Processus itératif d'extraction de classes en non supervisée

Alexandre Blansché et Lydia Boudjeloud-Assala

Université de Lorraine,
Laboratoire d'Informatique Théorique et Appliquée, LITA-EA 3097,
Metz, F-57045, France
alexandre.blansche|lydia.boudjeloud-assala@univ-Lorraine.fr

Résumé. Nous proposons dans cet article une nouvelle approche de classification non supervisée où les classes sont obtenues les unes après les autres suivant un processus itératif. L'approche utilise une méthode d'extraction de classes basée sur la détection de limite de classe, chaque classe étant définie par son centre. Nous avons également défini des critères d'évaluation adaptés à la méthode proposée. Plusieurs expérimentations ont montré l'intérêt de l'approche dans divers problèmes.

1 Introduction

La classification non supervisée est un problème étudié depuis plusieurs décennies, et récemment de nouvelles approches ont été développées pour s'adapter au challenge induit par les nouvelles méthodes d'acquisition automatique des données et le nombre croissant d'application produisant des données massives. Ces données doivent être étudiées par des algorithmes suffisamment efficaces afin de pouvoir exploiter les connaissances qu'elles contiennent. En procédant à une classification non supervisée, on cherche à construire des ensembles homogènes d'individus, c'est-à-dire partageant un certain nombre de caractéristiques identiques. Classiquement, les méthodes fonctionnent de la façon suivante : l'utilisateur fixe le nombre de classes, un partitionnement est ensuite généré puis évalué par l'utilisateur même ou par des critères d'homogénéité, le partitionnement obtenu pouvant être remis en question selon son évaluation. Nous proposons une approche différente qui consiste à présenter et évaluer une par une les classes sans en fixer préalablement le nombre. L'approche générique est basée sur un processus itératif qui va extraire les classes les unes après les autres permettant ainsi l'exploration pas à pas des données. L'approche propose à l'utilisateur en priorité les classes les plus pertinentes (selon un critère donné) et lui laisse le soin de décider quand arrêter le processus. L'approche peut être intégrée dans un système interactif qui lui permettra d'étudier les classes individuellement ou les unes par rapport aux autres. Cet article est organisé de la façon suivante. Nous allons d'abord présenter notre approche itérative ainsi que les critères utilisés pour l'extraction de classes puis présenter certains de nos résultats expérimentaux avant de conclure.

2 Méthode proposée

Nous proposons une approche itérative générique qui va extraire les classes les unes après les autres pour permettre une exploration pas à pas des données. Le processus itératif est répété à la demande de l'utilisateur. À chaque itération, une nouvelle classe est extraite : une méthode d'optimisation (un extracteur) recherche la meilleure classe à extraire selon un critère d'évaluation donné afin d'obtenir un sous-ensemble de données homogène et séparé des autres objets. La classe est alors proposée à l'utilisateur, qui va pouvoir l'analyser à l'aide d'outils de visualisation interactifs. L'utilisateur pourra ensuite demander au système d'extraire une nouvelle classe, pour poursuivre son exploration des données. L'extraction d'une nouvelle classe prendra en compte les classes extraites précédemment.

2.1 Extraction d'une classe

Il existe plusieurs méthode pour extraire une classe homogène à partir d'un ensemble de données. Dans cet article, nous introduisons une méthode d'extraction de classes basée sur la détection de limite de classe. Une classe est extraite à partir d'un « centre » : nous calculons la distance entre chaque objet et le centre de la classe et cherchons alors la première augmentation abrupte dans ces valeurs qui indiquera la limite de la classe extraite. Un algorithme d'optimisation pourra être utilisé pour chercher un centre de classe, qui produit une classe homogène selon un critère donné. Nous avons testé deux méthodes pour détecter la limite de la classe, la méthode CUSUM Basseville et Nikiforov (1993) et une méthode de détection de pics présentée dans Palshikar (2009), appliquée sur le différentiel des distances. Une fois la limite de classe évaluée, tous les objets qui ont une distance inférieure à cette limite appartiennent à la classe extraite.

2.2 Évaluation d'une classe

Pour proposer à l'utilisateur les classes les plus pertinentes, il faut définir des critères d'évaluation. La plupart des critères d'évaluation en classification non supervisée évaluent l'ensemble de données dans son intégralité et ne donnent pas une évaluation des classes indépendamment les unes des autres. Nous proposons donc deux nouveaux critères d'évaluation pour évaluer les classes indépendamment les unes des autres, dérivés de critères classiques en classification non supervisée. Dans l'ensemble des critères définis, C_k représente la k-ième classe extraite. Le premier critère est le rapport d'inertie IR (rapport entre l'inertie intra-classe et l'inertie totale des données, normalisé par le nombre d'objets dans la classe et dans l'ensemble de données :

$$IR\left(C_k\right) = \frac{Card(D) \sum_{o \in C_k} d\left(o, c_k\right)^2}{Card(C_k) \sum_{o \in D} d\left(o, g\right)^2}$$

.

Le second critère proposé est le rapport de limite de la classe CLR, représentant le rapport entre la distance du dernier objet de la classe sur la distance du premier objet hors de la classe :

$$CLR\left(C_k\right) = \frac{\max\limits_{o \in C_k}\left(d\left(o, c_k\right)\right)}{\min\limits_{o \notin C_k}\left(d\left(o, c_k\right)\right)}$$

.

Nous pouvons alors utiliser l'un ou l'autre des critères qui produisent une évaluation de la compacité d'une classe sphérique. Cependant, comme chaque classe est extraite individuellement, nous devons également nous assurer que les classes extraites sont différentes les unes des autres. Nous proposons donc d'ajouter une pénalisation des classes selon leur chevauchement avec les classes précédemment découvertes. Le critère de pénalité OP calcule une pénalité selon l'intersection et l'union de la classe extraite avec les classes précédentes ($\lambda \geq 0$ représente le poids de la pénalité selon l'importance que l'on donne aux chevauchements).

$$OP\left(C_k\right) = \lambda \max_{i=1\ldots k_1} \frac{Card(C_k \cap C_i)}{Card(C_k \cup C_i)}$$

3 Expérimentation

3.1 Méthode de *détection de limite*

Une première expérimentation a servie à étudier l'impact de la méthode de *détection de limite* de la classe. Nous avons, pour cela, utilisé des données artificielles en deux dimensions constituées de trois classes (figure 1), deux étant relativement proches, la dernière plus éloignée. Pour cet ensemble de données, nous illustrons la méthode de *détection de limite* de la classe centrée à l'origine et avons donc calculé les distances de chaque points à l'origine. Sur la figure 1, nous pouvons voir en haut les nuages de points des classes et en bas le graphique des distances calculées. La méthode de *détection de limite* de classes consiste, alors, à appliquer la méthode CUSUM (figure 2) ainsi que la méthode de détection de pics (figure 3) sur les distances. On remarque sur la figure 2 que la méthode CUSUM permet de détecter seulement le plus grand changement (minimum de la fonction CUSUM) qui ne correspond pas forcément à la limite de la classe considérée. Cependant, comme nous pouvons le voir sur la figure 3, l'approche basée sur les pics du différentiel de distances, permet de détecter plusieurs changements brusques sur les distances, il nous suffit, alors, de choisir le premier pic détecté pour déterminer la limite de la classe considérée.

3.2 Processus itératif

Une seconde expérimentation a servi à évaluer l'approche itérative d'extraction de classes avec les différents critères proposés. Nous avons donc appliqué notre approche sur un ensemble de données artificiel simple à deux dimensions pour en expliquer aisément le fonctionnement. Les données ont été créées avec quatre distributions gaussiennes, trois d'entre elles étant proches les unes des autres, la quatrième plus éloignée (cf. figure 4). Nous avons appliqué notre approche en utilisant la méthode de détection de pics pour déterminer la limite de la classe, le critère d'évaluation $CLR + OP$. Sur la figure 5, nous pouvons voir la première classe extraite, il s'agit de la classe isolée. Nous voyons, sur la figure 6, la deuxième classe extraite, qui représente un regroupement de trois petites classes proches les unes des autres. En

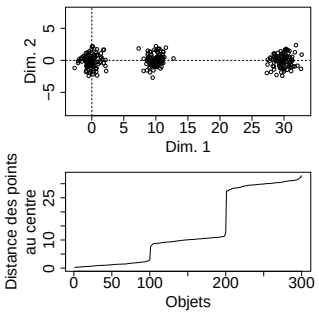

FIG. 1 – *Ensemble de données artificiel et distance des points à l'origine.*

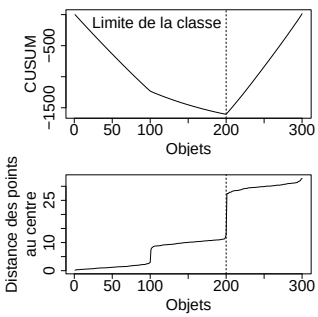

FIG. 2 – *Détermination de la limite de la classe avec CUSUM.*

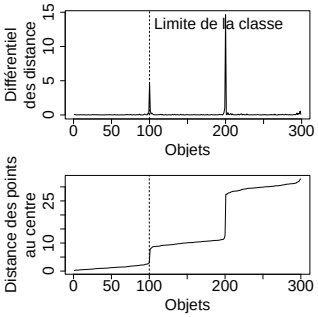

FIG. 3 – *Détermination de la limite de la classe avec la méthode des pics.*

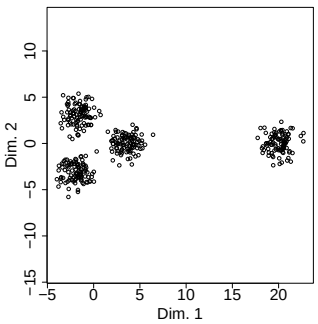

FIG. 4 – *Ensemble de données artificiel (4 classes).*

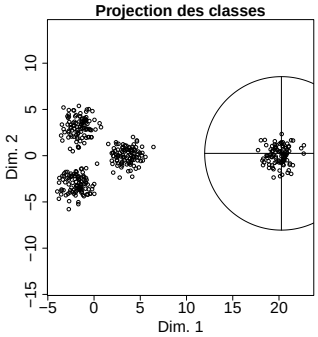

FIG. 5 – *1re classe extraite selon la méthode des Pics avec les critères CLR et OP.*

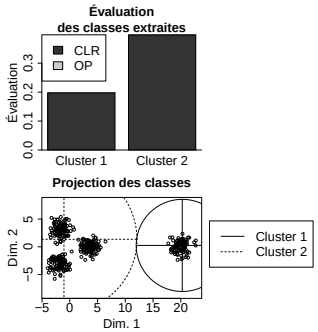

FIG. 6 – *2e classe extraite avec évaluation des critères CLR et OP.*

continuant le processus, les trois classes (figures 7, 8 et 9) constituant la classe 2 sont extraites l'une après l'autre. On remarque alors (haut de la figure) que la pénalité de chevauchement s'applique sur ces classes puisqu'elles sont contenues dans la classe 2. Nous avons ensuite appliqué notre algorithme avec le critère d'évaluation IR, sans modifier les autres paramètres. Sur la figure 10, on observe les cinq classes obtenues. Ici les trois classes proches les une des autres ont été découvertes individuellement avant la classe qui les englobe.

3.3 Application au biclustering

L'approche proposée a été utilisée dans l'article Boudjeloud-Assala et Blansché (2012) pour une application en biclustering. Dans ce cadre, la recherche de classes s'est accompagnée d'une recherche de sous-ensemble d'attributs. L'extraction et l'évaluation d'une classe se fait dans un sous-ensemble d'attributs décrivant les données. L'optimisation est faite par un algorithme évolutionnaire qui recherche en même temps un sous-ensemble d'attributs et un centre de classe optimal, définissant ainsi, l'extracteur de classe. Dans Boudjeloud-Assala et

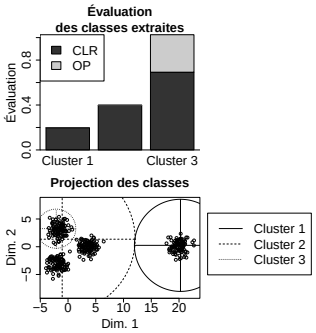

FIG. 7 – *3e classe extraite avec évaluation des critères CLR et OP.*

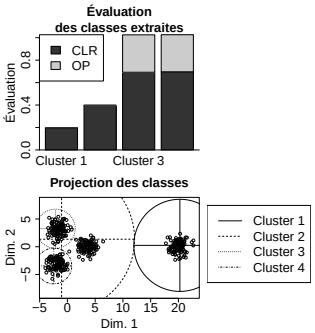

FIG. 8 – *4e classe extraite avec évaluation des critères CLR et OP.*

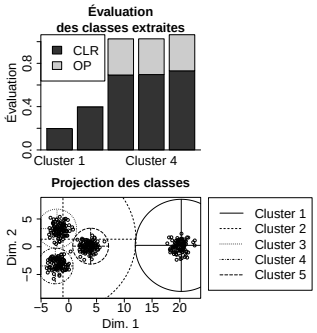

FIG. 9 – *5e classe extraite avec évaluation des critères CLR et OP.*

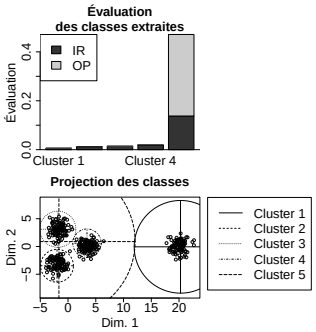

FIG. 10 – *Classes extraites avec évaluation des critères IR et OP.*

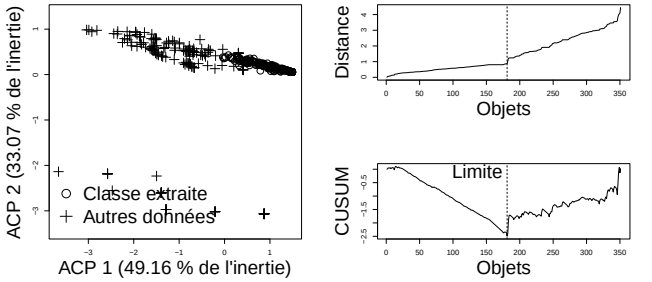

FIG. 11 – *Première classe extraite de l'ensemble de données Ionosphere.*

Blansché (2012) la méthode CUSUM a été utilisée pour détecter la limite de la classe. Une fois une classe extraite dans un sous-ensemble de données, des méthodes de visualisation peuvent être utilisées pour l'étudier. Nous présentons dans la figure 11, la première classe extraite sur l'ensemble de données Ionosphere de l'UCI Blake et Merz (1998) projetée sur des axes d'ACP. Les tests ont été réalisés avec une implantation en Java sur un processeur à 1,3GHz. Les résultats sont présentés sur la table 1, en indiquant le nombre d'objets et de dimensions. Notre approche s'est montrée relativement rapide sur des ensembles de données de taille modérée et ne prend que quelques minutes sur des ensembles de grande taille. Ces premiers résultats sont encourageants, l'approche pourra être utilisée pour explorer de grandes bases de données.

4 Conclusion

Nous avons proposé dans cet article une nouvelle approche de classification non supervisée où les classes sont obtenues les unes après les autres suivant un processus itératif sans préciser

Données	Nombre d'objets	Nombre de dimensions	Temps (s)
Ionosphere	351	34	8
Movement Libras	360	90	18
Bicatyeast	419	70	20
Isolet	6238	616	474
Ovarian	253	15154	77

TAB. 1 – *Temps d'extraction d'une classe.*

auparavant le nombre de classes. Nous avons introduit une méthode d'extraction de classes basée sur la détection de limite de classe, chaque classe étant définie par son centre. Nous avons également défini des critères d'évaluation adaptés à la méthode proposée associés à un critère de pénalité permettant plus ou moins le chevauchement. Plusieurs expérimentations ont montré l'intérêt de l'approche dans divers problèmes.

Concernant les travaux futurs, nous avons prévu de mettre en place une plate-forme interactive d'exploration pas à pas des données. L'approche propose à l'utilisateur en priorité les classes les plus pertinentes (selon un critère donné) et lui laisse le soin de décider, permettant ainsi à l'utilisateur d'étudier les classes individuellement ou les unes par rapport aux autres (treillis, visualisation,...).

Références

Basseville, M. et I. Nikiforov (1993). *Detection of Abrupt Changes : Theory and Application*. Prentice-Hall, Englewood Cliffs, N.J.

Blake, C. et C. Merz (1998). Uci repository of machine learning databases. http ://www.ics.uci.edu/∼mlearn/MLRepository.html.

Boudjeloud-Assala, L. et A. Blansché (2012). Iterative evolutionary subspace clustering. *Proceedings of 2012 International Conference on Neural Information Processing, LNCS 7663*, 424–431.

Palshikar, G. (2009). Simple algorithms for peak detection in time-series. *Proceedings of 1st International Conference on Advanced Data Analysis Business Analytics and Intelligence (ICADABAI)*.

Summary

We propose in this paper a new clustering approach in which different clusters are obtained iteratively. The approach uses a cluster extraction method based on a cluster limit detection method for clusters defined by their center. We also defined evaluation criteria suitable for the proposed method. Experimentations showed that the approach is interesting for different applications such as clustering with overlapping and coclustering.