Apport des modèles locaux pour les K-moyennes prédictives

Vincent Lemaire, Oumaima Alaoui Ismaili

2 Avenue Pierre Marzin, 22300 Lannion
(vincent,oumaima)@orange.com

Résumé. Dans le cadre du clustering prédictif, pour attribuer la classe aux groupes formés à la fin de la phase d'apprentissage, le vote majoritaire est la méthode communément utilisée. Cependant, cette approche comporte certaines limitations qui influent directement sur la qualité des résultats obtenus en termes de prédiction. Pour surmonter ce problème, nous proposons d'incorporer des modèles prédictifs localement dans les clusters formés afin d'améliorer la qualité prédictive du modèle global. Les résultats expérimentaux montrent que cette incorporation permet d'obtenir des résultats (en termes de prédiction) significativement meilleurs par rapport à ceux obtenus en utilisant le vote majoritaire ainsi que des résultats très compétitifs avec ceux obtenus par des algorithmes performants d'apprentissage supervisé "similaires". Ceci est effectué sans dégrader le pouvoir descriptif (explicatif) du modèle global.

1 Introduction

L'algorithme des K-moyennes prédictives (Eick et al., 2004; Al-Harbi et Rayward-Smith, 2006; Alaoui Ismaili, 2016) est une version modifiée de l'algorithme des K-moyennes standard. Il vise à décrire et à prédire d'une manière simultanée.

L'idée est de générer dans la phase d'apprentissage un nombre minimal de clusters compacts dont les instances doivent appartenir à la même classe. Ces clusters vont servir par la suite à décrire les données et à prédire la classe des nouvelles instances (voir la figure 1).

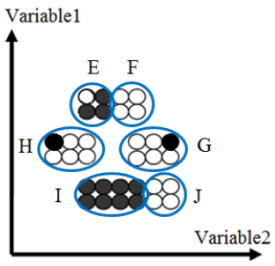

FIG. 1 – *Objectif du clustering prédictif*

La méthode communément utilisée dans la littérature permettant d'attribuer la classe aux clusters formés par l'algorithme des K-moyennes prédictives est le vote majoritaire. Bien que cette approche parvienne à obtenir de bons résultats, celle-ci a également certaines limites. On citera par example :

— pour le taux de bonne classification (ACC) : si un cluster contient Q% d'instances de la classe C1 et 100-Q% d'instances de la classe C2, alors l'utilisation du vote majoritaire va produire un taux de mauvaise classification très important (Q). La présence d'un modèle local à ce cluster devrait permettre de mieux discriminer les exemples selon leur classe d'appartenance. Ceci est très visible pour les clusters E, H, G de la figure 1.

— pour l'aire sous la courbe de ROC (AUC) : le fait de se baser sur la classe majoritairement présente dans un cluster produit une courbe de ROC, ou de lift, ayant l'allure de la courbe rouge dans la figure 2. Le classement des exemples se fait par cluster et la courbe comporte des segments. L'aire sous la courbe de ROC est sous-optimale (voir la figure 2).

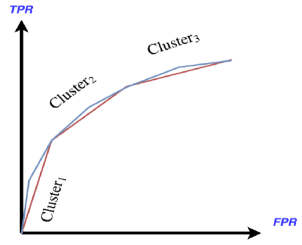

FIG. 2 – *Illustration courbes de ROC*

Pour surmonter ces problèmes, on propose dans cet article d'entrainer un modèle d'apprentissage localement à chaque cluster formé dans la phase d'apprentissage par l'algorithme des K-moyennes prédictives. Le modèle global appris dans cette phase va être par la suite utilisé pour prédire la classe des nouvelles instances.

Le reste de ce papier est organisé comme suit : la section 2 présente les différentes étapes de l'algorithme des K-moyennes prédictives utilisé dans cette étude ainsi que la méthode proposée pour surmonter les problèmes cités ci-dessus. La section 3 présente une brève description des classifieurs qui seront comparés à l'approche proposée, le protocole expérimental et les résultats obtenus. Puis la section 4 présente la partie analyse descriptive du modèle global obtenu. Finalement la section 5 conclue cet article et présente des pistes d'améliorations pour le futur.

2 Les K-moyennes prédictives et choix des modèles locaux

2.1 K-moyennes prédictives

L'algorithme des K-moyennes utilisé dans cette étude est l'algorithme proposé dans (Alaoui Ismaili, 2016). Cet algorithme fourni de meilleurs résultats (en termes de prédiction) par rapport aux algorithmes de clustering prédictif les plus répandus dans la littérature tels que l'algorithme de Eick (Eick et al., 2004) et l'algorithme de Al-Harbi (Al-Harbi et Rayward-Smith, 2006). Il est présenté succinctement ci-dessous (voir Algorithme 1). Il correspond à l'algorithme des k-moyennes classique mais où de la supervision a été incorporée dans certaines des étapes. La section 3.1 donnera plus détails sur chacune des étapes supervisées de l'algorithme 1.

L'objectif de l'étude qui sera menée au cours du présent article sera de mesurer le fait de changer le vote majoritaire (étape 5) en incorporant des modèles locaux au sein des clusters obtenus à la fin de la convergence de l'algorithme des k-moyennes prédictives.

2.2 Choix pour les modèles locaux

Afin de surmonter les problèmes rencontrés par le vote majoritaire lors de l'attribution des classes aux clusters formés par l'algorithme des K-moyennes prédictives, nous pensons que l'ajout de modèles locaux (ajout d'un classifieur localement à chaque cluster) est une solution. Cependant, il est nécessaire que les modèles locaux :

— puissent être entrainés éventuellement avec peu de données
— soient robustes (ratio performances train/test)

Entrée
- Un ensemble de données D, où chaque instance X_i est décrite par un vecteur de d dimensions et par une classe $Y_i \in \{1, \ldots, J\}$.
- Le nombre de clusters, noté K.

Début
1) Prétraitement supervisé des données (Ismaili et al., 2016)
2) Initialisation supervisée des centres (Lemaire et al., 2015)
Répéter
3) *Affectation :* générer une nouvelle partition en assignant chaque instance X_i au groupe dont le centre est le plus proche.
4) *Représentation :* calculer les centres associés à la nouvelle partition
jusqu'à ce que (convergence de l'algorithme)
5) Attribution des classes aux clusters formés : *vote majoritaire*

---

6) Prédiction de la classe des nouvelles lors de la phase de déploiement :
→ *classe du cluster le plus proche*

Fin

Algorithme 1: Algorithme des K-moyennes prédictives

— idéalement puissent ne comporter aucun paramètre utilisateur afin de ne pas avoir à réaliser de cross-validation localement à un cluster.
— aient une complexité algorithmique linéaire en apprentissage (en $O(N)$)
— ne soient pas créés si localement au cluster il n'y pas d'information suffisante pour la création d'un modèle local (dans ce cas et pour ce cluster le vote majoritaire serait conservé)
— conservent (voire améliorent) les qualités d'interprétation initiales du modèle global
— individualisent les prédictions dans un cluster donné afin d'améliorer l'AUC (courbe bleue vis-à-vis de la courbe rouge dans la figure 2).

Une large étude a été réalisée dans (Salperwyck et Lemaire, 2011) afin d'étudier la vitesse d'apprentissage des classifieurs les plus couramment utilisés. Vitesse au sens du nombre d'exemples utilisés versus les performances en classification. Cette étude montre la capacité d'un classifieur naïf de Bayes à apprendre avec peu de données (confirmant (Bouchard et Triggs, 2004)) ; que ce soit dans sa version standard ou dans la version où les variables reçoivent des poids (on parle alors de ANB (Averaging Naive Bayes) ou SNB (Selective Naive Bayes)) (Langley et Sage, 1994). On trouve de plus dans les travaux décrits dans (Boullé, 2007a) un critère analytique issu de la famille MODL (Bondu et al., 2013) (muni de son algorithme d'optimisation) permettant d'apprendre un Selective Naive Bayes (SNB) sans paramètre utilisateur et qui est de plus régularisé. Cette régularisation permet d'exhiber de bonnes performances tout en assurant une très bonne robustesse (ratio des performances entre l'apprentissage et le test proche de 1).

Pour ce SNB les prétraitements et le calcul des poids des variables sont basés sur l'approche MODL (Boullé, 2007b). On y trouve ici un point intéressant : si l'approche MODL ne découvre pas assez d'information dans une base de données alors elle sait se taire au sens dire "qu'aucune variable n'est informative". Dans ce cas le vote majoritaire sera alors consi-

déré comme meilleur à un autre choix. Dans notre cas applicatif l'intérêt est donc qu'on ne retournera un SNB localement à un cluster si et seulement si il est plus "informatif" que le vote majoritaire. Dans le cas contraire le vote majoritaire sera conservé. L'autre intérêt de l'approche MODL est qu'elle est auto-régularisée : il n'y a pas besoin de faire une cross validation pour trouver les bons paramètres d'un modèle. En termes d'interprétation un SNB s'interprète aisément (Lemaire et al., 2009). Par exemple, le poids des variables peut être utilisé pour décrire leur importance respective.

Nous décidons donc qu'un SNB de la famille MODL sera donc entrainé localement dans chacun des K clusters formés (fin de l'étape 4 de l'algorithme 1) : Pour chaque cluster $cluster_l$ ($l \in \{1, \ldots, K\}$), un classifieur SNB est entrainé, notons SNB_l le modèle obtenu.

3 Expérimentation

3.1 Classifieurs de comparaison

Afin d'étudier l'impact de l'utilisation des modèles locaux sur la qualité des résultats issus de l'algorithme des K-moyennes prédictives, nous allons comparer ses performances prédictives avec celles obtenues par la méthode utilisant le vote majoritaire et avec celles obtenues par trois autres algorithmes performants dans le cadre de la classification supervisée.

Le premier modèle combine les arbres de décision avec la régression logistique (Logistic Model Tree (LMT)) (Landwehr et al., 2005), le deuxième modèle combine les arbres de décision avec des classifieurs naïfs de Bayes (Naives Bayes Tree (NBT)) (Kohavi, 1996). Ces modèles ont été choisis pour leurs bonnes performances mais aussi pour leur proximité avec l'approche proposée qui consiste à avoir des modèles locaux entrainés sur une partie des données. Enfin comme notre modèle hybride utilise des $SNBs$ localement aux clusters nous ajoutons en juge paix un SNB 'global' qui lui est entrainé sur l'ensemble des données.

Note : Nous avons décidé de centrer ici l'analyse sur l'algorithme des K-moyennes en termes de classification et de ne pas comparer la méthode proposée avec des algorithmes de clustering concurrents avec lesquels il est parfaitement possible de faire de la prédiction. On concentre la comparaison dans cet article avec d'autres méthodes hybrides telles que LMT et NBT contenant des modèles locaux. La comparaison avec d'autres méthodes de clustering a néanmoins été partiellement déjà réalisée dans (Alaoui Ismaili, 2016) et l'algorithme des K-moyennes prédictives muni du vote majoritaire était déjà très bien positionné. On donne ci-dessous une brève description de LMT, NBT et du SNB.

3.1.1 Naive Bayes Tree (NBT)

L'arbre de "Naive Bayes" (NBT) est un algorithme d'induction d'arbres de décision avec des classifieurs naïfs Bayésiens dans ses feuilles (Kohavi, 1996). L'arbre est induit d'une manière descendante (top-down) avec des segmentations univariées en se basant sur le gain d'information. Un pré-élagage est utilisé lors de la phase d'entrainement de l'arbre pour décider si le nœud sera partitionné ou bien il est terminal. Dans ce cas, un modèle naïf Bayésien est entrainé localement sur les instances de la feuille. NBT est un classifieur qui a de bonnes performances par comparaison soit avec les arbres de décision soit avec le classifieur naïf Bayes seul.

3.1.2 Logistic Model Tree (LMT)

L'arbre de régression logistique (LMT) (Landwehr et al., 2005) est un modèle de classification basé sur un algorithme d'apprentissage supervisé qui combine la régression logistique et les arbres de décision. L'objectif est d'améliorer les performances de la classification obtenues par les arbres de décision. A cette fin, au lieu d'associer à chaque feuille un seul label et un seul vecteur de probabilité (piecewise constant model), un modèle de régression logistique est entrainé sur les feuilles afin d'estimer, pour chaque exemple de test, un vecteur de probabilités plus adapté (piecewise linear regression model). L'algorithme LogitBoost est employé pour ajuster un modèle de régression logistique à chaque nœud , ensuite le nœud est partitionné en utilisant le gain d'information comme fonction d'impureté. L'appel de l'algorithme LogitBoost dans chaque nœud utilise comme point initial le modèle obtenu dans le nœud parent. Finalement, l'arbre est élagué au moyen de l'algorithme d'élagage de CART. Le nombre d'itérations de LogitBoost dans chaque nœud est déterminé par une validation croisée pour éviter le sur-apprentissage.

3.1.3 Selective Naive Bayes (SNB)

Le classifieur naïf Bayésien est un outil largement utilisé dans les problèmes de classification supervisée. Il a pour avantage de se montrer efficace pour de nombreux jeux de données réels (Hand et Yu, 2001). Cependant, l'hypothèse naïve d'indépendance des variables peut, dans certains cas, dégrader les performances du classifieur. Aussi, des méthodes proposant de réaliser de la sélection de variables ont vu le jour (Langley et Sage, 1994). Elles consistent en la mise en place d'heuristiques d'ajout et de suppression de variables afin de sélectionner le meilleur sous-ensemble de variables maximisant un critère et donc les performances du classifieur, selon une approche wrapper (Guyon et Elisseeff, 2003). Il a été montré par Boullé (Boullé, 2007a) que moyenner un grand nombre de classifieurs Bayésiens naïfs sélectifs, réalisés avec différents sous-ensembles de variables, revenait à ne considérer qu'un seul modèle avec une pondération sur les variables. La formule de Bayes sous l'hypothèse d'indépendance des variables conditionnellement aux classes devient : $P(C_k|X) = \frac{P(C_k)\prod_i P(X_i|C_k)^{W_i}}{\sum_{j=1}^{K}(P(C_j)\prod_i P(X_i|C_j)^{W_i})}$ où W_i représente le poids de la variable i. La classe prédite est celle qui maximise la probabilité conditionnelle $P(C_k|X)$. Les probabilités $P(X_i|C_i)$ peuvent être estimées par intervalle à l'aide d'une discrétisation pour les variables numériques. Pour les variables catégorielles, cette estimation peut se faire directement si la variable prend peu de modalités différentes ou après un groupage dans le cas contraire.

3.1.4 K-moyennes prédictives (KM_{VM}, KM_{SNB})

Cet algorithme est présenté brièvement en section 2.1. Il correspond à l'algorithme des k-moyennes classique mais où 3 étapes ont été modifiées pour incorporer de la supervision :

• **Prétraitement des données** : Généralement, la tâche de clustering nécessite une étape de prétraitement non supervisé afin de fournir des clusters intéressants (pour l'algorithme des K-moyennes voir par exemple (Milligan et Cooper, 1988) et (Celebi et al., 2013)). Cette étape de prétraitement peut empêcher certaines variables de dominer lors du calcul des distances. En s'inspirant de ce résultat, Alaoui et al. ont montré dans (Ismaili et al., 2016) que l'utilisation

d'un prétraitement supervisé peut aider l'algorithme des K-moyennes standard à atteindre une bonne performance prédictive. Nous utilisons ici la méthode qu'ils ont proposée et nommée Conditional-Info. L'avantage de ce prétraitement supervisé est d'introduire une distance Bayésienne qui donne des garanties en termes de proximité des instances dans un cluster connaissant leur classe d'appartenance ((Alaoui Ismaili, 2016) chapitre 3).

• **Initialisation des centres :** Nous utilisons la méthode supervisée d'initialisation des k-moyennes décrite dans (Lemaire et al., 2015). Cette méthode est basée sur l'idée de la décomposition des classes (Vilalta et al., 2003; Basu et al., 2002). Dans le cas ou $K = C$ chaque centre initial correspond au centre de gravité d'une classe, si $K > C$ les centre suivants sont initialisés à l'aide de la méthode kmeans++ (Arthur et Vassilvitskii, 2007). Dans le cas où $K = C$, la méthode étant déterministe, l'algorithme de convergence des K-moyennes n'est donc réalisé qu'une seule fois.

• **Prédiction de la classe :** Une nouvelle instance en test X, est tout d'abord affectée au cluster, l, dont elle est le plus proche puis on distingue le cas où : (i) la classe prédite est la classe majoritaire présente dans ce cluster (KM_{VM}), (ii) la prédiction de la classe s'effectue selon la règle de décision du modèle SNB_l : $P(C|X) = argmax_{1 \leq j \leq J}(P_{SNB_l}(C_j|X))$ s'il existe un modèle local sinon le vote majoritaire est conservé (KM_{SNB}). Ces deux approches seront comparées plus loin dans l'article.

• **Nombre de clusters :** Cet article présente une première expérimentation des K-moyennes prédictives munies de modèles locaux. Nous avons arbitrairement décidé de fixer $K = C$ dans ce cadre. Il est évident que l'intérêt de travailler dans un contexte ou $K = C$ (i.e. nombre de clusters = nombre de classes à prédire) limite sérieusement l'apport du clustering prédictif relativement à une méthode concurrente de classification appliquée sur les mêmes données. L'exploitation directe d'un bon classifieur sera sûrement meilleure dans ce contexte. Cela étant dit, nous pourrons néanmoins évaluer si l'utilisation de modèles locaux est meilleure que celle du vote majoritaire. Si de plus avec $K = C$ les résultats sont proches des méthodes concurrentes de l'état de l'art alors des travaux futurs réglant la valeur de K seront surement intéressant à mener.

3.2 Protocole expérimental

3.2.1 Base de données utilisées

Pour évaluer et comparer les différents algorithmes, nous allons effectuer des tests sur différents jeux de données de l'UCI (Lichman, 2013). Ces jeux de données ont été choisis afin d'avoir des bases de données diverses en termes de nombre de classes C, de variables (continues V_n et/ou catégorielles V_c) et d'instances N (voir Tableau 1). Elles ont été choisies (sauf Adult) dans la liste de comparaison de l'article comparant LMT et NBT (Landwehr et al., 2005). Le lecteur pourra noter que ce sont de "petites" bases (sauf Adult).

3.2.2 Entrainement des modèles - éléments de reproductibilité

Les codes utilisés pour l'apprentissage :

— de LMT et de NBT sont ceux contenus dans le logiciel R (Hornik, 2017) (qui utilise des wrappers sur Weka (Hall et al., 2009)).

Données	Instances	$\#V_n$	$\#V_c$	# Classes
Glass	214	10	0	6
Pima	768	8	0	2
Vehicle	846	18	0	4
Segmentation	2310	19	0	7
Waveform	5000	40	0	3
Mushroom	8416	0	22	2
Pendigits	10992	16	0	10
Adult	48842	7	8	2

TAB. 1 – *Jeux de données utilisés, V_n : Variables numériques, V_c : Variables catégorielles.*

— des SNB : nous avons obtenu une licence provisoire du logiciel Khiops (Boullé, 2016) qui nous a permis de produire les SNB globaux (Boullé, 2007a)
— des KM_{VM} et KM_{SNB} : pour pouvoir réaliser les éléments de supervision de l'apprentissage de l'algorithme des K-moyennes décrits dans la section 3.1.4 nous avons aussi utilisé le logiciel Khiops afin de produire les SNB locaux aux clusters et les méthodes de prétraitement contenues dans (Ismaili et al., 2016), l'algorithme des K-Moyennes classique étant lui réalisé en Matlab (MATLAB, 2010).

Paramétrage des classifieurs :

— LMT et NBT : paramétrage par défaut dans R. *na.action :* gère les données manquantes ; *control = Weka_control() :* passage du Weka à R et *options = Null*.
— KM_{VM} et KM_{SNB} : le seul paramètre utilisateur est le nombre de clusters. Nous nous limitons dans cette étude dans le cas où le nombre de clusters (K) est égale au nombre de classes (voir Section 3.1.4).

3.2.3 Evaluation des performances

De manière à pouvoir comparer les résultats des 4 modèles KM_{VM}, KM_{SNB}, LMT et SNB les même folds en train / test ont été utilisés. Les performances prédictives présentées dans cet article (ci-dessous) sont données en test sur 10×10 folds cross validation "stratifié". Les 100 résultats en test ainsi obtenus permettent le calcul d'un résultat moyen muni de son écart type.

Le logiciel R et le logiciel Khiops n'ayant pas les mêmes façons de calculer les AUCs nous avons recodé ce calcul de manière à produire des valeurs comparables. Nous donnons ci-dessous dans les résultats proposés pour les valeurs d'AUC (aires sous les courbes de ROC (AUC)) l'espérance de l'AUC : $AUC = \sum_i^C P(C_i)AUC(C_i)$, où $AUC(i)$ désigne la valeur d'AUC de classe i contre toutes les autres et $P(C_i)$ désigne le prior sur la classe i (fréquence des éléments de la classe i). Le calcul $AUC(i)$ est réalisé à l'aide du vecteur des probabilités $P(C_i|X)\forall i$ (et non uniquement de la classe prédite). Ceci n'est en aucune façon un biais en faveur de l'une ou l'autre des méthodes.

3.3 Résultats

Le tableau 2 présente les performances prédictives en termes d'accuracy et en termes d'AUC (présenté en %) obtenues par les 4 algorithmes de l'état de l'art (LMT, NBT, SNB, KM_{VM}) et la variante proposée dans cet article (KM_{SNB}).

Résultats en test pour l'accuracy					
Données	KM_{VM}	KM_{SNB}	LMT	NBT	SNB
Glass	89.28 ± 6.62	95.11 ± 5.09	97.48 ± 2.68	94.63 ± 4.39	97.80 ± 3.04
Pima	66.90 ± 4.87	73.72 ± 4.37	76.85 ± 4.70	75.38 ± 4.71	75.41 ± 3.75
Vehicle	47.33 ± 5.91	72.75 ± 4.22	82.52 ± 3.64	70.46 ± 5.17	63.86 ± 4.43
Segment	80.94 ± 1.93	96.18 ± 1.26	96.30 ± 1.15	95.17 ± 1.29	94.44 ± 1.48
Waveform	49.72 ± 3.39	84.04 ± 1.63	86.94 ± 1.69	79.87 ± 2.32	83.14 ± 1.49
Mushroom	98.57 ± 3.60	99.94 ± 0.09	98.06 ± 4.13	95.69 ± 6.73	99.38 ± 0.27
PenDigits	76.82 ± 9.52	97.35 ± 1.36	98.50 ± 0.35	95.29 ± 0.76	89.92 ± 1.33
Adult	77.96 ± 0.41	86.81 ± 0.39	83.22 ± 1.80	79.41 ± 7.34	86.63 ± 0.40
Moyenne	73.44	88.23	89.98	85.73	86.32
Résultats en test pour l'AUC					
Données	KM_{VM}	KM_{SNB}	LMT	NBT	SNB
Glass	96.83 ± 2.67	98.21 ± 2.52	97.94 ± 0.19	98.67 ± 2.05	99.77 ± 0.60
Pima	65.81 ± 6.37	78.44 ± 5.35	83.05 ± 4.61	80.33 ± 5.21	80.59 ± 4.78
Vehicle	74.60 ± 2.89	91.17 ± 1.68	95.77 ± 1.44	88.07 ± 3.04	87.13 ± 1.95
Segment	69.32 ± 3.11	97.21 ± 0.09	99.65 ± 0.23	98.86 ± 0.51	96.52 ± 0.06
Waveform	69.21 ± 3.17	96.16 ± 0.58	97.10 ± 0.53	93.47 ± 1.41	95.81 ± 0.57
Mushroom	98.47 ± 0.38	99.99 ± 0.00	99.89 ± 0.69	99.08 ± 2.29	99.97 ± 0.02
Pendigits	95.84 ± 2.95	99.66 ± 1.03	99.81 ± 0.10	99.22 ± 1.78	99.19 ± 1.14
Adult	59.42 ± 3.70	92.37 ± 0.34	77.32 ± 10.93	84.25 ± 5.66	92.32 ± 0.34
Moyenne	78.68	94.15	93.91	92.74	93.91

TAB. 2 – *Performances en tests sur 10x10 folds cross-validation*

Les résultats montrent que l'algorithme des K-moyennes prédictives suivi par l'insertion de modèles locaux (KM_{SNB}) exhibe des résultats significativement meilleurs que ceux obtenus par le même algorithme utilisant le vote majoritaire (KM_{VM}). Même s'il est difficile de comparer des résultats moyens nous notons néanmoins que pour les 8 bases de données testées le gain moyen est de 20% tant en accuracy qu'en AUC.

Les comparaisons entre modèles ayant des classifieurs naïf de Bayes en modèles locaux montrent que les résultats de KM_{SNB} sont légèrement meilleurs que ceux de NBT. On notera que KM_{SNB} contient ($K = C$) un nombre de modèles locaux très inférieurs à ceux de NBT (voir (Landwehr et al., 2005) pour plus de détails sur la taille des arbres produits par NBT et LMT). Par contre nous notons un avantage de LMT pour les bases testées [1] sauf pour la base Adult qui est la plus peuplée.

Enfin la comparaison entre KM_{SNB} et le modèle global (juge de paix) SNB indique que la méthode proposée donne des résultats très légèrement supérieures notamment pour les bases de données où l'on sait que les variables explicatives sont très corrélées (où l'hypothèse d'indépendance s'affaiblie) comme pour 'PenDigits'.

1. Nous retrouvons dans nos expériences les résultats de (Landwehr et al., 2005) hormis pour Glass où les résultats que nous avons trouvés sont significativement meilleurs. Nous conservons néanmoins les résultats trouvés qui favorisent LMT.

Pour compléter la comparaison nous donnons ci-dessous dans le tableau 3 quelques éléments supplémentaires de comparaison.

Classifieur	LMT	NBT	SNB	KM_{VM}	KM_{SNB}
Modèle hybride / global	hybride	hybride	global	global	hybride
Gestion des valeurs manquantes	non	oui (arbre) / non (NB)	oui	oui	oui
Codage variables catégorielles pour les modèles locaux	codage disjonctif complet	codage disjonctif complet	-	- -	groupage supervisé
Cross validation requise pour les modèles locaux	oui	oui	-	-	non
Méthode de sélection variables dans les modèles locaux	non	non	-	-	oui

TAB. 3 – *Elements de comparaison d'après (Landwehr et al., 2005; Kohavi, 1996; Boullé, 2007a; Alaoui Ismaili, 2016)*

Nous y observons que la méthode proposée est assez bien placée car elle ne nécessite pas de cross validation, gère les valeurs manquantes nativement, opère une sélection de variables tant dans l'étape de clustering que lors de la construction des modèles locaux. Enfin, elle réalise un groupage supervisé des valeurs de variables catégorielles évitant ainsi de passer par un codage disjonctif complet qui entraine la création d'un vecteur d'entrée souvent grand compliquant ensuite l'interprétation du modèle obtenu.

D'autres axes de comparaison existent comme la complexité algorithmique, la robustesse, le nombre de modèles locaux produits... Mais la place nous manquant nous ne pouvons les aborder en détails. Nous mentionnerons juste le fait que le modèle KM_{SNB} est très compétitif (notamment en présence de variables catégorielles ayant beaucoup de modalités) sur ces points.

Synthèse : l'introduction des modèles locaux dans KM_{VM} pour obtenir KM_{SNB} répond aux objectifs que nous nous étions fixés tant en termes de performance que dans la facilité de mise en œuvre de la méthode. Nous estimons de plus que ce gain en performance n'a pas détruit le caractère interprétable des K-Moyennes prédictives. Même si le but principal de cet article n'est pas de discuter du pouvoir interprétable des K-moyennes prédictives (mais du pouvoir prédictif) nous essayons d'illustrer ce point au cours de la section suivante (dans la limite des considérations de place).

4 Méthodologie d'analyse des résultats

Pour avoir une idée sur la capacité de notre algorithme des K-moyennes prédictives à fournir à la fois des résultats performants (en termes de prédiction) et faciles à interpréter, la base de donnée Vehicle est utilisée comme un exemple illustratif. Cette base de données est constituée de 846 exemples, 18 variables descriptives et une variable à prédire contenant 4 classes (bus, opel, saab, van). Dans cette étude illustrative, nous utilisons l'ensemble des données pour apprendre le modèle avec K égal au nombre de classes $C = 4$. Nous ne détaillons pas ici la signification des variables A1 à A18 mais le lecteur pourra les trouver sur le site des bases de l'UCI (Lichman, 2013). Pour des raisons de place nous nous limitons ici à n'utiliser que les 6 variables les plus informatives dans le clustering initial sans que cela ne change la méthodologie d'analyse.

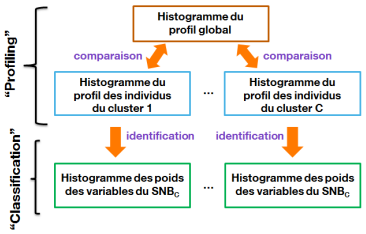

FIG. 3 – *Analyse à deux niveaux.*

L'interprétation d'un KM_{SNB} peut être réalisée simplement à l'aide d'une méthodologie à deux niveaux (voir figure 3). En première analyse, on s'intéresse au profil des clusters en présentant au sein d'une même figure (voir Figure 4), à l'aide d'histogrammes, le profil moyen de la population globale (chaque bâton représente le pourcentage d'individus possédant une valeur dans l'intervalle considéré ; intervalles issus de la phase de prétraitement) ainsi que le profil moyen des individus de chaque cluster. Cette visualisation permet par différenciation de comprendre pourquoi les individus ont été regroupés. A titre d'exemple on s'aperçoit que la variable A12 est très discriminante pour les individus du cluster 4 pour lesquels 100% d'entre eux sont A12<296.5.

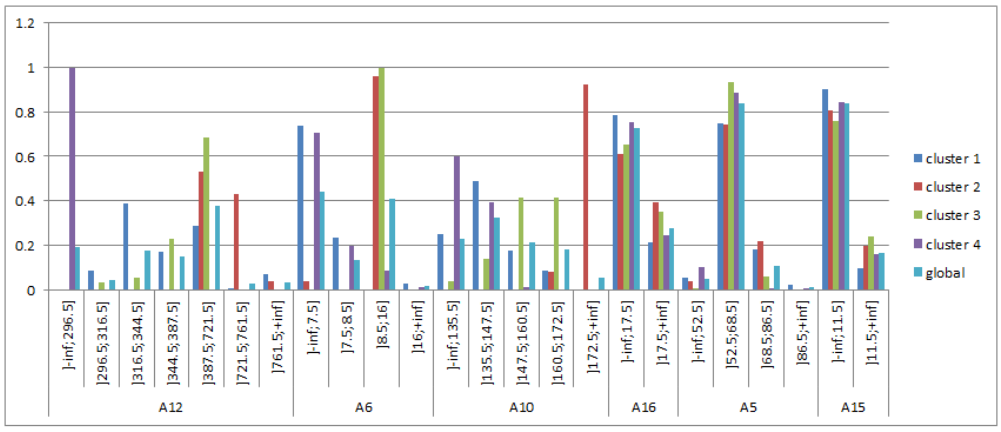

FIG. 4 – *Histogramme : profils moyens des individus en global et pour chaque clusters*

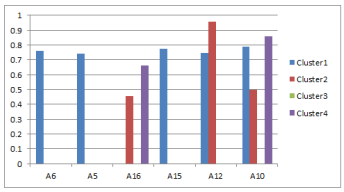

FIG. 5 – *Histogramme : poids des variables au sein des modèles locaux.*

Enfin en deuxième niveau on fournit un histogramme par cluster (voir Figure 5) qui donne le poids des variables des SNB locaux permettant de connaitre le pourquoi de la classification locale. Par exemple, on s'aperçoit que le cluster 3 n'a que des poids à 0 indiquant que le classifieur majoritaire a été conservé. Que pour le cluster 1 les variables ont des poids très proches tandis que pour le cluster 2 la variable A12 est la plus importante... Ce deuxième niveau d'analyse contient des éléments d'interprétation qui ne sont pas disponibles dans KM_{VM}.

L'un des avantages d'incorporer les modèles locaux dans notre algorithme des K-moyennes prédictives est d'améliorer leur pouvoir descriptif (interprétation des résultats). En effet, grâce aux modèles locaux, on est capable non seulement de connaître les variables les plus discriminantes dans

le modèle global mais également de connaître celles qui contribuent le plus à la construction de chaque groupe dans la phase d'apprentissage. Par conséquent, à l'arrivée d'une nouvelle instance, on est capable de connaître facilement les différentes raisons qui déterminent la prédiction de sa classe.

5 Conclusion et perspectives

L'introduction des modèles locaux dans KM_{VM} pour obtenir KM_{SNB} répond aux objectifs que nous nous étions fixés tant en termes de **performances**, que dans la facilité de mise en œuvre de la méthode tout en conservant l'aspect **interprétable** du modèle global hybride obtenu. Les résultats expérimentaux se placent très honorablement dans l'état de l'art sur ces deux aspects, et ce malgré le fait de fixer le nombre de clusters comme étant égal au nombre de classes. Dans de futurs travaux nous étudierons la possibilité de trouver automatiquement le bon nombre de clusters par exemple à l'aide d'un clustering hiérarchique descendant plaçant alors la méthode à la manière de LMT mais dans le champ du clustering prédictif. Nous espérons alors un nouveau gain en termes de performances. Enfin nous pensons développer un outil de visualisation des résultats permettant de naviguer dans les clusters afin d'observer aisément les profils moyens et les importances des variables localement aux clusters.

Références

Alaoui Ismaili, O. (2016). *Clustering prédictif - Décrire et Prédire simultanément*. Ph. D. thesis, University Paris Saclay - Agro Paris Tech.

Al-Harbi, S. H. et V. J. Rayward-Smith (2006). Adapting k-means for supervised clustering. *Applied Intelligence 24*(3), 219–226.

Arthur, D. et S. Vassilvitskii (2007). K-means++ : The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035.

Basu, S., A. Banerjee, et R. J. Mooney (2002). Semi-supervised clustering by seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 27–34.

Bondu, A., M. Boullé, et D. Gay (2013). Data grid models. Slides for the tutorial given at EGC 2013, Toulouse, France. http ://www.marc-boulle.fr/publications/TutorialEGC13.pdf.

Bouchard, G. et B. Triggs (2004). The tradeoff between generative and discriminative classifiers. In *IASC International Symposium on Computational Statistics (COMPSTAT)*, pp. 721–728.

Boullé, M. (2007a). Compression-based averaging of selective naive Bayes classifiers. *Journal of Machine Learning Research 8*, 1659–1685.

Boullé, M. (2007b). *Recherche d'une repr ?sentation des données efficace pour la fouille des grandes bases de donn ?es*. Ph. D. thesis, ENST.

Boullé, M. (2016). Khiops : outil d'apprentissage supervisé automatique pour la fouille de grandes bases de données multi-tables. In *16éme Journées Francophones Extraction et Gestion des Connaissances, EGC*, pp. 505–510.

Celebi, M. E., H. A. Kingravi, et P. A. Vela (2013). A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Syst. Appl. 40*(1), 200–210.

Eick, C. F., N. Zeidat, et Z. Zhao (2004). Supervised clustering - algorithms and benefits. In *International Conference on Tools with Artificial Intelligence*, pp. 774–776.

Guyon, I. et A. Elisseeff (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res. 3*, 1157–1182.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, et I. H. Witten (2009). The weka data mining software : An update. *SIGKDD Explor. Newsl. 11*(1), 10–18.

Hand, D. J. et K. Yu (2001). Idiot's bayes-not so stupid after all ? *International Statistical Review 69*(3), 385–398.

Hornik, K. (2017). R FAQ.

Ismaili, O. A., V. Lemaire, et A. Cornuejols (2016). *Supervised pre-processings are useful for supervised clustering*, pp. 147–157. Springer International Publishing.

Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers : a decision-tree hybrid. In *International Conference on Data Mining*, pp. 202–207. AAAI Press.

Landwehr, N., M. Hall, et E. Frank (2005). Logistic model trees. *Mach. Learn. 59*(1-2).

Langley, P. et S. Sage (1994). Induction of selective bayesian classifiers. In *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, USA, pp. 399–406. Morgan Kaufmann Publishers Inc.

Lemaire, V., O. Alaoui Ismaili, et A. Cornuejols (2015). An initialization scheme for supervized k-means. In *International Joint Conference on Neural Networks*, pp. 1–8.

Lemaire, V., C. Hue, et O. Bernier (2009). Correlation explorations in a classification model. In *Workshop Data Mining Case Studies and Practice Prize, KDD 2009*.

Lichman, M. (2013). UCI machine learning repository.

MATLAB (2010). *version 7.10.0 (R2010a)*. Natick, Massachusetts : The MathWorks Inc.

Milligan, G. W. et M. C. Cooper (1988). A study of standardization of variables in cluster analysis. *Journal of Classification 5*(2), 181–204.

Salperwyck, C. et V. Lemaire (2011). Learning with few examples : An empirical study on leading classifiers. In *International Joint Conference on Neural Networks*, pp. 1010–1019.

Vilalta, R., M.-K. Achari, et C. F. Eick (2003). Class decomposition via clustering : A new framework for low-variance classifiers. In *International conference on Data Mining (ICDM)*.

Summary

The majority vote is the commonly method used in the predictive clustering context for assigning the class to the resulting clusters in the training phase. However, this method has limits which could influence the results quality obtained. To overcome these problems, we proposed to incorporate a local model inside every cluster to improve the predictive performance of global model. Experimental results show that the incorporation of local models allow us to obtain better results than those obtained using the majority vote; this keeping the nice descriptive aspect of the global model.