Identification de blocs homogènes sur des données continues

François-Xavier Jollois, Mohamed Nadif

LITA - IUT de METZ, Université de Metz,
Ile du Saulcy, 57045 METZ Cedex, France
{jollois,nadif}@iut.univ-metz.fr,

Résumé. Contrairement aux méthodes usuelles de classification ne cherchant généralement qu'une seule partition, soit des instances, soit des attributs, les méthodes de classification croisée et de classification directe fournissent des blocs de données liant des instances à des attributs. Les premières consistent à chercher simultanément une partition en lignes et une partition en colonnes. Les secondes, elles, s'appliquent directement sur les données, et permettent d'obtenir des blocs de données homogènes de toutes tailles, ainsi que des hiérarchies de classes en lignes et en colonnes. Combinant les avantages des deux méthodes, nous présentons ici une méthodologie permettant de travailler sur de grandes bases de données.

1 Introduction

Lorsque le but d'une classification est d'obtenir une structure en blocs homogènes, nombre d'utilisateurs appliquent généralement des algorithmes de classification simple sur les instances et sur les attributs séparément, les blocs résultent du croisement des partitions obtenues. Une telle méthode ne permet pas d'expliquer la relation spécifique pouvant exister entre un groupe d'instances et un groupe d'attributs. Ainsi, il est préférable d'appliquer des algorithmes de classification croisée, tel que l'algorithme *Croeuc* [Govaert, 1983, Govaert, 1995]. Celui-ci cherche simultanément une partition en lignes et une partition en colonnes, dont les centres permettent de synthétiser les données sous forme d'une matrice d'information de taille réduite. Une deuxième façon d'aborder le problème de la classification simultanée est d'utiliser un algorithme de classification directe, comme *Two-way splitting* [Hartigan, 1975], qui cherche à obtenir des blocs de données homogènes et de toutes tailles. On peut aussi citer les travaux de Marcotorchino [Marcotorchino, 1987] sur la sériation.

Malgré sa rapidité et son efficacité de traiter des tables de grande taille, l'algorithme *Croeuc* présente un défaut majeur ; il nécessite la connaissance des nombres de classes en lignes et en colonnes. Par contre, l'algorithme *Two-way splitting* s'affranchit de cette hypothèse mais sa complexité rend son utilisation impossible sur des données de grande taille. Nous présentons donc ici une combinaison de ces deux algorithmes afin de pallier les inconvénients de chacun.

Dans un premier temps, nous décrivons l'algorithme de classification croisée *Croeuc*. Ensuite, nous présentons l'algorithme *Two-way splitting*. Puis, nous décrivons la combinaison de ces deux algorithmes, et nous illustrons cette démarche par une application sur des données simulées. Enfin, nous concluons sur l'intérêt de cette méthode, ainsi