

Vers un algorithme multi-agents de clustering dynamique

Gaële Simon*, Dominique Fournier**, Bruno Mermet*

*GREYC CNRS UMR 6072, 6 Boulevard du Maréchal Juin 14050 CAEN cedex
{gale.simon, bruno.mermet}@univ-lehavre.fr,

**LITIS EA 4051, 25 rue Philippe Lebon, BP 540 76058 Le Havre cedex
dominique.fournier@univ-lehavre.fr

Résumé. Dans cet article, nous présentons un algorithme multi-agents de clustering dynamique. Ce type de clustering doit permettre de gérer des données évolutives et donc être capable d'adapter en permanence les clusters construits.

1 Introduction

Dans cet article, nous proposons une technique de clustering dynamique de données évolutives. Cette problématique est née de l'objectif initial de nos travaux visant à permettre, au cours de l'exécution d'un système multi-agents, de détecter des groupes d'agents liés à des phénomènes d'auto-organisation. On se trouve donc face à un problème de clustering dynamique qui présente les deux particularités suivantes : le cardinal de l'ensemble de données à clusteriser n'est pas constant et des données déjà clusterisées peuvent être modifiées du fait de l'évolution des agents correspondants.

Cela peut entraîner des modifications ou des réorganisations de l'ensemble existant de clusters. Ainsi, une méthode de clustering dynamique est nécessaire afin d'adapter continuellement l'ensemble des clusters afin qu'ils reflètent le mieux possible l'état courant des données.

2 Travaux connexes

Il existe de nombreux travaux portant sur les techniques de clustering où l'ensemble des données à clusteriser n'est pas totalement connu dès le départ comme en clustering classique. On trouve en particulier dans cette catégorie les techniques de clustering de flux de données et de flux de données évolutifs. Malheureusement, ces algorithmes ne prennent pas en compte le fait que des données déjà clusterisées puissent elles aussi évoluer.

Les travaux les plus proches de notre problématique concernent un algorithme de clustering de données mobiles présenté dans [Li et al. (2004)] : un micro-clustering est effectué en enrichissant les données d'un vecteur vitesse. Cependant, dans un deuxième temps, l'algorithme k-means doit être utilisé pour regrouper les micro-clusters, ce qui oblige à donner un nombre de clusters attendu et à ce que ce nombre soit constant.

3 Notre approche

Les algorithmes fournis de clustering semblent plus adaptés à la prise en compte de l'évolution des données. Ainsi, l'algorithme AntClass [Monmarché (2000)] associe successivement en quatre phases un algorithme de fourragement et l'algorithme k-means. Cette approche n'étant pas compatible avec l'aspect dynamique de notre problématique, nous avons décidé

Vers un algorithme multi-agents de clustering dynamique

de ne garder que l'algorithme fourmi utilisé dans AntClass et d'associer une couche d'agents cluster aux fourmis, chaque tas créé par les fourmis étant encapsulé dans un agent cluster.

Un agent cluster évolue au fur et à mesure que les fourmis lui donnent ou lui prennent des données et que les données qu'il contient évoluent. Mais il peut également se développer en attaquant d'autres agents cluster avec lesquels il est en intersection. Les clusters sont alors mis à jour par fusion complète ou partielle lors de la fuite de l'agent agressé. Ceci permet de compenser la tendance des fourmis à construire trop de tas et de prendre en compte l'évolution des données.

4 Prototypes et expérimentations

Nous avons développé une plateforme de simulation qui, connectée à la plateforme de clustering dynamique, sert de support à nos expérimentations. Les deux plateformes ont été développées en JAVA en utilisant l'environnement de développement de SMA MADKIT. La plateforme de simulation permet de simuler l'évolution de populations d'agents. Une population est considérée comme étant un ensemble d'agents qui vont, *a priori*, évoluer de manière similaire. Une population doit donc être normalement perçue, au niveau du clustering, comme un cluster qui va évoluer au cours du temps.

L'analyse de nos résultats repose pour le moment sur trois types de courbes temporelles : nombre de données agrégées, nombre de clusters et pureté moyenne des clusters. De cette analyse, il ressort que les clusters détectés sont en général purs, mais un peu trop nombreux, et que quelques données restent non clusterisées. L'algorithme fonctionne également bien sur des données bruitées.

5 Conclusion et perspectives

Des résultats prometteurs nous incitent à poursuivre dans cette voie en améliorant un certain nombre de points, notamment pour mieux prendre en compte la dissociation de clusters. Nous comptons notamment prendre en compte la vitesse des données comme cela est fait dans [Li et al. (2004)]. Enfin, d'un point de vue implantation, nous travaillons actuellement sur une nouvelle version des plateformes permettant un meilleur contrôle sur l'activation des différents agents. À plus long terme, notre objectif est d'appliquer cette méthode à différents problèmes concernant notamment l'émergence dans les systèmes multi-agents.

Références

- Li, Y., J. Han, et J. Yang (2004). Clustering moving objects. In *KDD (Knowledge Discovery in Databases)*, pp. 617–622.
- Monmarché, N. (2000). *Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation*. Ph. D. thesis, Université de Tours, France.

Summary

In this paper, a multi-agents algorithm for dynamic clustering is presented.