

# Un survol des algorithmes évolutionnaires dans la fouille de données

Fatima-Zohra Kettaf\*  
Jean-Pierre Asselin de Beauville\*\*

\*IRIT-UPS, UMR 5505, 118 route de Narbonne 31062 toulouse cedex 04 France  
kettaf@irit.fr

\*\*Laboratoire d'Informatique, Ecole Polytechnique, Université de Tours, 64 avenue  
Jean Portalis 37200 Tours, Détaché Temporaire à l'Agence Universitaire de la  
Francophonie au Canada (Montréal)  
jean-pierre.asselin@auf.org

**Résumé.** Cet article a pour objectif la présentation des travaux récents dans le domaine de la fouille de données, basés sur l'évolution génétique.

## 1 Introduction

La fouille de données peut se définir comme un ensemble de méthodes permettant d'analyser des données déjà collectées dans de très grandes bases de données (transactions bancaires, séquences biologiques, ...) afin d'extraire des relations ou des structures ayant une sémantique utile pour les utilisateurs.

Les algorithmes évolutionnaires (AE), par leur capacité d'exploration de grands espaces de solutions, se sont révélés être des outils utiles et efficaces dans le processus de fouille de données. On aborde, dans cet article, leur contribution aux phases : de pré-traitement, de découverte de règles et de post-traitement.

## 2 La fouille de données et le processus de découverte de connaissances

La fouille de données [?] peut se diviser en trois étapes : le pré-traitement, l'extraction de concepts, et le post-traitement. Les résultats sont généralement des règles permettant d'expliquer les données et/ou de faire des prédictions. Les entrées du processus sont souvent des données brutes qui vont subir des traitements de nature différente allant des procédures de bas niveau telles que la discrétisation et le filtrage, aux procédures de haut niveau, telles que l'extraction de concepts et de règles. Pour une application donnée, il existe plusieurs profils d'utilisateur en fonction des objectifs fixés. Ces objectifs sont généralement regroupés en cinq catégories : exploratoires, descriptifs, portés sur l'analyse de dépendances, prédictifs ou encore tournés vers la recherche par contenu. Pour l'exploration, les techniques associées font appel à l'interaction et à la visualisation. On utilise souvent l'analyse en composantes principales (ACP) pour réduire la dimension de l'espace de représentation et faciliter la visualisation. Pour ce qui concerne la description, il est souvent utile de supposer des modèles probabilistes,

ou des partitionnements en groupes homogènes pour présenter et résumer les données. D'autres aspects de ce processus analysent les relations de dépendance entre les attributs descriptifs et définissent des modèles capables de prédire la valeur d'un attribut en fonction des valeurs d'autres attributs (régression). On peut aussi prédire la classe d'une donnée par rapport à la connaissance des classes des autres données ou s'orienter vers la découverte de règles expliquant les concepts extraits. Enfin, certaines tâches visent la recherche par contenu comme celles implémentées dans les moteurs Google et BIC.

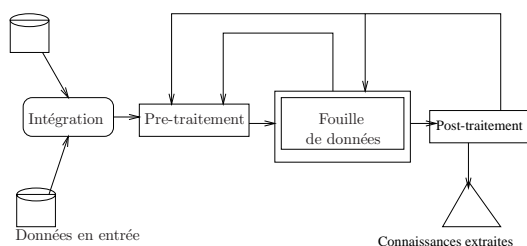


FIG. 1 – Une vue globale sur le processus de découverte de connaissances

La figure 1 montre les étapes du "processus de découverte de connaissances". Les algorithmes de fouille de données constituent le cœur de ce processus [?]. Avant de les appliquer, il convient d'appliquer un pré-traitement aux données. En fonction de leur nature, on peut effectuer soit l'ensemble des procédures d'intégration, de filtrage, de discrétisation et de sélection d'attributs ou se limiter uniquement à certaines d'entre elles. La dernière procédure est l'une des étapes qui a le plus impliqué la contribution des algorithmes évolutionnaires (AE) voir section 4.1. Après le pré-traitement les données sont prêtes à être utilisées pour la découverte de concepts et l'extraction de sémantique. Les différentes techniques associées sont issues du domaine de l'apprentissage et de l'intelligence artificiels et sont principalement la discrimination, la classification automatique, et la découverte de règles. Une fois les résultats obtenus, on pourra procéder à des post-traitements qui consistent, par exemple, à ne sélectionner parmi l'ensemble des règles retenues qu'un sous-ensemble de règles pertinentes.

### 3 Les algorithmes évolutionnaires

Ils regroupent les algorithmes génétiques (AG) [?], les stratégies d'évolution (SE), les programmes évolutionnaires (PE) [?] et la programmation génétique (PG) [?]. Ce sont des méthodes stochastiques d'optimisation, inspirées de l'évolution naturelle des espèces. Ils font évoluer une population de solutions potentielles, appelées "individus" et pouvant être représentées soit par des chaînes de bits (pour les AG) soit par des nombres (pour les SE et les PE) ou encore par des programmes (pour la PG). La population évolue à partir de trois opérateurs génétiques; la sélection, le croisement et la mutation. La sélection consiste à évaluer l'individu au regard du problème à résoudre pour décider de sa survie dans la génération suivante. Cette évaluation est appelée

”degré d’aptitude” ou ”fitness”. Le renouvellement de la population se fait par l’application de l’opérateur de croisement qui consiste à échanger des parties entre deux individus (parents) pour générer deux individus (enfants) héritant des caractéristiques de leurs parents. La mutation est un événement rare et consiste à perturber des positions dans les individus. Il existe bien évidemment plusieurs variantes de codage, et d’opérateurs génétiques. Elles dépendent fortement du problème à résoudre.

## 4 Le rôle des algorithmes évolutionnaires dans la fouille de données

Les AE s’utilisent dans presque toutes les étapes du processus de découverte de connaissances [?][?]. Une approche radicalement opposée consiste à proposer un cadre unifié pour leur utilisation [?]. On traite dans les sections suivantes les contributions des AE à la sélection de données, à la classification automatique, et à la discrimination pour la découverte de règles. On focalisera plus particulièrement sur la construction d’arbres de décision. On distinguera deux approches de découverte de règles : l’approche directe et l’approche hybride. On terminera par le post-traitement. Sont omises volontairement les contributions des AE dans les tâches de discrimination (au sens de l’utilisation des classifieurs tels que le classifieur de Bayes, les réseaux de neurones, les machines à vecteurs supports, les discriminateurs linéaires, et les approches basées sur les exemples ou le voisinage) et de recherche par contenu.

### 4.1 La sélection de données

Elle englobe deux types de techniques : la sélection de variables (attributs) [?][?][?] et la sélection de prototypes (exemples ou instances) [?]. Le génome manipulé par l’AG est donc un sous-ensemble de variables ou de prototypes. C’est une chaîne binaire de longueur égale au nombre initial de variables (prototypes), et dans laquelle la présence de la variable (prototype) est représentée par le bit 1 et son absence par le bit 0. Les variables comme les prototypes sont supposées indicées. Soit  $\{v_1, \dots, v_{12}\}$  l’ensemble des variables descriptives initiales, un génome ne retenant que les variables  $v_2$ ,  $v_7$  et  $v_{12}$  est représenté par la chaîne : 010000100001. Cette représentation offre l’avantage de pouvoir réutiliser tels quels les opérateurs classiques de croisement et de mutation. Si l’évaluation du génome se fait en dehors de l’apprentissage et de la généralisation du classifieur, l’approche utilisée est désignée comme étant une approche ”filter” ; elle procède comme un filtre. Si au contraire, l’évaluation se fait au sein d’un classifieur donné, l’approche est appelée ”wrapper” voir figure 2. Plus précisément, on fournit les variables retenues par ce génome à un algorithme de discrimination qui les utilisera dans ses phases d’apprentissage et de classement (exemples :  $(1 - PPV)$  [?] et arbres de décision Set-GEN [?], et GA-ID3 [?]). Punch *et al.* [?] utilisent une variante de ce codage en remplaçant les 0 et les 1 par un nombre compris entre 0 et 10. Ce code indique l’importance de la variable sélectionnée.

Dans leur approche Vafaie *et al.* [?] autorisent la combinaison des variables pour la création de nouvelles variables et les utilisent pour la construction d’un arbre de décision. Le lecteur trouvera dans [?] et [?] des études comparatives entre AG de

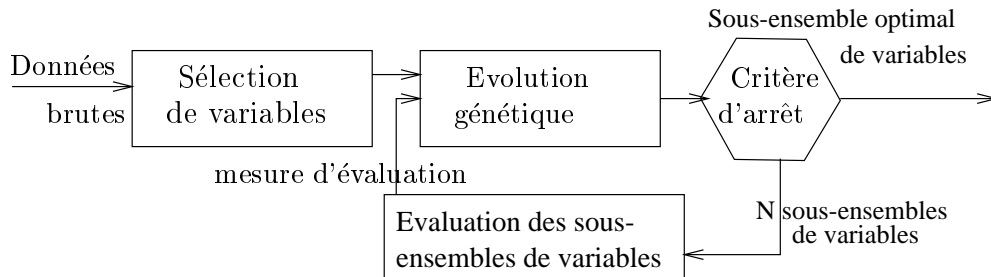


FIG. 2 – Exemple de système hybride de sélection de variables

sélection d'attributs. Moores [?] développe un nouveau système dans lequel sont combinées ACP et classifieur des plus proches voisins  $K - PPV$  (PCKaNN). Pappa *et al.* [?] utilisent un AG d'optimisation multi-objectif pour une sélection de variables combinée avec l'algorithme C4.5 de construction d'arbre de décision. Les deux objectifs fixés étaient l'obtention d'un arbre de taille minimum et d'un taux de bon classement élevé. Topchy *et al.* [?] proposent une nouvelle procédure de réduction de variables. La réduction du nombre de valeurs possibles prises par les variables, supposées ici nominales, se base sur des regroupements de valeurs proches. Après avoir trouvé les groupements de valeurs les plus significatifs, on remplace alors les valeurs dans chaque groupe par une seule valeur appelée meta-valeur.

Dans le contexte des techniques d'apprentissage multi-classifieurs [?], les AG ont été introduits dans une approche de sélection de variables pour un ensemble classifieurs [?] et ont dépassé en qualité les performances des méthodes de boosting et de bagging avec 80% de réduction sur les bases de données utilisées.

S'agissant de l'extraction de caractéristiques, les AE se sont aussi révélés très efficaces surtout dans le domaine de la fouille d'images. Plusieurs problèmes en analyse d'images (segmentation, détection de contours, recalage, ...) nécessitent la sélection de caractéristiques pertinentes pour la description des objets. La complexité des espaces de représentation et le manque d'informations sur les fonctions objectif rendent le problème d'optimisation très difficile. Les AE représentent alors une alternative intéressante pour ce type de problème, à titre d'exemple le lecteur pourra consulter [?].

## 4.2 La classification automatique

Les individus manipulés par les AE sont des partitions de l'ensemble de  $n$  données en  $c$  classes.  $c$  peut être connu ou non. Le premier codage [?] consistait à coder une partition par un chaîne de longueur  $n$ , chaque position est un numéro compris entre 1 et  $c$  et indique à quelle classe appartient l'instance. Un autre codage cette fois binaire et de longueur  $c * n$  alloue à chaque classe une sous-chaîne de longueur  $n$  : le bit 0 indique que l'instance courante n'appartient pas à la classe courante et 1 le contraire. Malheureusement, ces deux codages sont redondants. Un autre codage représente les

instances par leurs numéros et les sépare par des symboles pour délimiter les classes. Des solutions nouvelles proposent la représentation de la partition sous forme d'un ensemble de composantes connexes d'un graphe. La première est une représentation entière [?] et la seconde binaire [?]. Au lieu de coder des appartenances aux classes, des SE ont été utilisées pour faire évoluer des centres de classes. Les individus représentent une suite de coordonnées des  $c$  centres [?]. Pour plus de détails le lecteur pourra consulter [?].

### 4.3 La découverte de règles : approche directe

Il existe deux types de règles recherchées dans le domaine de fouille : les règles d'association et les règles prédictives. L'objectif est double et consiste à extraire des bases de données des concepts locaux ainsi que des règles permettant de les expliquer.

D'une façon plus générale, les règles d'association s'écrivent sous la forme :  $\theta \Rightarrow \phi$  où  $\theta$  est un ensemble fréquent (conjonction de faits) et  $\phi$  un "ensemble fréquent singleton". On définit  $freq(\theta)$  comme étant le nombre de cas où  $\theta$  est satisfait.  $freq(\theta \text{ et } \phi)$  est appelé support de  $\theta$ . On définit la confiance d'une règle  $c(\theta \rightarrow \phi)$  comme étant la proportion d'occurrences satisfaisant  $\phi$  parmi celles qui satisfont  $\theta$  :  $c(\theta \Rightarrow \phi) = \frac{freq(\theta \text{ et } \phi)}{freq(\theta)}$ . Les algorithmes de recherche d'ensembles fréquents et de règles d'association ne doivent retenir que des règles ayant une qualité au moins égale à certain seuil fixé par l'utilisateur, tout en effectuant le moins de passages possibles sur les données. L'algorithme le plus connu est "A Priori" [?]. Les trois sections suivantes caractérisent les AE de découverte de règles.

#### 4.3.1 Codage des règles

Il existe deux approches : Pittsburgh (GABIL)[?] qui préconise le codage de la connaissance (disjonction de règles, appelée aussi hypothèse) par un seul chromosome, tandis que l'approche Michigan (COGIN, REGAL)[?] associe à chaque règle un chromosome qui la représente.

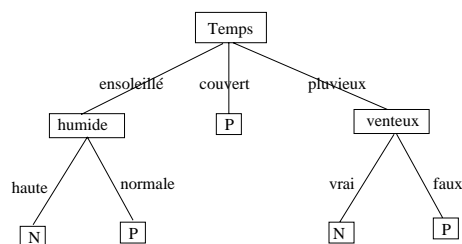


FIG. 3 – Un exemple d'arbre de décision

Le codage des règles est souvent binaire et consiste à juxtaposer le code des faits et à terminer par une sous-chaine représentant la conclusion. Prenons l'exemple "jouer au tennis" ; utilisé par Quinlan [?] dans son illustration pour la construction d'arbres de décision. Un chemin de l'arbre allant de la racine à une feuille est une règle de décision

formée par une conjonction de tests. Dans cet exemple, l'arbre obtenu est traduit par une disjonction de règles de production. La décision de jouer au tennis (représenté par des feuilles de l'arbre d'étiquette "P" et son contraire par l'étiquette "N") dépend des valeurs des attributs comme le temps, le vent, et l'humidité. Ils sont tous symboliques et prennent les valeurs indiquées dans la figure 3. Chaque attribut sera codé par autant de bits que de valeurs possibles. L'ordre retenu pour les attributs est le suivant : le temps, puis l'humidité et ensuite le vent. Par exemple la valeur "ensoleillé" de l'attribut temps sera codée par la chaîne 100. On peut coder le fait que le temps prenne soit la valeur "ensoleillé" soit "pluvieux" par 101. Si aucune contrainte n'est spécifiée pour cet attribut, c'est la chaîne 000 qui sera utilisée. La concaténation des faits traduit la partie condition de la règle. Sa partie conclusion est codée, dans cet exemple, par un seul bit (1 signifie "jouer au tennis" et 0 son contraire). Le génome 100.10.00|0||100.01.00|1||010.00.00|1||001.00.10|0||001.00.01|1 pourrait correspondre à un ensemble de règles de décision pour l'exemple sus-mentionné. Il constitue une solution potentielle dans l'approche Pittsburgh. Le génome 100.10.00|0 constituerait à lui seul une solution dans l'approche Michigan.

Dans les deux approches, le codage de la partie conclusion des règles pose un problème, surtout quand il s'agit de la classification d'instances.

Le codage des classes peut être vu selon trois approches :

- Dans la première [?][?], la partie conclusion est codée explicitement dans la règle et elle est sujette à évolution.
- Dans la deuxième [?][?], la partie conclusion est fixée d'avance, elle ne change pas. Ce qui nécessite plusieurs exécutions de l'AG et à fixer, chaque fois, la classe particulière que l'on recherche.
- Dans la troisième [?][?], on ne code pas la partie conclusion mais on l'estime avec les fréquences calculées à partir de l'ensemble d'apprentissage et de test.

Exemples

- GABIL [?], GIL [?], HDPDCS [?] : sont des systèmes de type Pittsburgh.
- COGIN [?], REGAL [?] : sont des systèmes de type Michigan.

Le choix entre les deux types de représentation dépend de la tâche à exécuter. La première pourrait être utilisée lorsque l'on recherche à prédire la classe des instances. Un AG faisant évoluer des arbres de décisions utilisera l'approche Pittsburgh. Si au contraire on recherche si un sous-ensemble de règles prédictives est de qualité élevée [?][?], c'est l'approche Michigan qu'il faut choisir. L'inconvénient de l'approche Michigan est de converger vers une seule solution (l'AG fournit la meilleure règle), alors que le but de la découverte des règles est l'extraction d'un ensemble de règles prédictives. Il est alors nécessaire d'introduire des techniques telles les "niches" pour que le système maintienne un ensemble de règles à chaque génération [?].

### 4.3.2 Les opérateurs génétiques

#### La sélection

Dans REGAL [?] par exemple, une sélection de type "suffrage universel" est utilisée. La qualité de la règle dans le contexte de l'apprentissage par la logique inductive, est

souvent fonction du nombre d'instances vérifiées par celle-ci (on dit aussi couvertes par la règle). Une sélection originale a été proposée dans REGAL : ce sont les exemples qui votent pour les règles qui les couvrent. Ils préfèrent bien sûr les meilleures. C'est un calcul de fréquence qui permet aux règles d'être élues. Ce procédé introduit la formation de niches (formation de sous-populations de règles par analogie avec les niches écologiques).

#### Opérateur de généralisation/spécialisation

Dans le domaine de l'apprentissage de règles, les algorithmes explorent l'espace des règles grâce à des opérateurs appelés "refinement operators", et qui sont les opérateurs de généralisation et de spécialisation de règles (ou d'hypothèses). L'opérateur de spécialisation choisit aléatoirement une partie du gène et change, dans cette partie, tout bit de '1' à '0' et inversement pour l'opérateur de généralisation.

Le fait d'ajouter une condition à la partie antécédent a aussi pour effet de spécialiser la règle (inversement pour la généralisation). Jourdan *et al.* [?] proposent un opérateur de généralisation qui supprime un terme avec une probabilité proportionnelle au nombre de termes de la partie condition par rapport au nombre maximum de termes ( $\frac{0.9(N_{termes}-1)}{Max_{termes}-1}$ ).

#### 4.3.3 Fitness ou la mesure de la qualité de la règle

La qualité ou l'utilité des règles se mesure souvent par un critère dérivé de la confiance. Les connaissances du domaine (à propos de  $\theta$  et  $\phi$ ) influent sur l'utilité de la règle. La mesure de Piatetsky-Shapiro, par exemple, exprime le déséquilibre qui peut apparaître dans la distribution des classes, elle est fonction du support de la règle et des deux nombres d'instances couvertes respectivement par les parties condition ( $\theta$ ) et conclusion ( $\phi$ ) :  $PS(\theta \Rightarrow \phi) = freq(\theta \wedge \phi) - \frac{freq(\theta)freq(\phi)}{n}$ , où  $n$  est le nombre total d'instances ou d'items. On remarque bien que cette mesure n'est appropriée que pour la découverte de règles prédisant des classes minoritaires. Un critère purement statistique appelé la J-mesure est :

$$J(\theta \Rightarrow \phi) = p(\theta)p(\phi|\theta) \log \frac{p(\phi|\theta)}{p(\phi)} + [(1 - p(\phi|\theta)) \log \frac{1 - p(\phi|\theta)}{1 - p(\phi)}]$$

Le premier terme  $p(\theta) = \frac{freq(\theta)}{n}$  favorise les règles générales alors que le second compare la distribution a priori de  $\phi$  et sa distribution a posteriori étant donnée  $\theta$ . Des valeurs élevées de cette mesure peuvent aussi résulter de règles sans intérêt, puisqu'une grande différence entre la distribution a priori et a posteriori augmente aussi cette mesure. Wang *et al.* [?] proposent une mesure modifiée pour pallier ce problème :

$$J_1 = p(\theta)p(\phi|\theta) \log \left[ \frac{p(\phi|\theta)}{p(\phi)} \right]$$

Dans le tableau 1 on définit  $TP$  comme étant le nombre d'instances satisfaisant  $\theta$  et  $\phi$ ,  $FP$  comme celui des instances satisfaisant  $\theta$  et non  $\phi$ ,  $FN$  comme celui des exemples non couverts par  $\theta$  mais couverts par  $\phi$  et enfin,  $TN$  comme le nombre d'exemples qui ne sont couverts, ni par  $\theta$ , ni par  $\phi$ . En réécrivant le coefficient de confiance de la règle en fonction de ces quantités, on obtient  $c(\theta \Rightarrow \phi) \equiv CF = \frac{TP}{TP+FP}$

|              | $\phi$                          | $\neg\phi$                          |
|--------------|---------------------------------|-------------------------------------|
| $\theta$     | $freq(\theta et \phi) = TP$     | $freq(\theta et \neg\phi) = FP$     |
| $\neg\theta$ | $freq(\neg\theta et \phi) = FN$ | $freq(\neg\theta et \neg\phi) = TN$ |

TAB. 1 – *Matrice de confusion*

Les valeurs élevées de  $TP$  et  $TN$  indiquent une bonne qualité de la règle. Pour tenir compte de la complétude de la règle, on utilise  $Comp = \frac{TP}{TP+FN}$  et enfin on propose  $CF * Comp$ .

On peut également utiliser des combinaisons linéaires du coefficient de confiance, de la complétude et de la simplicité des règles [?][?][?]. Noda *et al.* [?] proposent d'ajouter l'intérêt de la règle dans le coefficient de confiance. La J-mesure et  $J_1$  peuvent être utilisées à la place du coefficient de confiance de la règle. Malheureusement ces fonctions entraînent une convergence lente. Araujo *et al.* [?] modifient la mesure  $J_1$  en une mesure  $F$  :

$$F = \frac{w_1 J_1 + w_2 (\frac{N_{pu}}{NT})}{w_1 + w_2}$$

$N_{pu}$  est le nombre d'attributs utiles de la partie  $\theta$ . On appelle "attribut utile", un attribut apparaissant dans la règle avec une valeur donnée et tel qu'au moins une instance dans les données présente la valeur indiquée par la règle pour cet attribut ainsi que la même valeur pour l'attribut but de cette règle.  $NT$  est le nombre total d'attributs présents dans la partie condition  $\theta$ . Les valeurs préconisées de  $w_1$  et  $w_2$  sont respectivement 0.6 et 0.4. Melab *et al.* [?] remarquent que le rapport  $\frac{N_{pu}}{NT}$  ne change pas à partir d'un certain nombre de générations  $g_n$ . C'est pour cette raison que Jourdan *et al.* [?] proposent la fonction suivante :

$$F' = \begin{cases} \frac{w_1 J_1 + (w_2 \frac{N_{pu}}{NT})}{w_1 + w_2} & \text{si } g < g_n \\ J_1 & \text{sinon} \end{cases}$$

D'autres combinaisons avec des coefficients qui tiennent compte des jugements des utilisateurs ont été proposées dans [?][?].

Jourdan *et al.* [?] élaborent un AG de recherche de règles d'association et l'appliquent à un problème de déséquilibre de liaison afin de découvrir des haplotypes candidats à l'explication des maladies multifactorielles comme le diabète et l'obésité. L'AG qu'ils utilisent est sans doublons, et travaille en sous-populations. C'est d'ailleurs l'adaptation qu'il faut assurer pour que l'AG découvre un ensemble de règles et ne converge pas vers une seule règle comme dans l'approche Pittsburgh (voir paragraphe 4.3.1). Les individus d'une même sous-population (niche) ont le même attribut (soit avec la même valeur soit avec des valeurs différentes). Ils sont codés en une suite de gènes de type (attribut valeur attribut), et ne sont pas nécessairement de même longueur. La fonction de fitness mesurant la qualité des règles d'association qu'ils utilisent est  $F'$ .

#### 4.4 La découverte de règles : approche hybride

Carvalho *et al.* [?] proposent une méthode hybride de découverte de règles prédictives. Ils utilisent un AG et un algorithme de construction d'arbre de décision dans le but



de découvrir des règles de classification à partir d'un grand ensemble de données. Ils justifient leur choix (hybridation) par le fait que les arbres de décision sont plus appropriées pour la découverte de règles longues et sont moins efficaces dans la recherche de règles courtes. Ils commencent par appliquer l'algorithme C4.5 pour aboutir à un arbre de décision qu'ils élaguent et traduisent en un ensemble de règles propositionnelles. Ils séparent ensuite les règles en deux ensembles disjoints ; un ensemble de règles courtes et un ensemble de règles longues. L'AG a pour tâche, la découverte de données provenant du premier groupe et l'algorithme C4.5 le classement des données du deuxième groupe. Cantu-Paz *et al.* [?] combinent SE et AG pour la création d'arbres de décision obliques.

Une approche originale a été proposée dans Papagelis *et al.* [?] et qui consiste à faire évoluer directement des arbres de décision (contrairement à l'hybridation) grâce à des opérateurs génétiques adaptés aux structures d'arbres.

La programmation génétique (PG) offre un champ d'application beaucoup plus large que les AG, les SE et les PE dans le processus de fouille. Leur capacité à combiner les attributs, par exemple, peut se révéler très efficace pour les tâches prédictives ou d'extraction d'attributs [?]. Appliqués dès leur apparition au problème d'apprentissage [?], ils ont montré leur supériorité par rapport aux classifieurs traditionnels et leur efficacité surtout dans la construction d'arbres de décision [?]. On trouve aussi dans [?] un système hybride AG/PG qui combine sélection et extraction d'attributs pour reconnaître des visages.

L'apparition de nouveaux types de PG à contraintes syntaxiques [?] accompagnés d'opérateurs génétiques adaptés et de PG incorporant les connaissances du domaine dans leur grammaire a permis de découvrir des règles prédictives très intéressantes [?].

## 4.5 Le post-traitement

On se restreint ici au post-traitement qu'on applique aux règles d'association et à l'élagage des arbres de décision. Le but est d'extraire le sous-ensemble de règles le plus performant au regard des critères que l'on se fixe, et qui sont généralement ; la compréhensibilité de la règle (simplicité) et son intérêt. Il existe des critères subjectifs donnés par l'utilisateur et dépendants du domaine. L'utilisateur peut guider le système en lui spécifiant des patrons de règles d'association ainsi qu'une liste ou une combinaison d'attributs qu'il souhaite obtenir dans les règles [?]. On trouve dans [?][?][?] des exemples de critères objectifs.

## 5 Conclusion

La fouille de données n'a été reconnue comme un véritable axe de recherche que depuis moins de dix ans. Cet axe implique plusieurs disciplines scientifiques : les statistiques, l'intelligence artificielle, l'apprentissage artificiel, les bases de données, et d'autres domaines. Il reste beaucoup de problèmes à résoudre et plusieurs questions restent ouvertes. Une des orientations actuelles dans la découverte de règles est d'utiliser les programmes logiques inductifs. Ces programmes sont dotés d'un pouvoir d'explication basé sur la logique du premier ordre, ils sont actuellement très utilisés dans les bases de données biochimiques pour expliquer et trouver les différentes structures des

molécules. D'ailleurs, plusieurs travaux lient la PG et la programmation logique inductive (PLI) [?]. Cet article a porté plus sur la contribution des AG que sur celle de la PG. Pour compléter ce survol, le lecteur pourra consulter [?][?]. Les AG s'avèrent très utiles lorsque le problème nécessite une optimisation multi-critères. Ils sont malheureusement de complexité algorithmique élevée même si elle est atténuée par la baisse des prix et par l'augmentation des performances matérielles. Ils ont toutefois une bonne capacité à intégrer les connaissances expertes sous forme d'opérateurs génétiques [?] et se prêtent bien à l'hybridation [?]. Une des voies, à notre sens, prometteuse réside dans l'utilisation des AE interactifs pour le choix d'algorithmes de fouille appropriés au problème à résoudre et aux données à traiter.

## Summary

The aim of this article is to provide a survey on recent contributions of genetic approaches in the field of data mining .