Clustering prudent : une approche relationnelle par seuillage

Sébastien Destercke, Marie-Hélène Masson, Benjamin Quost

Clustering prudent : une approche relationnelle par seuillage

S. Destercke[1] M. Masson[2] B. Quost[1]
[1] Université de Technologie de Compiègne
[2] Université de Picardie Jules Verne
UMR CNRS 7253 Heudiasyc, Sorbonne Universités, Université de Technologie de Compiègne
CS 60319 - 60203 Compiègne cedex, France

Résumé :
Le problème du clustering peut être vu comme la recherche de classes d'équivalence d'objets. Dans la lignée de travaux récents sur la discrimination prudente, nous proposons dans cet article une méthode fournissant un clustering partiel sous la forme d'une matrice relationnelle incomplète. Cette approche permet de détecter des objets ambigus (par exemple sur les bords des classes), des classes de petite taille qui pourraient être fusionnées, ou encore des sous-ensembles d'objets dont la relation est mal définie.

Mots-clés :
Clustering relationnel, clustering prudent.

Abstract:
Clustering has mainly been treated as a relational problem, where equivalence classes are to be computed. In the spirit of recent research focused on partial predictions, we propose a method to return a partial clustering of the data, in the form of an incomplete relational (or block) matrix. This matrix can eventually be completed into a full partition (block matrix). This formalization can be used to detect ambiguous objects (i.e., on the borders of clusters), small clusters that may potentially be merged, subsets of objects whose relation is ambiguous, etc.

Keywords:
Relational clustering, partial clustering.

1 Introduction

La classification automatique, ou clustering, cherche à répartir un ensemble d'observations $\mathbf{x_1}, \ldots, \mathbf{x_n}$ en K sous-ensembles $\omega_1, \ldots, \omega_K$. Le clustering est formellement équivalent à la recherche de classes d'équivalence, c'est à dire, le calcul d'une *matrice relationnelle* R de terme général $R_{i,j}$ $(i, j = 1, \ldots, n)$, indiquant la présence ($R_{i,j} = 1$) ou l'absence ($R_{i,j} = 0$) de relation entre chaque paire d'objets $\mathbf{x_i}$ et $\mathbf{x_j}$. La matrice relationnelle représente une relation d'équivalence si et seulement si elle satisfait les propriétés suivantes : réflexivité, symétrie, et transitivité. Si ces conditions sont vérifiées, une partition des objets peut être retrouvée en réarrangeant les lignes et les colonnes de R de manière à obtenir une matrice diagonale par blocs. Une telle partition est appelée partition dure car chaque objet est affecté sans ambiguïté à une classe unique. Cette contrainte est relaxée dans les approches probabilistes [2], floues [6, 1] ou encore évidentielles [12]. Dans ces approches, on considère que les observations peuvent appartenir à plusieurs classes avec des degrés divers. Dans cet article, nous proposons une manière différente de relaxer la contrainte de partition dure. Nous proposons de fournir un clustering dur, mais *partiel*, dans lequel certaines relations entre paires d'objets sont inconnues et peuvent être remplacées de différentes manières tout en conservant la cohérence avec les autres relations. Cette idée a des liens avec la "classification approximative" [10] qui fournit également des partitions partielles. Nous proposons d'obtenir cette relation partielle à partir d'une matrice de scores mesurant le degré d'association possible entre chaque paire d'objets. Cette matrice peut provenir de différentes sources : il peut s'agir directement de similarités entre objets ou d'un moyennage de matrices de co-association issus de plusieurs algorithmes de clustering, par exemple. Le simple seuillage de cette matrice ne permet pas toujours d'assurer l'obtention d'une relation d'équivalence. Nous proposons dans cet article une méthode permettant de transformer une matrice de scores (ou de similarité) en une matrice relationnelle éventuellement partielle. Une description plus détaillée de ces travaux peut être trouvée dans [11]. La suite de cet article est organisée de la

manière suivante : le paragraphe 2 décrit le problème. Le paragraphe 3 décrit la méthode proposée. Enfin, le paragraphe 4 donne quelques résultats expérimentaux.

2 Concepts de base

Une partition correspond formellement à une relation d'équivalence R. Cette relation d'équivalence peut être représentée par une matrice R de taille $n \times n$, avec comme éléments $R_{i,j} \in \{0,1\}$: $R_{i,j} = 1$ si i et j sont dans la même classe et 0 sinon. Comme rappelé dans l'introduction, une relation d'équivalence est une relation réflexive (les éléments diagonaux $R_{ii} = 1$, $i = 1,\ldots,n$), symétrique ($R_{i,j} = R_{j,i}$, $i,j = 1,\ldots,n$) et transitive (si $R_{i,j} = R_{j,k}$, alors $R_{i,j} = R_{i,k}$). Nous noterons par la suite $R_{A,B}$ la sous-matrice constituée des lignes dans l'ensemble $A \subseteq \{1,\ldots,n\}$ et des colonnes dans l'ensemble $B \subseteq \{1,\ldots,n\}$.

Si la réflexivité et la symétrie peuvent facilement être vérifiées, il n'en va pas de même avec la transitivité, à moins que la matrice ne soit réarrangée sous forme de matrice diagonale par blocs. Une matrice relationnelle diagonale par blocs est composée de blocs de 1 autour de la diagonale et de zéros partout ailleurs. Plus précisément, supposons qu'il existe une partition P telle que ω_1 correspond aux n_1 premiers objets, ω_2 aux n_2 object suivants, etc. Alors, $R_{\omega_k\omega_k} = \mathbf{1}_{n_k \times n_k}$ pour tout $k = 1,\ldots,K$, et $R_{\omega_k\omega_\ell} = \mathbf{0}_{n_k \times n_\ell}$ pour tout $k \neq \ell$, où $\mathbf{1}_{n\times m}$ (respectivement, $\mathbf{0}_{n\times m}$) dénote une matrice de taille $n \times m$ remplie de 1 (respectivement, 0).

Exemple 1. Considérons l'ensemble $\{1,2,3,4\}$ avec la partition $\omega_1 = \{1,3\}$, $\omega_2 = \{2,4\}$. La matrice correspondante R est donnée sur la Figure 1(a). Notons que les deuxièmes et troisièmes lignes/colonnes peuvent être permutées de manière à obtenir une matrice diagonale par blocs (Figure 1(b)).

Dans la suite, une matrice avec des éléments $R_{i,j}$ violant les contraintes mentionnées précédemment sera qualifiée d'*incohérente*. Une matrice R est dite complète si $R_{i,j} \in \{0,1\}$ pour tout $i = 1,\ldots,n$, $j = 1,\ldots,n$. Si un élément $R_{i,j}$ est inconnu (ce que nous noterons $R_{i,j} = ?$), la matrice est dite *partielle*. Une matrice partielle est cohérente si chacun de ses éléments manquants peut être remplacé par soit un 0 soit un 1 de telle manière que la matrice résultante complète soit cohérente. Notre travail cherche à construire une telle matrice (la moins incomplète possible bien sûr) à partir de données de scores par paires entre objets. Le résultat sera un clustering incomplet puisque des relations entre objets peuvent être laissées inconnues, jusqu'à ce que de nouvelles informations puissent lever le doute. De manière alternative, on peut choisir de compléter la relation pour obtenir un ou plusieurs clustering possibles.

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad \text{(a)} \qquad \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \quad \text{(b)}$$

Figure 1 – Matrice relationnelle et matrice diagonale par blocs associée.

Exemple 2. Sur la Figure 2, on trouve un exemple de matrice incohérente pour un ensemble de 4 objets : elle ne respecte pas la symétrie (on a $R_{1,2} \neq R_{2,1}$) ni la transitivité (car $R_{1,2} = 1$ et $R_{2,4} = 1$, mais $R_{1,4} = 0$). Elle peut être relâchée en une matrice partielle cohérente, où les valeurs restantes spécifiées correspondent à celles de la matrice originale. Les complétions possibles correspondent à 3 partitions :$P_1 = \{\{1\},\{2,3\},\{4\}\}, P_2 = \{\{1,2,3\},\{4\}\}, P_3 = \{\{1\},\{2,3,4\}\}$.

Dans la suite de cet article, nous allons montrer comment, à partir d'une matrice de scores quelconque, obtenir une matrice partielle cohérente. Nous allons également montrer comment compléter une matrice partielle.

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \xrightarrow{\text{relax}} \begin{pmatrix} 1 & ? & ? & 0 \\ ? & 1 & 1 & ? \\ ? & 1 & 1 & ? \\ 0 & ? & ? & 1 \end{pmatrix}$$

(a) (b)

Figure 2 – Une matrice incohérente (a), et une matrice partielle cohérente (b).

$$\begin{pmatrix} 1 & 0.55 & 0.45 & 0.2 \\ 0.42 & 1 & 0.7 & 0.52 \\ 0.55 & 0.9 & 1 & 0.47 \\ 0.3 & 0.48 & 0.45 & 1 \end{pmatrix}$$

Matrice de scores initiale

$$\varepsilon = 0.1 \downarrow$$

$$\begin{pmatrix} 1 & [0.45, 0.65] & [0.35, 0.55] & [0.1, 0.3] \\ [0.32, 0.52] & 1 & [0.6, 0.8] & [0.42, 0.62] \\ [0.45, 0.65] & [0.8, 1] & 1 & [0.37, 0.57] \\ [0.2, 0.4] & [0.38, 0.58] & [0.35, 0.55] & 1 \end{pmatrix}$$

Matrice de scores imprécis

Figure 3 – Matrice de scores probabilistes incohérente S, correspondant à la matrice relationnelle partielle R de la Figure 2 ; scores imprécis obtenus avec $\varepsilon = 0.1$.

3 Obtention et manipulation de matrices relationnelles partielles

Dans ce travail, on suppose que l'information d'entrée est disponible sous forme d'une matrice S de scores de taille $n \times n$. Nous interprétons S_{ij} comme une information sur la relation entre les objets i et j. De plus, nous supposons que les scores sont compris dans un intervalle $[a, b]$, plus le score est proche de a, moins il est probable que les objets soient liés. Plus il est proche de b, plus la liaison est probable. On suppose de plus qu'il existe un élement neutre $c \in [a, b]$ selon lequel des relations plus ou moins certaines peuvent être identifiées : des scores $S_{ij} \in [a, c)$ (respectivement, $\in [c, b]$) supportent la conclusion $R_{i,j} = 0$ (resp., $R_{i,j} = 1$), et ce support est consideré comme faible si le score est proche de c. En pratique, on choisira souvent $c = {}^{(a+b)}/_{2}$. Par exemple, des algorithmes probabilistes fourniront un score $S_{ij} \in [0, 1]$, avec 0.5 comme élément neutre.

Il est facile de transformer S en une matrice binaire R en fixant $R_{i,j} = 0$ si $S_{ij} < c$, et $R_{i,j} = 1$ dans le cas contraire. Cependant, si les scores S_{ij} sont estimés indépendamment les uns des autres, la matrice résultante a des chances d'être incohérente. De manière alternative, on peut définir une matrice partielle R^{ε} telle que $R^{\varepsilon}_{ij} = 0$ si $S_{ij} < c - \varepsilon$, $R^{\varepsilon}_{ij} = 1$ si $S_{ij} \geq c + \varepsilon$ et $R^{\varepsilon}_{ij} = ?$ sinon, pour un $\varepsilon \geq 0$. Cela revient à transformer la matrice S en une matrice de scores imprécis et à poser que $R^{\varepsilon}_{ij} = ?$ lorsque $c \in [S_{ij} - \varepsilon, S_{ij} + \varepsilon] \cap [a, b]$. Un exemple est donné en Figure 3.

On peut alors chercher les valeurs ε correspondant à des matrices R^{ε} cohérentes. La plus petite valeur satisfaisant cette condition, correspondant à la plus petite altération de la matrice initiale S, sera notée dans la suite ε^* et la matrice partielle associée R^*. Calculer ε^* et R^* est un des objectifs de ce travail. Notons que, plutôt que d'affaiblir S en R^{ε}, on pourrait définir des intervalles sur les scores par exemple avec des intervalles de confiance. On peut aussi imaginer que l'on dispose d'intervalles dès le début. Notre méthode s'applique facilement à ces cas.

3.1 Contrôler et obtenir la cohérence

Comme expliqué plus haut, le but est de calculer l'altération minimale des scores conduisant à une matrice relationnelle cohérente. Cet objectif peut être atteint en partant de $\varepsilon = 0$ et en augmentant progressivement sa valeur jusqu'à ce que R^{ε} soit cohérente, ou bien par une recherche dichotomique entre 0 et $\overline{\varepsilon}$ avec, par exemple, $\overline{\varepsilon} = \max_{i,j} |R_{ij} - c|$ (voir l'Algorithme 1).

La cohérence requiert de satisfaire la réflexivité, la symétrie et la transitivité. Comme nous l'avons déjà souligné, vérifier les deux premières propriétés est immédiat. Pour la transitivité, notre stratégie repose sur le fait que le graphe d'une relation correspondant à une par-

Algorithme 1: Déterminer ε^*

Input: Matrice de scores S, precision δ, $\overline{\varepsilon}$
Output: ε^*
$\varepsilon_* \leftarrow 0$;
$\varepsilon^* \leftarrow \overline{\varepsilon}$;
$\varepsilon \leftarrow {}^{(\varepsilon_* + \varepsilon^*)}/_2$;
while $|\varepsilon^* - \varepsilon| \geq \delta$ **do**
  $\mathrm{Cons}(\varepsilon) \leftarrow$ sortie de l'algorithme 2 pour R^ε ;
  if $\mathrm{Cons}(\varepsilon) = 1$ **then** $\varepsilon^* \leftarrow \varepsilon$ **else** $\varepsilon_* \leftarrow \varepsilon$;
  $\varepsilon \leftarrow {}^{(\varepsilon_* + \varepsilon^*)}/_2$;

tition est un ensemble de cliques disjointes : chaque composante connexe D_i est fortement connectée. [1]

Proposition 1. *Une matrice partielle R est cohérente si et seulement si chaque composante connexe D_i du graphe correspondant peut être complétée pour obtenir une composante fortement connexe (une clique), i.e. $0 \notin R_{D_i,D_i}$.*

L'algorithme 2 décrit une méthode simple pour vérifier la cohérence, dérivée de la Proposition 1 ; elle se fonde sur l'utilisation d'une méthode permettant d'extraire les composantes connexes d'un graphe (voir, par exemple, [7]).

Algorithme 2: Vérification de la cohérence de R

Input: matrice partielle R
Output: Variable Cons (0=incohérente, 1=cohérente)
Cons=1;
Extraction des composantes connexes $D_1, \ldots, D_L$;
foreach component D_i **do**
  if $\exists k, l \in D_i^2$ with $R_{k,l} = 0$ **then** set Cons=0 and stop;
return Cons

1. Rappelons qu'un sous-ensemble de noeuds D_i est une composante connexe s'il existe un chemin entre chaque paire de noeuds $k, l \in D_i$; c'est une composante fortement connexe si, pour tout $k, l \in D_i$, (k, l) est un arc du graphe. Ceci reste vrai pour les graphes partiels.

3.2 Déduire des valeurs dans des matrices partielles cohérentes

Etant donnée une matrice partielle cohérente, il est possible de déduire un certain nombre de valeurs manquantes en exploitant les propriétés de symétrie et de transitivité de la relation. Par exemple, si $R_{k,l}$ est connu, on peut immédiatement déduire la valeur de $R_{l,k} = ?$. On peut également exploiter la transitivité, par exemple si $R_{k,l} = 1$ et $R_{l,m} = 1$ alors que $R_{k,m} = ?$. On procède à tous les remplacements nécessaires en deux étapes :

– pour chaque composante connexe D_i du graphe de R (obtenue par l'algorithme 2), on a $R_{D_i,D_i} = \mathbf{1}_{|D_i|}$, qui résulte de la transitivité : cela signifie que tout $? \in R_{D_i,D_i}$ doit être remplacé par un 1.
– puis, si pour une paire d'objets appartenant à deux composantes différentes D_i, D_j, $i \neq j$, on observe un $0 \in R_{D_i,D_j}$ (et donc $0 \in R_{D_j,D_i}$ par symétrie), R ne peut donc pas être complétée de telle sorte que $R_{D_i \cup D_j, D_i \cup D_j}$ forme une matrice unité (c.à.d. les composantes D_i et D_j ne peuvent pas être agrégées en une seule composante fortement connexe) : tous les éléments manquants doivent être remplacés par des 0.

Une fois que tous ces remplacements ont été effectués, le reste des éléments manquants est impossible à déduire sans information additionnelle. Ils pourraient en principe être remplacés soit par des 1 soit par des 0 — pourvu que les remplacements soient compatibles les uns avec les autres. Une illustration de cette procédure est donnée en Figure 4, dans laquelle on identifie d'abord les liens obligatoires ($?$ dans R à remplacer par des 1), puis toutes les séparations obligatoires ($?$ dans R à remplacer par des 0). Le résultat final est toujours imprécis puisqu'il reste quelques $?$. Il est clair que le résultat correspond à un unique clustering si et seulement si la matrice ne contient plus d'éléments inconnus. Dans ce cas, la procédure peut être vue comme "réparant" la matrice initiale.

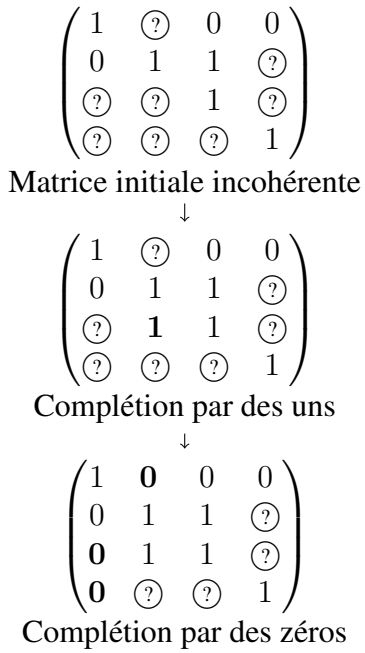

Figure 4 – Déduire des éléments de la matrice.

3.3 Enumérer les complétions : une discussion

Un clustering partiel comme celui décrit au paragraphe 3.2 permet à l'utilisateur des inférences prudentes. En effet, il est possible d'identifier les relations fortement supportées par les données, et de s'abstenir lorsque l'information est trop incertaine. On peut envisager par la suite un apprentissage actif des relations manquantes. Dans certains contextes, il est naturel d'estimer combien de complétions pourraient être faites et combien de classes pourraient résulter de ces complétions, puisque notre approche ne nécessite pas de fixer le nombre de classes à l'avance. Une solution à ce problème peut être de fournir des bornes inférieure et supérieure $(\underline{K}, \overline{K})$ du nombre de classes. Déterminer une borne supérieure exacte est très simple, puisque remplacer tous les ⑦ par des zéros sépare complètement les composantes connexes les unes des autres : si $D_1, \ldots, D_L$ sont les composantes connexes de R, alors $\overline{K} = L$.

Par contre, une borne inférieure ne peut pas être obtenue simplement en remplaçant tous les éléments manquants par des 1, puisque la matrice résultante peut être incohérente (c'est le cas sur la Figure 2). Pour déterminer $\underline{K}$, nous considérons le graphe G représentant l'absence de relation entre objets : les sommets du graphe sont les objets, et un arc relie deux objets i et j si et seuelement si $R_{i,j} = 0$. Il est clair que chaque classe correspond à un ensemble d'objets qui ne sont pas connectés dans G et que chaque paire d'objets connectés dans G doit appartenir à des classes différentes. Le nombre minimum de classes peut donc être obtenu en résolvant un problème de coloration du graphe G [8]. Si C est le nombre minimum de couleurs requises (nombre chromatique) pour que deux sommets connectés n'aient pas la même couleur, alors $\underline{K} = C$.

Le problème d'explorer toutes les complétions possibles d'une matrice partielle cohérente est plus complexe. Pour toute sous-matrice $R_{D_i \cup D_j, D_i \cup D_j}$ contenant des 1 et des ⑦, on peut soit agréger D_i et D_j en une classe soit les laisser séparées : le nombre de complétions possibles croît de manière exponentielle avec le nombre L de composantes connexes, qui est lui-même lié au nombre n d'objets à classer. La valeur de n étant généralement élevée, on peut s'attendre à ce que le nombre de complétions soit lui-même élevé : en conséquence, déterminer l'ensemble exhaustif des complétions n'est pas envisageable d'un point de vue calculatoire. On peut cependant proposer une méthode simple pour échantillonner dans l'ensemble des complétions possibles :

1. Soit une paire (i, j) pour laquelle $R_{i,j} =$ ⑦
2. Grâce au tirage aléatoire d'une variable de Bernoulli de paramètre p, décider si $R_{i,j} = 1$ or $R_{i,j} = 0$;
3. Effectuer les deux étapes du paragraphe 3.2 pour compléter la matrice ;
4. Répéter l'opération jusqu'à ce que la matrice soit complète.

Notons que le paramètre p va directement in-

fluencer le nombre moyen de classes : une valeur élevée de p (probabilité d'observer un 1) induira un nombre faible de classes et vice-versa. Notons aussi que cette procédure ne permet pas d'échantillonner de manière uniforme dans l'ensemble des complétions possibles. Pour finir, on peut considérer le problème de déterminer une complétion spécifique satisfaisant certaines propriétés. Par exemple, on peut fixer le nombre de classes, ou encore choisir la complétion qui minimise l'écart entre la partition dure et la matrice de scores initiale S. Notons, cependant, que si l'objectif est d'obtenir une partition dure unique à partir des données, l'avantage de d'abord déterminer une partition prudente puis de la compléter, n'est pas très clair, en comparaison de l'utilisation directe de méthodes classiques (comme par exemple le clustering spectral [14]).

4 Résultats expérimentaux

4.1 Données de type intervalle

Nous illustrons notre approche dans le cas particulier de données de type intervalle. Nous utilisons le jeu de données Cars [4] qui consiste en 33 modèles de véhicules décrits par 8 intervalles (prix, capacité du moteur, vitesse max, marche, accéleration, longueur, largeur, hauteur). Ces véhicules sont *a priori* classé en 4 catégories : Utilitaires, Berlines, Sportives et Luxueuses.

Une représentation 2D obtenue par une ACP sur données intervalles [5] est donnée en Figure 5. Nous appliquons notre méthode sur une matrice de co-association moyenne obtenue en rééchantillonnant les données initiales. Plus précisement, on tire au hasard un échantillon précis dans l'ensemble de données intervalles initial, et on applique une procédure standard des c-moyennes floues [1] avec $K = 4$ classes. Ce procédé est répété 50 fois et les matrices de co-association sont moyennées pour obtenir la matrice de scores. En utilisant un élément neutre $c = 0.5$, on trouve $\varepsilon^* = 0$, ce qui signifie que la matrice de scores est cohérente dès le départ.

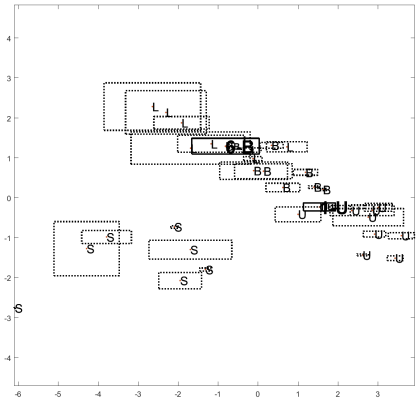

Figure 5 – Representation du jeu de données *Cars*

La partition obtenue est représentée sur la Figure 6. Pour des raisons de clarté, nous n'avons pas représenté la transitivité dans chaque classe. A la place, un objet a été choisi arbitrairement dans chaque classe et utilisé pour représenter les relations intra et inter-cluster. La partition trouvée est cohérente avec ce qui est rapporté dans la littérature (voir par exemple [3]). De manière intéressante, fixer manuellement $\varepsilon = 0.25$ et compléter la matrice résultante avec des 1 et des 0 donne le clustering prudent représenté sur la Figure 7. Dans cette représentation, lorsqu'une relation entre objets est manquante, elle est représentée par une arête en pointillés. On voit que le clustering partiel isole les véhicules *6-B* et *1-U* (indiqués en gras sur la Figure 5), qui étaient précedemment mal classés. Ces véhicules sont identifiés comme appartenant potentiellement à deux classes, parmi lesquelles se trouve la bonne.

4.2 Compromis complétude-précision

Un comportement attendu de notre méthode est que la pertinence des liens restants augmente avec le taux d'abstention. Pour des raisons de place, nous illustrons ce comportement avec un unique jeu de données *segment* issu de l'UCI Machine Learning Repository [9]. Des résultats très similaires ont été trouvés avec 5 autres jeux de données de l'UCI

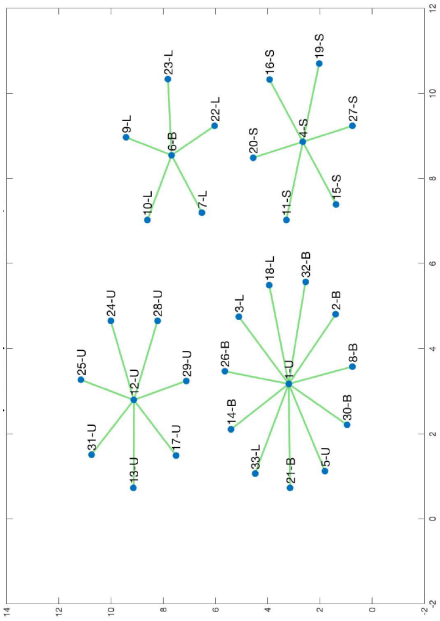

Figure 6 – Clustering du jeu de données *Cars* en 4 classes ($\epsilon = 0$)

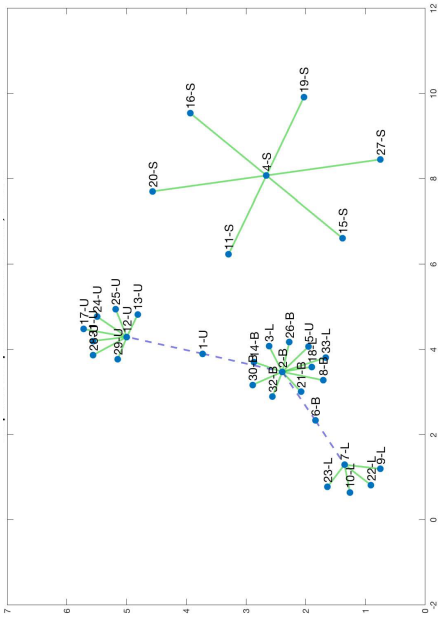

Figure 7 – Clustering du jeu de données *Cars* en 4 classes ($\epsilon = 0,25$).

(iris,wine,ecoli,seeds et optdigits). On applique la procédure suivante : on crée des échantillons du booststrap en rééchantillonnant avec remise le jeu de données initial. Un modèle de mélange gaussien est ajusté sur chaque échantillon grâce au logiciel `MIXMOD`, un logiciel de classification supervisée et non supervisée disponible à l'adresse `http://www.mixmod.org/` (en utilisant un nombre de composants égal au nombre de classes). On l'utilise pour calculer la probabilité que chaque paire d'instances de l'ensemble initial appartienne à la même classe. Ce processus est répété 20 fois et les matrices de co-association sont moyennées. Puisque notre approche conduit potentiellement à un clustering partiel, on utilise deux mesures pour l'évaluation du résultat. La première est une extension de l'indice de Rand [13] qui est classiquement utilisé pour évaluer la similarité entre deux partitions. L'indice de Rand calcule la proportion de paires d'objets pour lesquels les deux partitions sont en accord : soit les deux objets sont dans la même classe, soit les deux objets sont dans deux classes différentes pour les deux partitions. Une extension naturelle de cet indice consiste à calculer la proportion d'accords uniquement sur les paires pour lesquelles $R^{\epsilon}_{i,j} \neq ?$. Le deuxième critère d'évaluation est la complétude de la relation. Une bonne méthode doit dans l'idéal voir son indice de Rand augmenter lorsque la complétude décroît. Le compromis est contrôlé par la valeur de ϵ. La Figure 8 présente le résultat obtenu en faisant varier la valeur de ϵ. On voit que la complétude n'est jamais égale à un, ce qui signifie que les matrices intiales ne sont pas cohérentes. Comme attendu, s'abstenir de s'exprimer sur certaines relations permet d'accroître les performances. En effet, l'indice de Rand est augmenté de manière significative, au prix d'une décroissance raisonnable de la complétude.

5 Conclusion

Dans cet article, nous avons proposé une approche permettant de produire un clustering partiel des objets à partir de données de scores de relation entre les paires d'objets. La première étape de la méthode consiste à affaiblir la matrice de scores sous forme d'une matrice

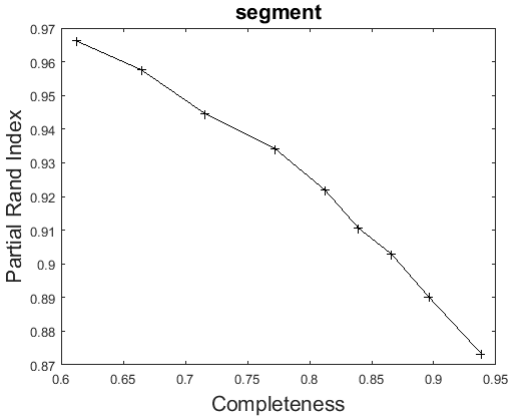

Figure 8 – Résultats de clustering sur les données *Segment*

de scores imprécis de telle manière qu'après seuillage on ne viole pas les propriétés de symétrie et de transitivité. La matrice obtenue est ensuite complétée en utilisant ces deux propriétés. Si la matrice complétée comporte encore des valeurs manquantes, alors le clustering est imprécis car plusieurs clusterings peuvent être produits à partir de la matrice incomplète. Nous avons montré sur quelques expériences l'intérêt de la méthode. Dans le futur, nous envisageons de coupler cette méthode avec une méthode d'apprentissage actif pour améliorer les résultats du clustering.

Références

[1] J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[2] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7) :719–725, 2000.

[3] F. De Carvalho and Y. Lechevallier. Partitional clustering algorithms forsymbolic interval data based on single adaptive distances. *Pattern Recognition*, 42 :1223–1236, 2009.

[4] F. De Carvalho, R. De Souza, M. Chavent, and Y. Lechevallier. Adaptative hausdorff distances and dynamic clustering of symbolic interval data. *Pattern Recognition Letters*, 27(3) :167–179, 2006.

[5] P. Cazes, A. Chouakria, E. Diday, , and Y. Schektman. Extension de l'analyse en composantes principales à des données de type intervalle. *Revue de Statistique Appliquée*, 14 :5–24, 1997.

[6] J. De Oliveira and W. Pedrycz. *Advances in fuzzy clustering and its applications*. John Wiley & Sons, 2007.

[7] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao. The connected-component labeling problem : a review of state-of-the-art algorithms. *Pattern Recognition*, 70 :25–43, 2017.

[8] R.M.R. Lewis. *A Guide to Graph Colouring : Algorithms and Applications*. Springer, 2016.

[9] M. Lichman. UCI machine learning repository, 2013.

[10] P. Lingras. Unsupervised rough set classification using gas. *Journal of Intelligent Information Systems*, 16 :215–228, 2001.

[11] M.-H. Masson, B. Quost, and S. Destercke. Cautious relational clustering : A thresholding approach. *Expert Systems with Applications*, 139, 2020.

[12] M.H. Masson and T. Denoeux. ECM : An evidential version of the fuzzy c-means algorithm. *Pattern Recognition*, 41(4) :1384–1397, 2008.

[13] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336) :846–850, 1971.

[14] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4) :395–416, 2007.