. Nous utilisons les coordonnées parallèles pour montrer où se trouve l'outlier, sur quelles dimensions (attributs) ses valeurs sont significativement différentes.

Il ne faut pas oublier que l'on travaille sur des fichiers de grandes tailles. L'interprétation de résultats numériques n'est pas simple, d'où l'intérêt de faire coopérer les méthodes automatiques et la visualisation pour faciliter l'interprétation des résultats numérique [Poulet 2002]. Les figures 1A et 1B selon les différentes procédures montrent que l'élément 10 de l'ensemble de données Lung a bien des valeurs extrêmes sur les combinaisons de dimensions trouvées par l'algorithme génétique, ce qui le différencie des autres éléments.

La visualisation des résultats obtenus sur l'ensemble de données Colon tumor (figure 2) sur 4 dimensions montre bien que l'élément détecté par l'algorithme génétique est clairement séparé et éloigné de la masse et qu'il prend aussi des valeurs extrêmes sur toutes les dimensions. On les retrouve bien parmi les éléments outliers détectés par LOCI. Concernant l'ensemble de données Breast cancer, l'algorithme génétique a pu sélectionner des sous ensembles d'attributs de taille 4 ou 9 pertinents pour l'élément outlier 9, qui est aussi élément outlier de tout l'ensemble de données (détecté par LOCI). Les visualisations des figures 3A et 3B montrent effectivement le comportement différent de l'élément détecté comparé au reste des données.

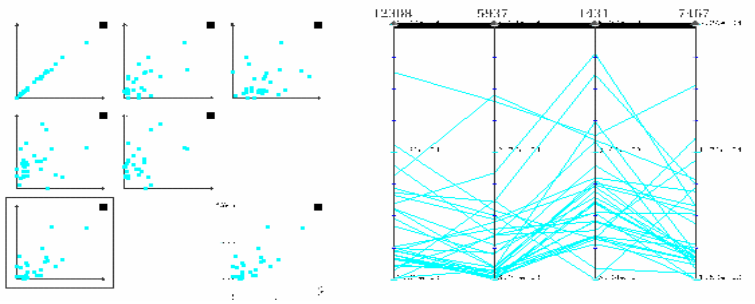

FIG. 1A - *Visualisation des résultats de AG_P1 sur Lung Cancer*

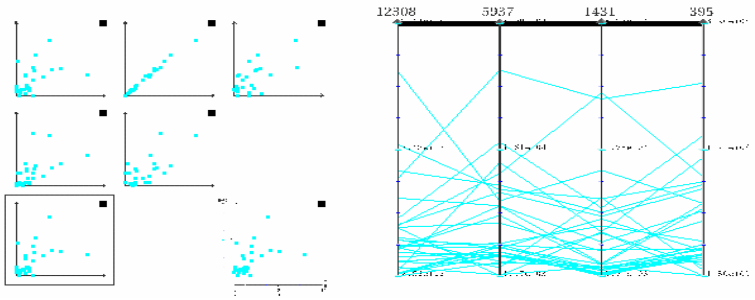

FIG.1B - *Visualisation des résultats de AG_P2 sur Lung Cancer*

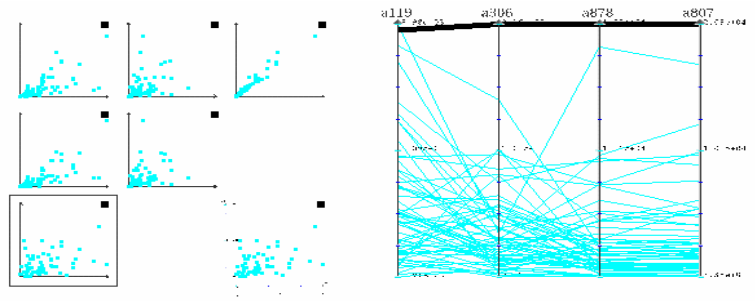

FIG.2A - *Visualisation des résultats de AG_P1 sur Colon Cancer*

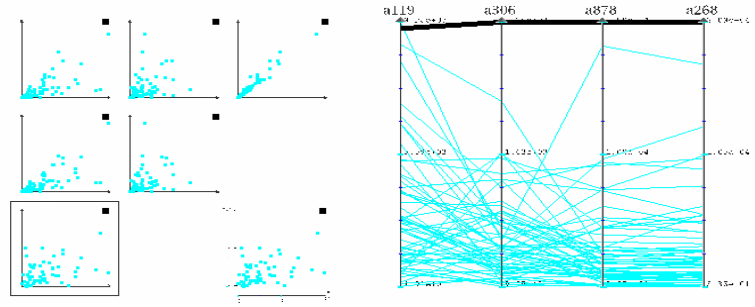

FIG.2B - *Visualisation des résultats de AG_P2 sur Colon Cancer*

Cependant, aucun attribut n'est commun aux différentes sélections, ceci peut être expliqué par le fait que Breast cancer est l'un des ensembles de données ayant le plus grand nombre d'attributs, ce qui implique un très grand nombre de combinaisons possibles sans oublier que certains attributs peuvent être redondants ou bruités. Il en est de même pour les ensembles de données Ovarian et MLL.

Le fait de calculer les distances de tout l'ensemble des données à chaque itération de l'algorithme génétique explique le temps d'exécution. Plus l'ensemble de données est important en taille (en nombre d'individus) plus l'exécution est longue. Il faut en effet calculer et stocker les distances entre un individu et chacun des autres (pour P1), pour tous les individus alors que dans le cas de la procédure P2, ce calcul est effectué une seule fois pour chaque individu par rapport au centre de gravité.

La diversité des résultats de LOCI sur Segmentation (que l'ensemble de données soit centré réduit ou pas), ne nous donne pas trop d'éléments de comparaison avec notre algorithme génétique. Il peut arriver que dans certains cas, selon le type de données (figure 5), l'élément peut être outlier sans avoir de valeurs extrêmes pour ses attributs, cependant, grâce à l'outil de visualisation utilisé, on voit que les valeurs de cet élément sur les attributs se retrouvent dans la masse, mais qu'il a un comportement atypique par rapport aux autres éléments de l'ensemble (comme par exemple : la droite horizontale sur la gauche de la figure 5). La visualisation permet de voir efficacement pourquoi l'élément est outlier, alors que des colonnes de chiffres ne permettraient pas facilement une telle interprétation (figure 4).

Le comportement de l'élément indiqué en gras sur les figures 4A ou 4B, a plus tendance à avoir une valeur élevée sur l'attribut a9 tandis que l'ensemble se concentre plus vers les valeurs faibles et pour certains attributs, l'élément a des valeurs complètement dans la masse (on le voit aussi dans les matrices de scatter-plot).

Concernant les visualisations de l'ensemble de données Ovarian (figures 6A, 6B), on remarque bien que les éléments détectés (en gras) ont des valeurs extrêmes (comportement atypique, outliers), même si l'élément détecté par P2 semble plus outlier que celui détecté par P1, ceci rejoint les remarques faites sur le nombre de générations non atteint par la variante P1 sur l'ensemble de données Ovarian.

L'ensemble des visualisations suivantes (figures 7 : A…E) décrit les résultats obtenus par l'algorithme génétique (P2) sur 4 dimensions, testé sur les autres ensembles de données. La visualisation des résultats obtenus montre bien que l'élément est différent de la masse ou bien présente un comportement différent (inverse) par rapport au reste des données (exemple : MLL, DLBLC et Prostate Figures 7A, 7C et 7D).

4.3. Modélisation de l'expertise

La visualisation des résultats obtenus montre bien que les points détectés sont éloignés et présentent un comportement atypique par rapport au reste des données, néanmoins nous ne pouvons fournir plus d'explication sur le type (erreur ou « outlier réel ») des points détectés par notre algorithme.

En effet, dans le cas de valeurs extrêmes on ne sait pas dire si cette valeur est une valeur possible ou non. Seul l'expert des données peut répondre à cette question. Dans le cas où le point détecté est une erreur on l'élimine de l'ensemble des données et dans le cas contraire on le garde dans les données car il peut représenter à lui seul des informations importantes.

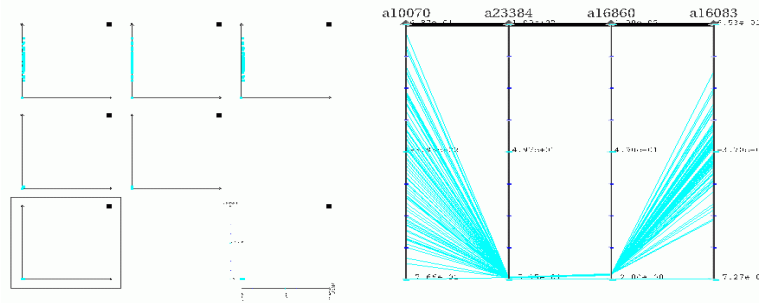

FIG.3A - *Visualisation des résultats de AG_P1 sur Breast Cancer*

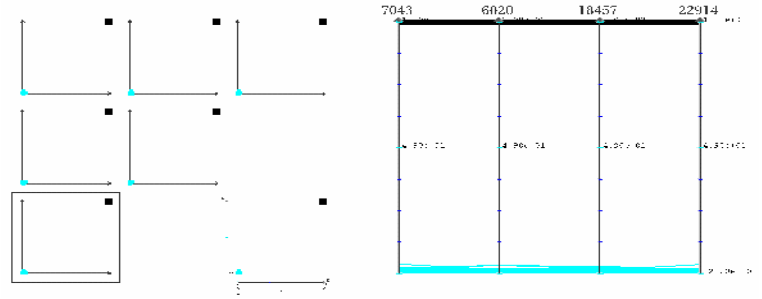

FIG.3B - *Visualisation des résultats de AG_P2 sur Breast Cancer*

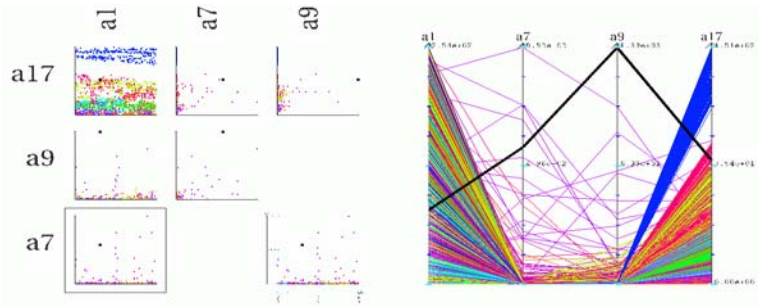

FIG. 4A - *Visualisation des résultats de AG_P1 sur Segmentation*

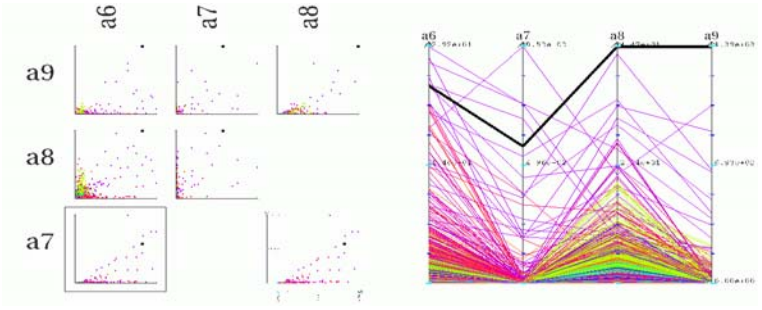

FIG. 4B - *Visualisation des résultats de AG_P2 sur Segmentation*

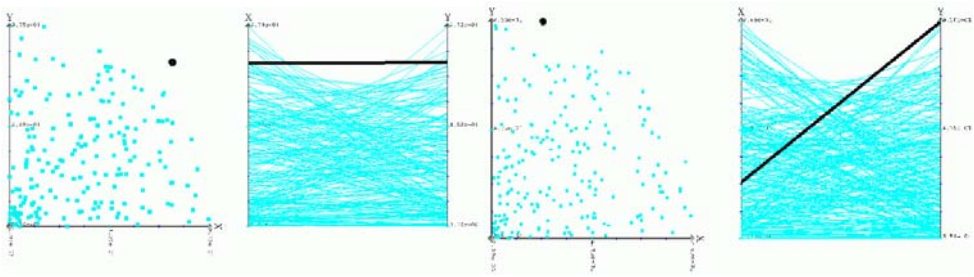

FIG. 5 - *Exemples de cas particuliers de visualisation*

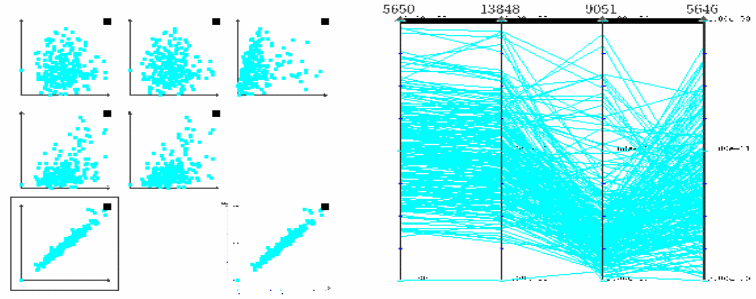

FIG. 6A - *Visualisation des résultats de AG_P1 sur Ovarian*

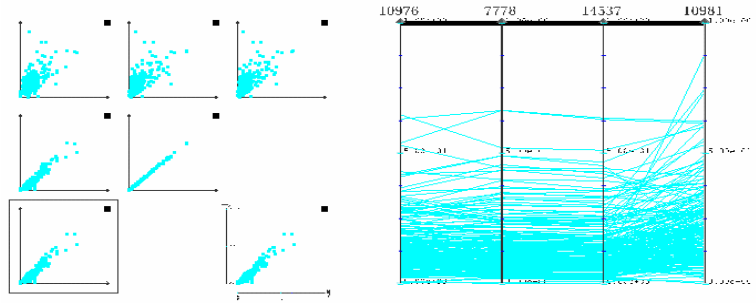

FIG. 6B - *Visualisation des résultats de AG_P2 sur Ovarian*

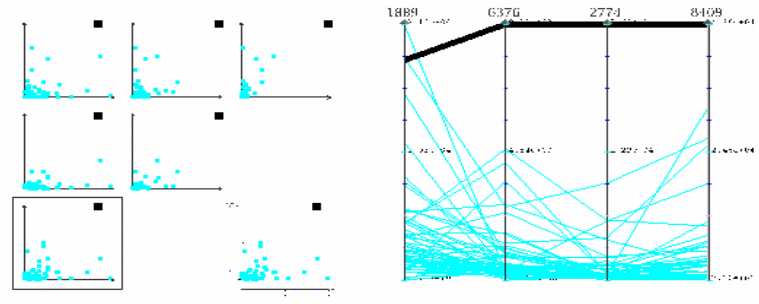

FIG. 7A - *Visualisation des résultats de AG_P2 sur MLL*

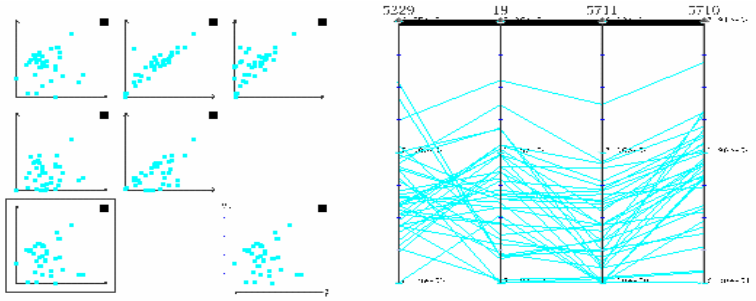

FIG. 7B - *Visualisation des résultats de AG_P2 sur AMLL-ALL*

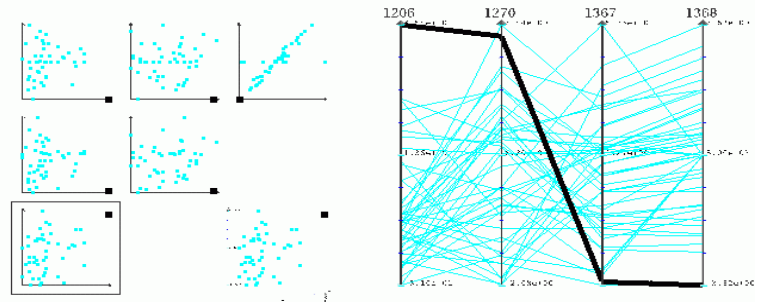

FIG. 7C - *Visualisation des résultats de AG_P2 sur DLBLC*

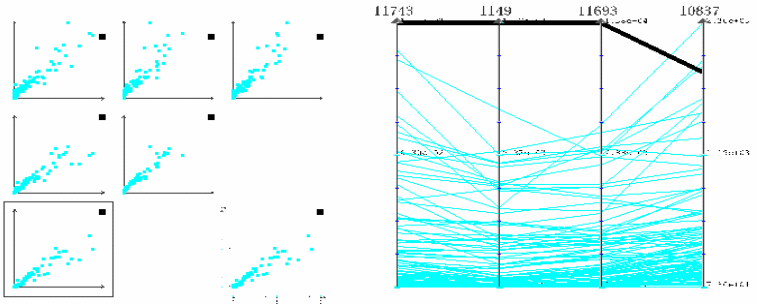

FIG. 7D - *Visualisation des résultats de AG_P2 sur Prostate*

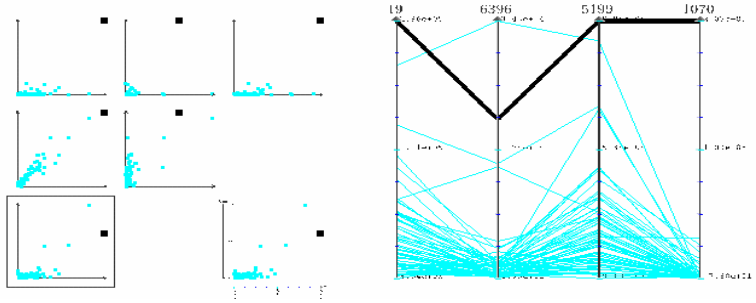

FIG. 7E - *Visualisation des résultats de AG_P2 sur CNS*

Un des moyens de combler cette lacune est de proposer un modèle des données permettant de qualifier au mieux les éléments détectés d'outliers ou d'erreurs. Ainsi, étant donné un nouvel élément introduit dans l'ensemble des données, nous pourrons utiliser le modèle pour prédire son état : outlier, erreur ou donnée normale.

Nous proposons donc de construire un modèle de l'expertise de l'expert. Celui-ci doit tout d'abord étiqueter les éléments qui ont été détectés comme étant outliers (on peut supposer qu'il n'y a que 2 types d'éléments : les erreurs et les « vrais outliers »).

A partir de cet ensemble de données étiquetées, on utilise un algorithme de classification supervisée (par exemple un algorithme d'induction d'arbre de décision) pour construire un modèle de l'expertise du spécialiste des données. Les nouveaux éléments outliers seront alors analysés avec le modèle construit et la présence de l'expert n'est plus indispensable pour qualifier les outliers.

5. Conclusion et perspectives

Nous avons présenté un algorithme permettant la détection d'outliers dans des ensembles de données ayant un grand nombre de dimensions en n'utilisant qu'un sous-ensemble de dimensions de l'ensemble initial. Notre but n'étant pas de créer un nouvel algorithme de détection d'outliers, nous avons utilisé une procédure basée sur la distance pour cette détection (n'importe quel autre algorithme de détection d'outliers peut le remplacer).

Les éléments détectés comme outliers dans le sous-ensemble de dimensions sont bien les mêmes que ceux que l'on trouve avec toutes les dimensions. Puisque le nombre de dimensions utilisé est faible, on peut ensuite visualiser ces éléments (à l'aide de matrices de scatter-plot ou de coordonnées parallèles) pour permettre à l'expert des données de qualifier ces outliers (par exemple en deux classes : erreur ou élément significativement différent de la masse).

Il ne faut pas oublier que l'on travaille sur des fichiers de grandes tailles. Cette étape n'est possible que parce que nous n'utilisons qu'un sous ensemble restreint de dimensions de l'ensemble de données initial. Cette qualification des outliers serait absolument impossible en considérant l'ensemble de dimensions comme l'illustre très bien l'exemple de la figure 8.

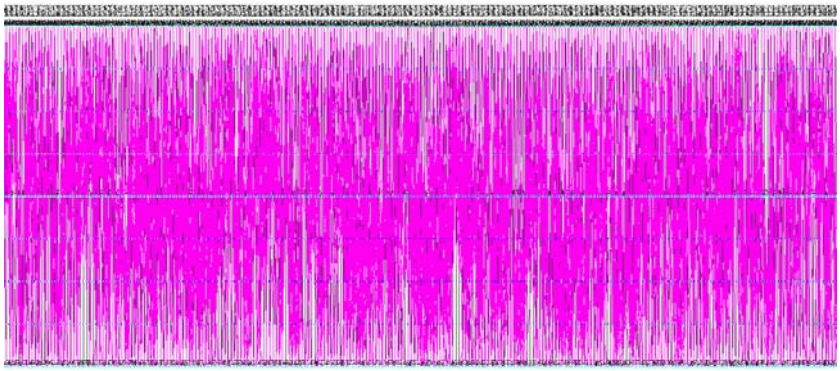

FIG. 8 – *Visualisation de Lung Cancer sur quelques centaines de dimensions*

Enfin une fois cette qualification effectuée, nous utilisons un algorithme de classification supervisée pour créer un modèle de l'expertise du spécialiste des données. Les nouveaux outliers peuvent alors être qualifiés par le modèle construit sans la présence de l'expert des données.

Nous pensons par la suite poursuivre notre objectif qui est de qualifier au mieux une combinaison d'attributs, retrouver la meilleure combinaison d'attributs pour l'élément outlier, dans le but de réduire l'espace de recherche sans perte de qualité des résultats. Trouver dans ce sens un facteur ou une fonction objectif pour l'algorithme génétique qualifiant une combinaison d'attributs nous permettra d'optimiser au mieux l'algorithme et d'améliorer les temps d'exécution.

Une autre extension sur laquelle nous avons commencé à travailler [Boudjeloud et Poulet, 2005] concerne l'utilisation d'un algorithme génétique interactif.

Références

[Aggarwal et Yu, 2001], Aggarwal C.C., Yu P.S. Outlier detection for high dimensional data, ACM Press New York, NY, USA, Periodical-Issue-Article, pp 37 - 46, 2001.

[Ankerst et al., 1999], Ankerst M., Breuning M., Kriegel H., and Sander J., Optics: Ordering points to identify the clustering structure. Proceedings, ACM SIGMOD International Conference on Management of Data (SIGMOD'99), pp 49-60, 1999.

[Arning et al., 1996] Arning A., Agrawal R., and Raghavan P., A linear method for deviation detection in large databases. KDD-96 Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp 164-169, 1996.

[Banzhaf et al., 1998] Banzhaf W., Francone F.D., Keller R.E. et Nordin P. Genetic programming: an introduction: on the automatic evolution of computer programs and its applications, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1998.

[Barnett et Lewis, 1994] Barnett V., Lewis T. Outliers in statistical data, John Wiley, 1994.

[Becker et al., 1987] Becker, R., Cleveland, W., Wilks, A. Dynamic graphics for data analysis, Statistical Science, 2, pp 355-395, 1987.

[Breunig et al., 2000] Breunig M.M., Kriegel H.P., Ng R.T., Sander J. LOF: Identifying density-based local outliers, Proceedings of the ACM SIGMOD 2000, International Conference On management of Data, 2000.

[Blake et Merz, 1998] Blake, C.L., Merz, C.J. UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[Boudjeloud et Poulet, 2004] Boudjeloud L., Poulet F., A genetic approach for outlier detection in high-dimensional data sets, in Modelling, Computation and Optimization in Information Systems and Management Sciences, Le Thi H.A., Pham D.T. Eds, Hermes Sciences Publishing, 2004, 543-550.

[Boudjeloud et Poulet, 2005] Boudjeloud L., Poulet F., Visual Interactive Evolutionary Algorithm for High Dimensional Data Clustering and Outlier Detection, in Advances in Knowledge Discovery and Data Mining: 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD'05, May 18-20, 2005, Hanoi, Vietnam, LNAI-3518, 426-431.

[Carr et al., 1987] Carr D. B., Littlefield R. J., Nicholson W. L. Scatterplot matrix techniques for large N, Journal of the American Statistical Association 82(398), pp 424-436, Littlefield, 1987.

[Cherkauer et Shavlik, 1996], Cherkauer KJ et Shavlik JW, Growing simpler decision trees to facilitate knowledge discovery, in proc. 2nd International Conference Knowledge Discovery & Data Mining KDD96, pp 315-318, AAAI Press, 1996.

[Ester et al., 1996] Ester M., Kriegel H.P., Xu X. Density Based Spatial Clustering of Applications with Noise, Proceedings, 2nd International Conference on Knowledge Discovery in Databases and Data mining, pp 226-231, Portland, OR, 1996.

[Fayyad et al., 1996] Fayyad U. , Piatetsky-Shapiro G., Smyth P. From Data Mining to Knowledge Discovery in Databases, AI Magazine Vol. 17, No. 3, pp 37-54, 1996.

[Goldberg, 1989] Goldberg D.E. Genetics Algorithms in Search, Optimisation and Machine Learning. Addison-Wesley, 1989.

[Guha et al., 1998] Guha, S., Rastogi, R., Shim, K. CURE: An efficient clustering algorithm for large databases. Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD-98), pp. 73-84, New York, 1998.

[Hawkins et al., 2002] Hawkins S., He H., Williams G., Baxter R. outlier detection using replicator neural networks. DAWAK'2002, pp 170-180, 2002.

[Hinneburg et al., 2000] Hinneburg A., Aggarwal C. C., Keim D. A. What is the nearest neighbor in high dimensional spaces. Proceedings of the VLDB Conference, 2000.

[Holland, 1975] Holland J. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, 1975.

[Inselberg, 1985] Inselberg A. The plane with Parallel Coordinates, special issue on computational geometry, The Visual Computer, Vol. 1, pp 69-97, 1985.

[Inselberg et Dimsdale, 1990] Inselberg A., Dimsdale B. Parallel coordinates: a tool for visualizing multi-dimensional geometry, visualization'90, San Francisco, pp 361-370, 1990.

[Jain et al., 1999] Jain A. K., Marty M. N., Flynn P. J. Data clustering: A review ACM Computing Surveys, Vol. 31, No. 3, pp 246-323, 1999.

[Jinyan et Huiqing, 2002] Jinyan L., Huiqing L. Kent ridge bio-medical data set repository, http://sdmc-lit.org.sg/GEDatasets. accédé en décembre 2004.

[Johnson et al., 1998] Johnson T., Kwok I., Ng R. T. Fast computational of 2 dimensional depth contours. In Poceeding KDD 1998, pp 224-228, 1998.

[Kohavi et John, 1997] Kohavi R. et John G. Wrappers for feature subset selection. Artificial Intelligence, N° 97, Vol. 1-2, pp 273-324, 1997.

[Knorr et al., 2000] Knorr E. M., Ng R. T., Tucakov V. Distance based outliers: algorithms and applications. VLDB Journal, Vol. 8, pp 237-253, 2000.

[Knorr et Ng, 1998] Knorr E., Ng R. Algorithms for mining distance-based outliers in large data sets. VLDB Conference Proceedings, September 1998.

[Kudo et Skalansky, 2000], Kudo M et Skalansky J. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1), 25-41, Jan. 2000.

[Michalewicz, 1996], Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, New York, Third edition, 1996.

[Mitchell, 1996], Mitchell M. An introduction to genetic algorithms, MIT Press, Cambridge, MA.

[Papadimitriou et al., 2003] Papadimitriou S., Kitawaga H., Gibbons P. B., Faloutsos C. LOCI: Fast Outlier Detection Using the Local Correlation Integral. ICDE'03, 19th

International Conference on Data Engineering, Sponsored by the IEEE Computer Society, 5- 8 March, 2003, Bangalore, India.

[Poulet, 2002] Poulet F., Cooperation Between Automatic Algorithms, Interactive Algorithms and Visualization Tools for Visual Data Mining, in proc. of VDM@ECML/PKDD-2002, 2nd International Workshop on Visual Data Mining, Helsinki, Aug.2002, pp. 67-79.

[Ramaswamy et al., 2000] Ramaswamy S., Rastogi R., Shim K. Efficient algorithms for mining outliers from large data sets. pp 427-438, SIGMOD Conference 2000.

[Reeves, 1995] Reeves C.R., A genetic algorithm for flowshop sequencing, computers and Operations Research, Vol. 22, No. 1, pp 5-21.

[Rocke et Woodruff, 1999] Rocke D. M. and Woodruff D. L. A Synthesis of Outlier Detection and Cluster Identification, Working Paper, University of California. 1999.

[Sander et al., 1998] Sander J. Ester M., Kriegel H.P., Xu X. Density Based Spatial Clustering of Applications with Noise: the algorithm GDBSCAN and its applications, Data mining and Knowledge Discovery, Vol. 2, pp 169-194, 1998.

[Williams et al., 2002] Williams G. J., Baxter R. A., He H., Hawkins S., Gu L. A Comparative Study of RNN for Outlier Detection in Data Mining, pp 709-712, ICDM 2002.

[Yang et Honavar, 1998], Yang, J. et Honavar, V. Feature Subset Selection Using a Genetic Algorithm. Invited chapter in: Feature Extraction, Construction, and Subset Selection: A Data Mining Perspective. Motoda, H. and Liu, H. (Ed.) New York: Kluwer. 1998.

[Zhang et al., 1996] Zhang T., Ramakrishnan R., Livny M. BIRCH: An Efficient Data Clustering Method for Very Large Databases. Proceedings ACM SIGMOD International Conference on Management of Data, pp 103-114, Montreal, Canada, 1996.

Summary

We present a hybrid outlier detection algorithm in high dimensional data sets. Usual outlier detection algorithms have a complexity increasing with the number of dimensions (columns) of the data set. To avoid this complexity, our approach uses a genetic algorithm to select a subset of "interesting" dimensions. The outlier detection algorithm is then applied on the subset of dimensions. Our algorithm is evaluated different data sets having very large number of dimensions and compared with other outlier detection algorithms. The reduction of dimension number also allows us to visualize the outlier to try to qualify it: it is probably an error or just an item very different from the other ones.