Classification non supervisée à l'aide d'algorithmes génétiques

Jean-Marc Trémeaux

jm.tremeaux@gmail.com

Université Lyon 2

Master Recherche en Informatique, spécialité ECD

Travail personnel du cours « modèles classificatoires »

Année 2005-2006

Résumé La classification non supervisée est la recherche de partitions d'un ensemble de données. Des algorithmes génétiques ont été décrits dans la littérature afin d'effectuer ce partitionnement. Ceux-cis permettent une exploration efficace des espaces de recherche vastes et complexes. Dans ce rapport, je présente deux classes d'algorithmes de classification : non-hiérarchique (adapté des k-moyennes) et hiérarchique, et je commente les problématiques des algorithmes génétiques pour la classification : choix du codage, des opérateurs génétiques et de la fonction objectif.

1 Introduction

La classification non supervisée est une technique visant à établir une structure dans un ensemble de données. Celui-ci est partitionné en un ensemble de groupes, tels que les objets contenus à l'intérieur de chaque groupe soient les plus semblables possible au vu de leurs descripteurs, et que les groupes soient les plus dissemblables possible entre eux.

De nombreux algorithmes permettent d'effectuer ce partitionnement. Une méthode populaire est la classification ascendante hiérarchique, produisant une suite de partitions, ordonnées de la plus fine à la plus grossière, par aggrégation successive des objets ou groupes selon une mesure de dissimilarité et une stratégie gloutonne. Cette méthode est limitée aux petits ensembles de données car elle nécessite le stockage en mémoire d'une matrice de dissimilarité, dont la taille est quadratique en fonction du nombre de sommets.

Une autre méthode populaire est l'algorithme des k-moyennes, qui vise à constituer un nombre fixé de groupes optimisants une mesure d'inertie par réallocations successives et une stratégie de *hill-climbing*. Un inconvénient de cette méthode est de conduire à un résultat sous-optimal, dépendant du choix de la partition initiale.

Le nombre de partitions en k classes d'un ensemble de n éléments est [3] le suivant :

$$N(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^i C_k^i (k-i)^n$$

Il est donc impossible de faire une recherche exhaustive afin de trouver l'optimum global. D'autre part, les méthodes traditionnelles de recherche de partition n'exploitent

qu'une faible partie de l'espace des hypothèses. C'est pourquoi il est nécessaire de s'intéresser à des techniques de recherche permettant d'explorer plus efficacement cet espace.

Les algorithmes génétiques, inventés par John Holland dans les années 1960, sont une classe de méthodes de recherche qui s'inspirent des principes de l'évolution naturelle. Dans un algorithme génétique, une hypothèse est codée sous la forme d'une séquence correspondant à un chromosome. À partir d'une population initiale de ces chromosomes, on applique un cycle de sélections, croisements et mutations dans l'espoir d'évoluer vers une population plus adaptée au problème à résoudre, spécifié par la fonction d'évaluation.

Les algorithmes génétiques ont été appliqués avec succès dans des problèmes d'optimisation et d'apprentissage automatique [5]. Ceux-cis sont particulièrement efficaces pour parcourir des espaces de recherche à la fois vastes et complexes. Ils ne nécessitent notamment pas de connaissances spécifiques du domaine afin de converger vers une bonne solution, mais au contraire mettent en œuvre un cadre général permettant d'évaluer et de recombiner des solutions partielles. L'hypothèse sous-jacente est que la recombinaison de ces solutions partielles peut conduire à des solutions de meilleure qualité.

La mise en œuvre d'un algorithmes génétique nécessite d'étudier les quatres points suivants :

- Choix du codage des hypothèses ;
- Choix de la fonction d'évaluation ;
- Choix des opérateurs génétiques de sélection, croisement et de mutation ;
- Choix des valeurs des paramètres (taille de la population, probabilité de croisement et de mutation, critère d'arrêt).

Dans ce rapport, je présenterai le fonctionnement général des algorithmes génétiques, puis les représentations possibles des hypothèses pour la classification non supervisée. Ensuite, je montrerai comme les algorithmes génétiques peuvent être utilisés pour apprendre des partitions, puis des hiérarchies de partitions. Je concluerai sur quelques remarques.

2 Algorithmes génétiques

Les algorithmes génétiques sont une modélisation de l'évolution naturelle pour résoudre un problème de recherche. Les solutions potentielles sont codées sous forme de séquences, le plus souvent des chaînes de bits, appelées chromosomes. À partir d'une population initiale générée aléatoirement, de nouvelles populations sont générées itérativement par application d'opérateurs génétiques de sélection, croisement et mutation jusqu'à satisfaction d'un critère d'arrêt (*cf.* figure 1).

Tout d'abord, chaque candidat est évalué selon une fonction d'évaluation f. L'objectif de l'algorithme génétique est de résoudre un problème d'optimisation en identifiant les solutions candidates qui maximisent cette fonction f. Ensuite, une sous-population des candidats est sélectionnée pour la reproduction. Une méthode couramment employée est la sélection selon une roulette biaisée, où chaque candidat x a une chance d'être sélectionné proportionnelle à son évaluation $f(x)$. Une autre méthode est la sélection élitiste, où les n_0 candidats les plus aptes de chaque génération sont systématiquement sélectionnés. Une dernière méthode est la sélection par tournoi, où les solutions candidates sont confrontées par paires, la solution possédant le meilleur score l'emportant.

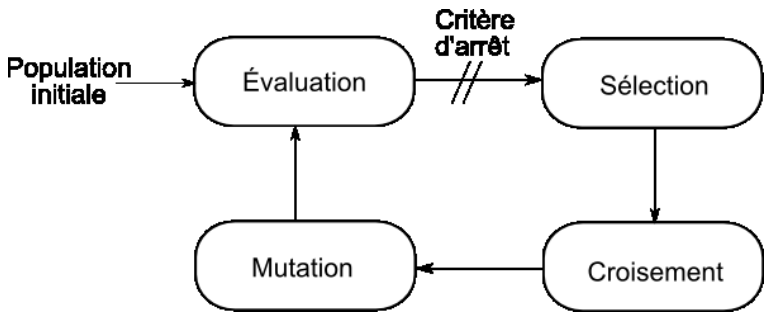

FIG. 1 – *Vue d'ensemble d'un algorithme génétique*

Pour chaque paire de chromosomes sélectionnés, celle-ci est croisée en un point aléatoire avec une probabilité p_c de croisement afin de former deux nouveaux chromosomes (*cf.* figure 2). Ce croisement modélise la recombinaison entre deux organismes haploïdes à un seul chromosome, et permet de combiner deux solutions sous-optimales afin d'obtenir des solutions potentiellement meilleurs. Si la recombinaison n'a pas lieu, alors une copie à l'identique des deux chromosomes est faite.

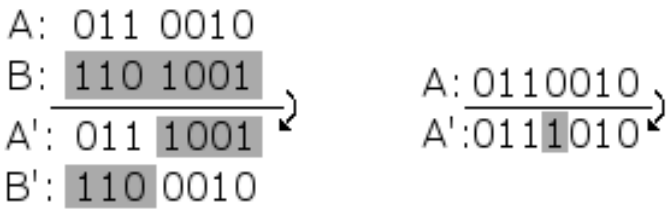

FIG. 2 – *Opérations de croisement et de mutation sur des représentations binaires*

Enfin, les chromosomes sont soumis aléatoirement à une mutation avec une probabilité de mutation p_m (*cf.* figure 2). Pour une représentation binaire, la mutation consiste à inverser la valeur d'un bit à une position aléatoire. D'autres opérateurs de mutation ont été définis pour les autres types de représentation.

Ce cycle est répété jusqu'à satisfaction d'un critère d'arrêt, c'est-à-dire soit après que n générations aient été créées, soit après la génération d'une solution suffisamment satisfaisante au sens de la fonction d'évaluation.

Une des forces des algorithmes génétiques est qu'ils sont intrinsèquement parallèles. La constitution d'une population de solutions candidates permet d'explorer indépendamment plusieurs régions de l'espace de recherche. Ce parallélisme permet d'explorer des espaces de recherche vastes sans avoir besoin d'exploiter une heuristique spécifique. De plus, si le codage des hypothèses et l'opérateur de croisement le permettent, il y a effectivement un échange d'information efficace entre les solutions candidates.

La principale difficulté des algorithmes génétiques est donc de trouver une repré-

sentation correcte des hypothèses. Diverses représentations (séquences binaires, entières, réelles, arborescences) ont été employées suivant les problèmes à résoudre. Une bonne représentation doit permettre de bien tirer parti des opérateurs génétiques de croisement et de mutation. Une autre difficulté est de définir une fonction d'évaluation pertinente pour le problème à résoudre. De plus, les valeurs des paramètres (taille de la population, probabilité de croisement p_c et de mutation p_m, type de sélection) doivent être choisis avec soin. Une taille trop faible de population ne permet pas d'explorer sufisamment l'espace de recherche, tandit que des taux de mutation ou de croisement trop élevés peuvent mener à une exploration chaotique.

3 Représentation des hypothèses

Différents auteurs ont appliqués des algorithmes génétiques au problème de la classification non supervisée en utilisant des représentations différentes. Les premiers travaux dans ce domaine sont dus à Raghavan et Birchard en 1979 [6]. Ceux-cis ont travaillé sur le partitionnement d'un ensemble de données de taille n en un nombre fixé k de classes. Ils utilisent comme représentation d'une hypothèse un vecteur de taille n associant à chaque objet sa classe d'appartenance (*cf.* tableau 1).

Objet	1	2	...	n
Classe ($\in [1,k]$)	4	1	...	5

TAB. 1 – Codage par classe d'appartenance

Il faut noter que chaque combinaison possible de valeurs d'un chromosome ne correspond pas nécessairement à une solution valide. En effet, des classes peuvent être vides, ce qui ne constitue plus une partition en k classes. Ceci est un problème général des algorithmes génétiques, et deux approches peuvent être employées pour conserver des chromosomes valides :

- utiliser des opérateurs de croisement et de mutation qui produisent toujours un résultat valide ;
- réparer le chromosome non-valide après opération.

Un codage similaire est une matrice booléenne d'appartenance (*cf.* tableau 2). Chaque élément m_{ij} ($i \in [1,n], j \in [1,k]$) de la matrice prend la valeur 1 si l'objet i appartient à la classe j, et la valeur 0 sinon.

Objet/Classe	c_1	c_2	...	c_k
x_1	0	1	...	0
x_2	1	0	...	0
...	...	...	...	...
x_n	0	0	...	1

TAB. 2 – Codage par matrice d'appartenance

Un autre codage utilisé est à base de permutations avec séparateurs (*cf.* tableau 3),

utilisant les entiers 1 à n pour identifier les objets, et les entiers $n+1$ à $n+k-1$ pour indiquer les bornes des classes.

1	4	2	**7**	3	6	5

TAB. 3 – Codage en 2 classes par permutation avec séparateur

Un autre codage employé spécifiquement pour les techniques inspirées des k-moyennes est de représenter chaque classe par un objet prototype, les objets les plus proches de chaque prototype étant implicitement affectés à leur classe (*cf.* tableau 4).

Centroïde	1	2	...	k
Coordonnées	$(15, 20)$	$(123, 82)$	...	$(250, 132)$

TAB. 4 – Codage par objets prototypes

Enfin, un dernier codage employé est une hiérarchie de partitions, la racine représentant la population totale et chaque sous-sommet une sous-population de son père. L'opérateur de croisement consiste alors à permuter des branches de deux arbres, tout en maintenant la cohérence de celui-ci. Un objet ne doit en effet pas de trouver dans deux sommets voisins, et l'arbre doit comporter tous les objets de l'ensemble de données.

4 K-moyennes génétiques

Je décris dans ce chapitre deux méthodes, basées sur les k-moyennes, employées pour effectuer un partitionnement d'un ensemble de données en un nombre k fixé de classes.

Maulik et al. [4] utilisent une représentation sous forme d'un vecteur de k tuples, codant les coordonnées de k centroïdes par des nombres réels. Cette représentation a l'avantage d'être toujours valide quelle que soit l'opération effectuée. La population initiale consiste en P chromosomes initialisés aléatoirement. L'évaluation consiste tout d'abord à constituer les groupes en assignant chaque objet au centroïde le plus proche, puis à recalculer les coordonnées des centroïdes comme les points moyens de leur groupe, et enfin à calculer une mesure d'inertie totale intra-groupes. Cette mesure est inversée afin de procéder à un problème de maximisation de fonction. La sélection consiste en une roulette proportionnelle élitiste (le meilleur candidat de la génération précédente est systématiquement conservé). Des opérateurs de croisement et de mutation classiques sont employés.

Cette technique peut être qualifiée d'algorithme génétique hybride, car une opération importante de constitution de groupes (similaire à l'algorithme des k-moyennes) est réalisée lors de la phase d'évaluation. Il faut noter que cet algorithme converge asymptotiquement vers l'optimum global, grâce à une sélection élitiste et à une probabilité strictement positive de passer d'une population quelconque à une population contenant une solution optimale.

Krishna et al. [2] utilisent une représentation sous forme de matrice d'appartenance. Celle-ci peut coder des solution invalides (comportant des groupes vides), et un soin est donc apporté pour conserver des solutions valides. À l'initialisation, chaque objet est

affecté à un groupe $1, ..., k$ aléatoirement. La fonction d'évaluation est une mesure d'inertie totale intra-groupes, transformée afin de se ramener à une problème de maximisation. La sélection se fait par une roulette proportionnelle. L'opérateur de mutation, nommé *« mutation basée sur la distance »* consiste à réallouer un objet aléatoire à un groupe voisin. Les groupes les plus proches ont plus de chance d'être selectionnés. L'algorithme ne comporte pas d'opération de croisement, mais remplace celle-ci par un *« opérateur k-moyennes »*, semblable à celui de l'algorithme précédent : calcul des centroïdes, puis réallocation de chaque objet au centroïde le plus proche.

Une preuve de la convergence globale asymptotique de cet algorithme est donnée.

5 Classification hiérarchique

Greene [1] propose une méthode originale permettant de constituer des hiérarchies de partitions. Il décrit tout d'abord une nouvelle méthode descendante, subdivisant récursivement une population de départ en sous-populations, en essayant d'optimiser une mesure tenant compte de l'inertie intra-groupes et inter-groupes et de la taille des groupes constitués. Cette méthode est dépendante des conditions initiales et en particulier de l'ordre dans lequel sont insérés les objets dans l'arbre. Greene propose alors de génerer le meilleur arbre possible par application d'un algorithme génétique. Pour cela, une population initiale d'arbres est générée en choisissant un ordre aléatoire d'insertion des objets. Une mesure de la qualité de chaque arbre est effectuée, puis ceux-ci sont sélectionnés par roulette proportionnelle élitiste (les deux meilleurs solutions sont conservées). Le croisement de deux arbres consiste à choisir les meilleures branches du premier niveau de chaque arbre et à les réunir. Ce croisement peut produire des solutions non-valides de deux façons. Un objet peut se retrouver dans deux classes simultanément, il est alors supprimé de la moins bonne. Un objet peut être absent de la partition, il est alors réinséré par l'algorithme original. L'opération de mutation consiste à supprimer un objet aléatoirement de l'arbre et à le réinsérer dans l'arbre par l'algorithme original, ce qui peut changer la constitution des groupes.

Cette méthode, qui a pour originalité d'être hiérarchique et de génerer un nombre de classes k libre, ne fournit pas d'indication sur l'optimalité de la solution générée.

6 Conclusion

Dans ce travail, j'ai présenté succintement différentes méthodes et points importants pour réaliser une classification non supervisée à l'aide d'algorithmes génétiques. Les techniques employées par tous les auteurs sont des algorithmes hybrides (algorithmes génétiques biaisés pour être efficaces pour la classification), et sont très différents et donc difficiles à comparer entre eux. Le tableau 5 tente cependant de résumer les points principaux des trois méthodes décrites. Les performances de ces algorithmes semblent intéressantes, celles-cis étant comparables aux algorithmes traditionnels, tout en assurant pour deux d'entre eux une convergence globale asymptotique.

	Maulik et al. [4]	Krishna et al. [2]	Greene [1]
Modèle	partition en k fixé	partition en k fixé	hiérarchie de partitions (k libre)
Représentation	centroïdes (coordonnées réelles)	indices des classes d'appartenance	arbre
Validité	toujours valide	non (groupes vides)	non (groupes vides, doublons)
Initialisation	position aléatoire des centroïdes	groupe d'appartenance aléatoire	ordre d'insertion aléatoire
Évaluation	inertie intra-classes totale (inverse)	inertie intra-classe totale (σ-tronquée)	mesure d'inertie intra. et inter. et finesse des groupes
Sélection	roulette proportionnelle élitiste	roulette proportionnelle	roulette proportionnelle élitiste
Croisement	croisement des chaînes en un point	non (k-moyennes)	croisement des meilleures branches du premier niveau
Mutation	mutation d'une coordonnée d'un centroïde	réallocation biaisée groupe plus proche plus probable	suppression puis réinsertion d'un objet dans l'arbre
Remarques	convergence globale k-moyennes caché dans l'évaluation	convergence globale correction après k-moyennes	algorithme hiérarchique non-standard dépendant des conditions initiales

TAB. 5 – Comparatif des méthodes de classification

Références

[1] William A. Greene. Unsupervised hierarchical clustering via a genetic algorithm. In *Proceedings of the 2003 Congress on Evolutionary Computation*, pages 998–1005, 2003.

[2] K. Krishna and M. Narasimha Murty. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 29(3) :433–439, 1999.

[3] G. L. Liu. *Introduction to combinatorial mathematics.* McGraw Hill, 1968.

[4] U. Maulik and S. Bandyopadhyay. Genetic algorithm-based clustering technique. *Pattern Recognition*, 33 :1455–1465, 2000.

[5] Melanie Mitchell. *An introduction to genetic algorithms.* The MIT Press, 1996.

[6] V.V. Raghavan and K. Birchard. A clustering strategy based on a formalism of the reproductive process in natural systems. *Information Implications into the Eighties, Proceedings of the Second International Conference on Information Storage and Retrieval*, pages 10–12, 1979.