

Un survol des algorithmes biomimétiques pour la classification

Hanene Azzag*, Fabien Picarougne*
Christiane Guinot**, Gilles Venturini*

*Laboratoire d'Informatique de l'Université de Tours,
École Polytechnique de l'Université de Tours - Département Informatique,
64, Avenue Jean Portalis, 37200 Tours, FRANCE.
{hanene.azzag,fabien.picarougne}@etu.univ-tours.fr, venturini@univ-tours.fr
<http://www.antsearch.univ-tours.fr/webrtic>

**CE.R.I.E.S.

20 rue Victor Noir, F-92521 Neuilly-sur-Seine Cedex.
christiane.guinot@ceries-lab.com

Résumé. Nous présentons dans cet article un survol des algorithmes et méthodes biomimétiques pour résoudre le problème de la classification. Nous décrivons les approches utilisant les algorithmes génétiques et évolutionnaires avec les différents codages et représentations ayant été utilisés. Nous abordons l'approche à base de fourmis artificielles qui se trouve être une riche source d'inspiration pour la classification. Nous détaillons finalement d'autres approches à base d'agents avec notamment l'intelligence en essaim (nuages d'agents) et avec les systèmes immunitaires. Enfin, nous résumons les ressemblances et différences des travaux présentés et nous concluons sur les perspectives liées à l'approche biomimétique pour la classification.

1 Introduction

Le problème de la classification de données est identifié comme une des problématiques majeures en extraction des connaissances à partir de données. Depuis des décennies, de nombreux sous-problèmes ont été identifiés, comme par exemple la sélection des données ou des descripteurs, la variété des espaces de représentation (numérique, symbolique, etc), l'incrémentalité, la nécessité de découvrir des concepts, d'obtenir une hiérarchie, etc. La popularité, la complexité et toutes ces variantes du problème de la classification de données ont donné naissance à une multitude de méthodes de résolution. Ces méthodes peuvent à la fois faire appel à des principes heuristiques ou encore mathématiques. Parmi celles-ci, il existe une branche qui s'inspire plus spécialement de principes issus de la biologie. Les motivations des chercheurs sont d'une part de tester de nouveaux algorithmes sur le problème de la classification et de connaître leurs apports. Mais elles sont aussi de proposer de nouvelles sources d'inspiration, car le problème de la classification se rencontre souvent chez les animaux et dans les systèmes biologiques.

Nous allons donc donner un aperçu de ces méthodes. Nous ne traiterons pas ici les approches neuronales mais plutôt les approches à base de population d'agents (algo-

algorithmes évolutionnaires, fourmis artificielles, intelligence en essaim, systèmes immunitaires). Nous allons considérer par la suite un ensemble de n données d_1, \dots, d_n à regrouper en classes. Nous ne ferons pas plus d'hypothèses à propos de la représentation des données ou de la forme de la classification désirée. Nous allons commencer dans la section 2 par détailler les approches génétiques qui manipulent une population de classifications candidates et qui les font évoluer en utilisant les principes de la sélection naturelle. Ensuite, nous abordons dans la section 3 la manière dont les fourmis artificielles sont appliquées à ce problème : chaque fourmi va intervenir sur une partie de la classification en cours de construction, avec des modèles très diversifiés qui dénotent une richesse importante de ce domaine. Dans la section 4, nous détaillons des approches moins connues mais qui apportent néanmoins leur potentiel à notre problématique : d'une part les déplacements sociaux d'une population d'agents permettent de créer des groupes, et d'autre part l'utilisation des systèmes immunitaires qui vont répondre aux stimulations d'antigènes (les données) en produisant des anticorps (les éléments de la structure classificatoire). Enfin, nous donnons dans la section 5 une discussion sur les méthodes présentées ainsi que les perspectives que nous pouvons déduire des différents travaux en cours. Compte tenu du nombre de pages limité, nous ne citons pas d'articles fondateurs ou d'introduction à la classification ou aux méthodes biomimétiques, de même pour les travaux de biologie sous jacents aux modèles informatiques.

2 Approches évolutionnaires

2.1 Quatre catégories d'algorithmes

Dans les années 70, les premiers travaux sur l'évolution artificielle ont concerné les algorithmes génétiques (AG), les stratégies d'évolution (SE) et la programmation évolutive (PE). Ces trois types d'algorithmes ont utilisé des principes globalement communs car ils se sont tous inspirés des mêmes principes du neo-darwinisme : utilisation d'une population d'individus (dans notre cas chaque individu représente une classification des données), évaluation des individus par une fonction, sélection des meilleurs et génération d'une nouvelle population avec des opérateurs de croisement et de mutation. Cependant, des choix méthodologiques ont initialement opposé ces méthodes. Ainsi, les premiers AG utilisaient plutôt un codage binaire des individus alors que les SE utilisaient un codage en nombre réel. Ensuite, dans les années 90 est apparue la programmation génétique (PG) qui introduit notamment des représentations arborescentes.

Pour toutes ces approches, la représentation va également imposer des opérateurs particuliers pour engendrer de nouvelles solutions. Par exemple, l'un des principes fondamentaux des AG étant d'utiliser un opérateur de croisement combinant utilement les gènes de deux individus, le problème posé est alors de définir des opérateurs de croisement permettant l'échange de caractéristiques entre deux classifications. Les SE utilisent plutôt des mutations à base de lois gaussiennes qui vont modifier les paramètres réels d'un individu.

2.2 Algorithmes génétiques

Les premiers travaux proposant un AG (et plus généralement un algorithme évolutionnaire) pour le problème de la classification sont dus à [Raghavan et Birchard, 1979]. Le nombre de classes est fixé à l'avance et la représentation de longueur n associe une classe à chaque donnée, comme dans l'illustration suivante :

Données	d_1	d_2	...	d_n
Classes ($\in [1,k]$)	3	1	...	4

Les opérateurs génétiques sont une adaptation directe des opérateurs génétiques binaires pour le cas d'un individu représenté par une chaîne n -aire (croisement avec un point de coupure). Par exemple, le croisement à un point échange des étiquettes de classe entre deux individus. Cet opérateur peut donc faire disparaître des classes. Seule la mutation peut faire apparaître de nouvelles classes. La fonction d'évaluation consiste à minimiser une erreur quadratique. Notons que ce codage est utilisé également par [Hansohm, 2000] dans le contexte du bipartitionnement : le premier vecteur d'entiers code la partition sur les données, le deuxième sur les variables.

De nouveaux codages utilisant des permutations ont été introduits par plusieurs auteurs. Dans [Jones et Beltramo, 1991], un premier codage consiste à utiliser une permutation des n données représentées par leur indice en ajoutant en plus des symboles servant de séparateurs : par exemple pour $n = 6$, l'individu $(2,4,-,5,1,3,-,6)$ représente un partitionnement en 3 classes. Pour obtenir k classes, le nombre de séparateurs introduits est égal à $k - 1$. Un autre codage à base de permutation proposé dans le même travail consiste à utiliser deux parties dans un même individu. Les prototypes sont codés sur la première partie de la partition puis la suite de la partition représente un ordre sur la manière d'affecter les données restantes à ces prototypes. Dans [Bhuyan *et al.*, 1991], ce type de permutation est utilisé uniquement pour fixer un ordre sur les données : un algorithme heuristique décide alors comment construire la partition en "coupant" la permutation en des endroits judicieux. Pour ces codages sont utilisés des opérateurs génétiques de croisement définis dans le cadre du problème du voyageur de commerce (OX et PMX, voir [Goldberg, 1989]).

Un autre codage alternatif consiste non pas à représenter une classe pour chacun des objets dans un individu de longueur n , mais plutôt k prototypes de ces classes [Lucasius *et al.*, 1993]. Ces k prototypes sont choisis parmi l'ensemble des n données. Ainsi, un individu devient alors un vecteur d'indices de prototype :

Prototypes	p_1	p_2	...	p_k
Données ($\in [1,n]$)	120	40	...	55

Ensuite, pour calculer la partition résultante, les données sont affectées à chaque prototype de classe sur la base d'un algorithme de type plus proche voisin. Dans cette représentation, les opérateurs génétiques classiques peuvent poser des problèmes : à la suite d'un croisement, un même prototype peut se retrouver deux fois dans un individu.

Le codage introduit dans [Bezdek *et al.*, 1994] consiste à représenter le partitionnement à l'aide d'une matrice M booléenne de type classe \times données. $M(i,j)$ prend

la valeur 1 ($i \in [1,k], j \in [1,n]$) si la donnée d_j appartient à la classe i , 0 sinon. Un individu est donc de la forme :

Données/Classes	c_1	c_2	...	c_k
d_1	0	1	...	0
d_2	0	0	...	1
...
d_n	1	0	...	0

Dans cette représentation, l'opérateur de croisement est défini cette fois en 2D. Un point important à noter dans cette représentation est la possibilité de la généraliser à des classifications recouvrantes ainsi que floues : dans le premier cas plusieurs 1 peuvent apparaître sur une même ligne, dans le deuxième les valeurs ne sont plus booléennes mais représentent des degrés d'appartenance.

Un codage permettant de manipuler directement des groupes a été proposé dans [Falkenauer, 1994]. Une classification est constituée de n gènes représentant la classe de chaque donnée (comme dans le premier codage introduit dans cette section), suivis de la liste des groupes apparaissant dans l'individu (par exemple si les données appartiennent à trois classes, l'individu finit par (3, 2, 1)). Cette représentation utilise donc un opérateur de croisement permettant d'échanger directement des groupes. L'opérateur de mutation agit également au niveau des groupes (éclatement, regroupement, etc) avec des heuristiques locales (réaffectation des données isolées).

Dans [Greene, 2003] a été développé à notre connaissance le seul AG apprenant une classification hiérarchique présentée sous la forme d'un arbre de centroïdes. Cet algorithme est restreint aux données numériques mais ne fait pas d'hypothèses sur le nombre de classes.

2.3 Autres approches évolutives

Les trois autres catégories d'algorithmes évolutives ont été nettement moins développées que les AG. Par exemple, dans [Babu et Murty, 1994], les SE ont été utilisées avec un codage matriciel : chaque colonne du tableau représente un centre de classe de la même dimension que l'espace numérique de description des données. La PE a également été utilisée mais dans un seul travail à notre connaissance [Fogel et Simpson, 1993]. Également, nous n'avons pas trouvé d'articles traitant du problème général de la classification avec la PG.

Plusieurs approches hybrides ont cependant été proposées en utilisant conjointement les AG avec des approches plus classiques comme K-Means ou encore Fuzzy-C-Means. Ces heuristiques sont utilisées par exemple juste après l'AG qui sert donc à trouver une bonne partition initiale [Babu et Murty, 1993]. Elles peuvent également servir au même titre que les opérateurs génétiques dans la boucle de l'AG [Krishna et Murty, 1999] : elles sont appliquées sur chaque individu. Cela permet d'accélérer la convergence des AG tout en conservant les avantages d'une méthode globale. Ces hybridations restent cependant liées aux données numériques.

3 Fourmis artificielles

3.1 Quelques principes

Les fourmis réelles ont inspiré les chercheurs en informatique dans de nombreux domaines. Cela se justifie particulièrement quand on connaît la richesse comportementale de ces animaux. L'un des modèles les plus connus (ACO pour Ant Colony Optimization) a été introduit par [Colorni *et al.*, 1991] initialement dans le cadre du problème du voyageur de commerce. Les fourmis utilisent des phéromones pour marquer des arcs entre les villes. Ces phéromones représentent en fait une distribution de probabilités qui est mise à jour en fonction des résultats observés (longueur totale du chemin par exemple). Cette approche a été depuis largement développée et appliquée à de nombreux problèmes d'optimisation combinatoire et numérique. Un survol de ces articles ne rentre cependant pas dans le cadre de notre étude puisque ce modèle n'a pas été utilisé à notre connaissance pour résoudre le problème de la classification (voir cependant [Alexandrov, 2000] mais apparemment aucune suite n'a été donnée à ces travaux). Pourtant, les sections suivantes vont montrer que le modèle des fourmis artificielles est très riche dans ce domaine. La raison vient du fait que d'autres comportements observés chez les fourmis peuvent être directement mis en relation avec le problème de la classification, à commencer par le tri du couvain.

3.2 Les travaux fondateurs

Ces travaux datent des années 1990. Il s'agit d'abord des travaux de biologistes s'intéressant de près à la modélisation des fourmis en termes mathématiques et informatiques, et à l'utilisation concrète de ces modèles. Deneubourg apparaît donc comme un pionnier dans le domaine du tri d'objets par des fourmis artificielles. Dans [Deneubourg *et al.*, 1990], il propose avec ses collègues les principes suivants : des fourmis artificielles se déplacent sur un plan. Les objets à rassembler sont répartis sur ce plan. Une fourmi ne dispose que d'une perception locale de ces objets et ne communique pas avec les autres. Au lieu de cela, la configuration des objets sur le sol va influencer leurs actions. Lorsqu'une fourmi rencontre un objet, elle le ramasse avec une probabilité $\frac{c_1}{c_1+f}$, où f représente la fréquence de rencontre d'objets dans un passé récent. Autrement dit, plus une fourmi rencontre d'objets, moins elle a de chance d'en prendre un (elle se trouve dans une zone avec beaucoup d'objets). Ensuite, une fois un objet ramassé, la fourmi se déplace au hasard dans le plan, et elle dépose l'objet avec une probabilité $\frac{f}{c_2+f}$. Cette probabilité est d'autant plus grande que la fourmi a rencontré récemment des objets. Ces principes relativement simples font qu'il apparaît des regroupements d'objets. L'approche peut être généralisée à plusieurs types d'objets (les fréquences f sont spécifiques à chaque type d'objets) : cet algorithme permet alors de trier des objets.

Le pas qui sépare le tri d'objets de la classification a ensuite été franchi dans [Lumer et Faieta, 1994]. Ils ont adapté l'algorithme présenté précédemment (voir figure 1) : les données sont initialement réparties aléatoirement sur une grille 2D. Chaque fourmi est située dans une case de cette grille et ne perçoit que les données situées dans son voisinage (8 voisins par exemple). Ensuite la fréquence f utilisée dans l'algorithme

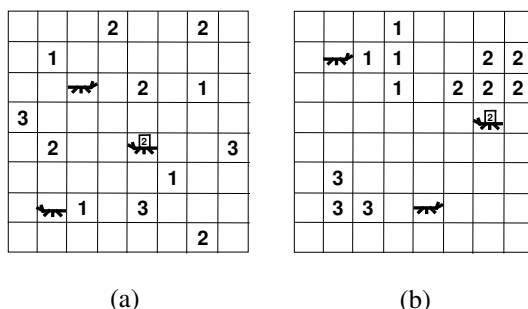


FIG. 1 – Principe de la classification de données par des fourmis artificielles selon l'algorithme présenté par [Lumer et Faieta, 1994]. En (a) les objets sont répartis aléatoirement sur la grille. Les fourmis peuvent s'en saisir et les déposer dans des cases où la densité d'objets similaires est élevée. Il en résulte la formation de groupes comme en (b).

de tri vu précédemment peut être remplacée par une moyenne des similarités entre une donnée d_i portée par une fourmi et les données d_j situées dans son voisinage. Une donnée d_i sur la grille est ramassée avec une probabilité d'autant plus grande qu'elle est peu similaire aux données voisines. De la même manière, une donnée d_i portée par une fourmi est plus facilement déposée dans une région comportant des données qui lui sont similaires. Cet algorithme a été depuis étendu à d'autres applications comme le partitionnement de graphes [Kuntz *et al.*, 1997] ou la classification de sessions sur des sites Web [Abraham et Ramos, 2003].

3.3 Approches récentes

Une extension de l'algorithme de [Lumer et Faieta, 1994] a été présentée dans [Monmarché *et al.*, 1999]. D'une part, les fourmis peuvent empiler les objets les uns sur les autres dans une même case de la grille. Lorsqu'elles rencontrent un tas d'objets, elles peuvent ainsi se saisir de l'objet le plus dissimilaire. D'autre part, une hybridation a été effectuée avec l'algorithme des K-Means. Cette hybridation consiste à utiliser la séquence d'algorithmes suivante: AntClass, K-Means, AntClass, K-Means. AntClass fournit une partition initiale, les K-Means corrige des erreurs d'AntClass qui mettrait beaucoup plus de temps à être corrigée avec AntClass seul.

Dans [Labroche *et al.*, 2002] a été introduit un nouveau modèle à base de fourmis pour la classification utilisant le système d'identification chimique des fourmis. Celui-ci est fondé sur la construction d'une odeur coloniale qui est le fruit des apports génétiques, environnementaux et comportementaux. Cette odeur est construite par les individus pour identifier qui fait partie du groupe et qui doit être rejeté. A partir de ce modèle, un nouvel algorithme de classification a été proposé dans lequel chaque donnée est une fourmi dont l'odeur est déterminée par les valeurs prises par les attributs décrivant cette donnée. Les fourmis effectuent des rencontres aléatoires et décident d'appartenir au même groupe ou non. Il en résulte l'établissement d'une classification.

Enfin, dans [Azzag *et al.*, 2003], a été introduit un nouveau modèle permettant d'ef-

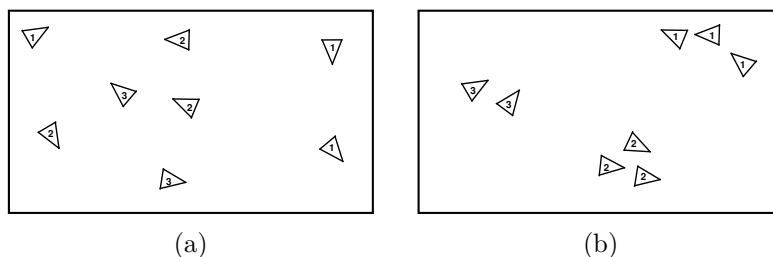


FIG. 2 – *Principes utilisés pour la classification par nuages d’agents. Les agents sont placés initialement avec des coordonnées et des vecteurs vitesse aléatoires (voir (a)). Les mouvements d’un agent dépendent des autres agents perçus dans son voisinage et des similarités entre les données qu’ils représentent. Le comportement local de chaque agent tend à former globalement des groupes d’agents similaires se déplaçant de manière cohérente (voir (b)).*

fectuer rapidement une classification hiérarchique. Il s’agit de copier la manière dont les fourmis construisent des structures vivantes en s’accrochant les unes aux autres en fonction de critères locaux (la forme de la structure influençant le comportement d’accrochage ou de décrochage). Dans ce modèle, chaque fourmi artificielle représente une donnée. Les fourmis sont placées initialement à la racine de l’arbre et vont pouvoir se déplacer dans cet arbre et s’accrocher afin de construire une structure hiérarchique dont chaque noeud représente une donnée. L’objectif est de construire automatiquement un site portail (données textuelles) et d’obtenir la propriété suivante : chaque noeud o de l’arbre est une catégorie composée de toutes les données portées par les sous-arbres de o . Les sous catégories (représentées par les noeuds connectés à o) doivent être très similaires à leur mère dans l’arbre, mais également les plus dissimilaires entre elles. Les résultats obtenus sont très compétitifs par rapport à la classification ascendante hiérarchique notamment.

4 Autres approches

4.1 Intelligence en essaim

L’intelligence en essaim (“swarm intelligence”) regroupe de nombreux algorithmes à base de population d’agents. Les fourmis artificielles en font partie mais nous nous intéressons dans cette section à des algorithmes plus spécifiques qui utilisent les déplacements d’un essaim d’agents pour résoudre un problème. A titre d’exemple, les algorithmes PSO (“particle swarm optimization”) utilisent un ensemble de particules caractérisées par leur position et leur vitesse pour maximiser une fonction dans un espace de recherche. Des interactions ont lieu entre les particules afin d’obtenir des comportements globaux efficaces.

Dans la biologie, de nombreux chercheurs se sont intéressés à la manière dont les animaux se déplacent en groupe. Aucun individu ne contrôle les autres mais pourtant des formes et des comportements complexes peuvent apparaître lors de ces déplacements.

[Reynolds, 1987] a été probablement le premier à proposer une utilisation informatique de tels modèles, simulations qui sont utilisées notamment dans l'industrie du cinéma pour donner des mouvements réalistes à des groupes d'individus. Dans ces travaux, chaque individu évolue dans un espace 3D. Il est donc caractérisé par sa position et sa vitesse. Un individu perçoit les autres dans un voisinage donné. Des règles comportementales généralement simples permettent aux individus de se déplacer en groupe, d'éviter des obstacles, etc.

En 1998, ces principes ont été appliqués pour la première fois à un problème de classification [Proctor et Winter, 1998] (voir figure 2). Les agents représentent chacun une donnée. Un agent réagit aux autres agents présents dans son voisinage en tenant compte de la similarité des données. Un agent se déplacera plutôt vers des données qui lui sont similaires. Cette règle comportementale permet donc de former des groupes de données similaires.

Dans [Monmarché *et al.*, 2002], cet algorithme a été amélioré et évalué d'une manière plus systématique. Une distance idéale entre individus est définie, distance qui dépend de la similarité entre les données. Un critère d'arrêt est utilisé également en mesurant l'entropie spatiale du nuage d'agents. Cet algorithme a été intégré dans un système de fouille visuelle de données utilisant la réalité virtuelle.

4.2 Systèmes immunitaires

Les systèmes immunitaires (SI) sont un ensemble de modélisations du système immunitaire humain et animal appliqués à différents problèmes en informatique. Ils utilisent les principes suivants : des agents (lymphocytes) qui génèrent des anticorps vont apprendre à reconnaître le soi du non-soi (les antigènes). Pour cela, ces agents doivent d'abord être engendrés en utilisant un principe de composition de briques élémentaires. Ensuite, ils subissent un test de sélection (dit de sélection négative) : les agents rejetant le soi sont éliminés, et les autres, qui vont rejeter le non-soi, sont gardés. Chaque fois qu'il y a reconnaissance d'un antigène par un anticorps, la présence des lymphocytes générant ces anticorps est favorisée par un processus de sélection par clonage et par la disparition des lymphocytes non stimulés par les antigènes. Ce clonage donne donc lieu à des interactions entre les lymphocytes et peut mettre en oeuvre des mutations. Certains lymphocytes, lorsqu'ils sont souvent utilisés, prennent un rôle d'élément de mémorisation à long terme. Ces systèmes disposent de propriétés complexes car ils sont capables de générer des solutions et de les sélectionner en fonction de leur efficacité selon des heuristiques originales.

En ce qui concerne la classification, les principes des systèmes immunitaires sont, à un niveau général, les suivants (voir par exemple le système aiNet [de Castro et Von Zuben, 2000]) : les données d_1, \dots, d_n représentent les antigènes. Ces antigènes sont présentés itérativement au système jusqu'à l'obtention d'une condition d'arrêt. On suppose que les données sont numériques, et donc qu'un antigène est un vecteur de dimension n . A chaque itération, l'antigène présenté va activer des anticorps (assimilés dans cette modélisation à des lymphocytes-B). Un anticorps est également représenté par un vecteur de dimension n . Les anticorps suffisamment proches de l'antigène (au sens de la distance euclidienne) vont subir des clonages avec mutation (interaction anticorps/antigènes) afin d'amplifier et d'affiner la réponse du système. Egale-

ment, ces anticorps vont subir une sélection (interaction anticorps/anticorps) : ceux qui sont trop proches les uns des autres seront diminués en nombre. Après ces itérations, le système converge en plaçant des anticorps (qui agissent comme des détecteurs) de manière judicieuse et en nombre adapté aux données.

D'autres modèles plus complexes existent. Ainsi dans [Knight et Timmis, 2002] le système utilise plusieurs niveaux de cellules et d'interaction (anticorps, lymphocytes, cellules de mémorisation). Dans [Nasaroui *et al.*, 2002], ce même système est généralisé et amélioré en utilisant des fonctions d'appartenance floue plutôt qu'une distance euclidienne et un seuil.

5 Discussion et perspectives

Il ressort de cette étude des points saillants que l'on peut résumer comme suit. Il est certain que les AG pour la classification à eux seuls ont fait l'objet d'un volume de travaux plus important que toutes les autres méthodes réunies, cela étant certainement dû à leur popularité mais aussi aux succès rencontrés en tant que méthode globale d'optimisation. Cependant, au sein des méthodes biomimétiques, ces algorithmes n'ont pas nécessairement tous les avantages de leur côté. Le problème du choix des paramètres reste difficile (cela ne concerne pas le choix du nombre de classes mais plutôt des paramètres liés à la méthode comme la taille de la population, les opérateurs, etc). La diversité des codages utilisés montre par ailleurs que les AG sont sensibles aux choix de la représentation et des opérateurs, et que le choix d'un opérateur de croisement pour optimiser des partitions n'est pas simple.

Il faut noter également que les principales différences entre les AG et les autres méthodes viennent du fait que dans les AG, un individu représente généralement une classification entière et la population est un ensemble de classifications, alors que dans les autres algorithmes, un individu représente une donnée et la population dans son ensemble représente la classification. Dans un AG, la solution au problème est le meilleur individu de la population, alors que dans les autres algorithmes, la solution est l'ensemble des individus. Cette différence est fondamentale puisque elle va obliger l'AG à centraliser son fonctionnement. Les autres algorithmes vont au contraire utiliser des principes heuristiques plutôt locaux et agissant en parallèle sur toute la classification. Sans doute que cela a des répercussions sur le temps d'apprentissage dans les AG, ce qui justifie l'étude d'approches génétiques hybrides.

Parmi les perspectives que l'on peut dégager, il faut noter qu'il existe encore peu de méthodes biomimétiques qui soient incrémentales, conceptuelles et/ou hiérarchiques. Des potentialités existent cependant : on peut ajouter des fourmis/données dans un algorithme tel [Lumer et Faieta, 1994], ou encore faire apparaître de nouveaux agents en déplacement dans un essaim. L'incrémentalité est possible dans de nombreux algorithmes, mais n'a pas encore été réellement testée, sans doute parce que les problèmes traités ne le requièrent pas. La classification conceptuelle semble un peu plus difficile car les méthodes représentent plutôt un partitionnement (au sens ensembliste du terme) plutôt que des regroupements selon des caractéristiques communes. Des pistes sont lancées notamment avec les systèmes immunitaires, mais en général les algorithmes n'utilisent pas, sauf dans le cas numérique avec des centroïdes, l'espace de description

des données pour créer des groupes. Les approches hiérarchiques sont également globalement ignorées, pourtant leur intérêt est grand pour l'interprétation des résultats par un expert humain. On aurait pu penser pour les AG que l'apparition de représentations arborescentes (programmation génétique, etc) allait donner lieu à des études génétiques et hiérarchiques, mais cela n'a apparemment pas encore été le cas. Un autre axe encore inexploré mais qui devrait pouvoir l'être par ces méthodes est le traitement de grandes bases de données. D'une part ces méthodes biomimétiques peuvent être assez facilement parallélisées ce qui n'est pas nécessairement le cas des méthodes classiques en classification. D'autres part elles utilisent (fourmi, essaim et systèmes immunitaires notamment) des propriétés statistiques des données plutôt que des cas particuliers : on peut imaginer par exemple augmenter par des facteurs importants la taille d'une population de fourmis ou encore celle d'un nuage d'agents, ou recourir par exemple à de l'échantillonnage.

Un des axes extrêmement prometteur pour certaines de ces méthodes (fourmis, essaim principalement) vient de leur capacité à fournir une classification comme les autres méthodes, mais également une visualisation de ces classifications. Cette capacité est une grande force des algorithmes à base d'essaim et de l'algorithme de [Lumer et Faieta, 1994] notamment (à l'image des cartes de Kohonen pour les réseaux de neurones). Cela permet à l'expert du domaine d'interpréter directement et interactivement les résultats et de formuler graphiquement des requêtes sur ces données. Notons que d'autres approches biomimétiques, comme les automates cellulaires par exemple, n'ont pas encore été réellement exploités pour le domaine de la classification.

Références

- [Abraham et Ramos, 2003] A. Abraham et V. Ramos. Web usage mining using artificial ant colony clustering and linear genetic programming. In *The Congress on Evolutionary Computation*, pages 1384–1391, Canberra, Australia, 08-12 December 2003. IEEE-Press.
- [Alexandrov, 2000] D. Alexandrov. Randomized algorithms for the minmax diameter k-clustering problem. In *Proceedings of ECCO 13*, pages 193–194, Capri, Italy, May 2000.
- [Azzag et al., 2003] N. Azzag, H. Monmarché, M. Slimane, G. Venturini, et C. Guinot. Anttree: a new model for clustering with artificial ants. In *IEEE Congress on Evolutionary Computation*, Canberra, Australia, 08-12 December 2003.
- [Babu et Murty, 1993] G.P. Babu et M.N. Murty. A near-optimal initial seed value selection in k-means algorithm using a genetic algorithm. 14:763–769, 1993.
- [Babu et Murty, 1994] G.P. Babu et M.N. Murty. Clustering with evolution strategies. 27(2):321–329, 1994.
- [Bezdek et al., 1994] J.C. Bezdek, S. Boggavarapu, L. Hall, et A. Bensaid. Genetic algorithm guided clustering. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 34–39, 1994.
- [Bhuyan et al., 1991] J.N. Bhuyan, V.V. Raghavan, et V.K. Elayavalli. Genetic algorithms for clustering with an ordered representation. In R.K. Belew et L.B. Booker,

- editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 408–15, San Diego, CA, 1991. Morgan Kaufmann.
- [Colorni *et al.*, 1991] A. Colorni, M. Dorigo, et V. Maniezzo. Distributed optimization by ant colonies. In *Proceedings of the First European Conference on Artificial Life*, pages 134–142, 1991.
- [de Castro et Von Zuben, 2000] L.N. de Castro et F.J. Von Zuben. An evolutionary immune network for data clustering. In *In Proceedings of the IEEE SBRN'00 (Brazilian Symposium on Artificial Neural Networks)*, pages 84–89, 2000.
- [Deneubourg *et al.*, 1990] J.-L. Deneubourg, S. Goss, N.R. Franks, A. Sendova-Franks, C. Detrain, et L. Chretien. The dynamics of collective sorting: robot-like ant and ant-like robots. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 356–365, 1990.
- [Falkenauer, 1994] E. Falkenauer. A new representation and operators for genetic algorithms applied to grouping problems. *Evolutionary Computation*, 2(2):123–144, 1994.
- [Fogel et Simpson, 1993] D.B. Fogel et P.K. Simpson. Evolving fuzzy clusters. In *ICNN93*, pages 1829–1834, San Francisco, 1993.
- [Goldberg, 1989] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
- [Greene, 2003] W.A. Greene. Unsupervised hierarchical clustering via a genetic algorithm. In IEEE Press, editor, *Proceedings of the 2003 Congress on Evolutionary Computation*, pages 998–1005, Canberra, Australia, 2003.
- [Hansohm, 2000] J. Hansohm. Two-mode clustering with genetic algorithms. In *Classification, Automation, and New Media: Proceedings of the 24th Annual Conference of the Gesellschaft Fur Klassifikation E.V.*, pages 87–94, 2000.
- [Jones et Beltramo, 1991] D.R. Jones et M.A. Beltramo. Solving partitioning problems with genetic algorithms. In R.K. Belew et L.B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 442–449, San Diego, CA, 1991. Morgan Kaufmann.
- [Knight et Timmis, 2002] T. Knight et J. Timmis. On data clustering with artificial ants. In J. Garibaldi A. Lotfi et R. John, editors, *Proceedings of the 4th International Conference on Recent Advances in Soft Computing*, pages 266–271, Nottingham, UK., December 2002.
- [Krishna et Murty, 1999] K. Krishna et M. Murty. Genetic k-means algorithm. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 29(3):433–439, 1999.
- [Kuntz *et al.*, 1997] P. Kuntz, P. Layzell, et D. Snyers. A colony of ant-like agents for partitioning in vlsi technology. In P. Husbands et I. Harvey, editors, *Proceedings of the Fourth European Conference on Artificial Life*, pages 417–424, 1997.
- [Labroche *et al.*, 2002] N. Labroche, N. Monmarché, et G. Venturini. A new clustering algorithm based on the chemical recognition system of ants. In F. van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence*, pages 345–349, Lyon, France, july 2002. IOS Press.

- [Lucasius *et al.*, 1993] C.B. Lucasius, A.D. Dane, et G. Kateman. On k-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasibility and comparison. *Analytica Chimica Acta*, 282:647–669, 1993.
- [Lumer et Faieta, 1994] E.D. Lumer et B. Faieta. Diversity and adaptation in populations of clustering ants. In *Proceedings of the Third International Conference on Simulation of Adaptive Behaviour*, pages 501–508, 1994.
- [Monmarché *et al.*, 1999] N. Monmarché, M. Slimane, et G. Venturini. On improving clustering in numerical databases with artificial ants. In D. Floreano, J.D. Nicoud, et F. Mondala, editors, *5th European Conference on Artificial Life (ECAL'99), Lecture Notes in Artificial Intelligence*, volume 1674, pages 626–635, Swiss Federal Institute of Technology, Lausanne, Switzerland, 13-17 September 1999. Springer-Verlag.
- [Monmarché *et al.*, 2002] N. Monmarché, C. Guinot, et G. Venturini. Fouille visuelle et classification de données par nuage d’insectes volants. *RSTI-RIA-ECA : Méthodes d’optimisation pour l’extraction de connaissances et l’apprentissage*, (6):729–752, 2002.
- [Nasaroui *et al.*, 2002] O. Nasaroui, D. Dasgupta, et F. Gonzalez. The fuzzy artificial immune system: Motivations, basic concepts, and application to clustering and web profiling. In *Proceedings of the IEEE International Conference on Fuzzy Systems at WCCI*, pages 711–716, May 12-17 2002.
- [Proctor et Winter, 1998] G. Proctor et C. Winter. Information flocking: Data visualisation in virtual worlds using emergent behaviours. In J.-C. Heudin, editor, *Proc. 1st Int. Conf. Virtual Worlds, VW*, volume 1434, pages 168–176. Springer-Verlag, 1998.
- [Raghavan et Birchard, 1979] V.V. Raghavan et K. Birchard. A clustering strategy based on a formalism of the reproductive process in natural systems. In *Information Implications into the Eighties, Proceedings of the Second International Conference on Information Storage and Retrieval*, pages 10–22. ACM, 1979.
- [Reynolds, 1987] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics (SIGGRAPH '87 Conference Proceedings)*, 21(4):25–34, 1987.

Summary

We present in this paper a survey of the methods and algorithms which have been used to solve the clustering problem. We describe the genetic and evolutionary approaches as well as the different encoding and operators which have been defined. Then we describe the artificial ant approach for clustering which is a rich source of inspiration. We detail other approaches using "agents" in a broad sense like for instance swarm intelligence (flocks) and like immune systems. Finally we summarize the presented approaches and we conclude on the perspectives related to the use of biomimetic methods for clustering.