Évaluation en cascade d'algorithmes de clustering

L. Candillier[1,2], I. Tellier[1], F. Torre[1], O. Bousquet[2]

[1] GRAppA, Université Charles de Gaulle, Lille 3
candillier@grappa.univ-lille3.fr
[2] Pertinence, 32 rue des Jeûneurs, 75002 Paris
olivier.bousquet@pertinence.com

Résumé :
Cet article se place dans le cadre de l'évaluation des résultats d'algorithmes de clustering et de la comparaison de tels algorithmes. Nous proposons une nouvelle méthode basée sur l'enrichissement d'un ensemble de jeux de données étiquetés indépendants par les résultats des algorithmes de clustering considérés, et sur l'utilisation d'un algorithme supervisé pour évaluer l'intérêt de ces nouvelles informations apportées.

Nous adaptons ainsi la technique de *cascade generalization* (Gama & Brazdil, 2000) au cas où l'on combine un apprenant supervisé et un apprenant non supervisé. Nous considérons également le cas où des apprentissages supervisés indépendants sont exécutés sur les différents groupes de données identifiés par le clustering (Apte *et al.*, 2002).

Nous avons mené des expérimentations en considérant différents algorithmes supervisés pour comparer plusieurs algorithmes de clustering. Nous montrons ainsi le comportement cohérent de la méthode proposée qui met en avant, par exemple, le fait que les algorithmes de clustering basés sur l'utilisation de modèles probabilistes plus *complexes* surpassent les algorithmes basés sur des modèles plus *simples*.

1 Introduction

En apprentissage supervisé comme en apprentissage non supervisé, l'évaluation des résultats d'une méthode donnée, de même que la comparaison de différentes méthodes, est une problématique importante. Mais si la *validation croisée* est une méthodologie reconnue pour l'évaluation en classification supervisée, l'évaluation de la pertinence des groupes formés en classification non supervisée reste un problème ouvert. La difficulté vient principalement du fait que l'évaluation des résultats d'algorithmes de clustering est subjective par nature car il existe souvent différents regroupements pertinents possibles pour un même jeu de données.

En pratique, il existe quatre méthodes principales pour mesurer la qualité des résultats d'algorithmes de clustering, mais chacune souffre de différentes limitations.

1. Utiliser des données artificielles pour lesquelles le regroupement attendu est connu. Mais les méthodes sont alors évaluées uniquement sur les distributions spéci-

fiques correspondantes, et les résultats sur données artificielles ne peuvent pas être généralisés aux données réelles.

2. Utiliser des jeux de données étiquetés et observer si la méthode retrouve les classes initiales. Mais les classes d'un problème supervisé ne correspondent pas nécessairement aux groupes qu'il est pertinent d'identifier pour une méthode non supervisée, d'autres regroupements pouvant être également pertinents.
3. Utiliser des critères numériques, tels l'inertie intra-cluster et/ou la séparation inter-clusters. Mais de tels critères ont une part importante de subjectivité car ils utilisent des notions prédéfinies de ce qu'est un bon clustering. Par exemple, la séparation des clusters n'est pas toujours un bon critère à utiliser car parfois, des clusters qui se chevauchent peuvent être plus pertinents.
4. Utiliser un expert pour évaluer le sens d'un clustering donné dans un champ d'application spécifique. Mais s'il est possible à un expert de dire si un regroupement donné a du sens, il est beaucoup plus problématique de quantifier son intérêt ou de dire si un regroupement est meilleur qu'un autre. De plus, l'intérêt de la méthode ne peut être généralisé à différents types de jeux de données.

Le risque principal dans l'évaluation d'une méthode de clustering est de considérer celle-ci comme un but en soi. En fait, ce que l'on cherche à évaluer est la capacité d'une méthode de clustering à identifier de l'information nouvelle, intéressante et utile, c'est-à-dire une connaissance nouvelle qu'il est intéressant d'utiliser dans un certain cadre. Nous attendons également que la méthode soit capable d'identifier de telles informations intéressantes sur différents types de problèmes. L'idée principale de notre approche est donc de considérer le clustering comme un prétraitement à une autre tâche que nous sommes capables d'évaluer : l'apprentissage supervisé par exemple. Ainsi, le biais dépendant de la tâche choisie nous empêche d'ordonner les méthodes de clustering indépendemment de toute tâche, mais ce biais est moins important que lorsqu'une concordance directe entre les groupes obtenus et les classes initiales est évaluée. De plus, il est ainsi possible d'évaluer la contribution du clustering dans l'achèvement de cette tâche objective et réelle.

Nous proposons donc dans ce papier une nouvelle méthode d'évaluation qui consiste à comparer les résultats d'un algorithme supervisé lorsqu'il est (ou pas) aidé par de l'information issue d'un algorithme de clustering. Ainsi, si les résultats de l'algorithme supervisé sont améliorés lorsqu'il utilise de l'information issue d'un algorithme de clustering, alors nous postulons que cela signifie que cette information est nouvelle et utile. Cette méthode nous permet donc d'évaluer objectivement l'intérêt de l'information apportée par l'algorithme de clustering. De plus, la diminution de l'erreur de l'algorithme supervisé aidé par l'information issue des résultats du clustering nous permet également de quantifier cet intérêt.

Notre méthode se place donc dans le cadre de la combinaison de classifieurs, celle d'une méthode supervisée et d'une méthode non supervisée dans notre cas. Plusieurs façons de combiner différents classifieurs par des votes peuvent être trouvées dans (Ali & Pazzani, 1996), les deux méthodes de vote les plus reconnues étant le *bagging* (Breiman, 1996a) et le *boosting* (Freund & Schapire, 1996). Quelques généralisations théoriques de ces techniques ont également été étudiées, menant aux *arcing classifiers* (Breiman, 1996b), aux *méthodes d'ensemble* (Dietterich, 2000) et aux *leveraging methods* (Meir & Rätsch, 2003).

Nous nous focalisons ici sur les techniques qui utilisent différents apprenants de manière séquentielle. Dans de telles méthodes, la sortie d'un apprenant est un enrichissement de la description des exemples qui est ensuite utilisé par l'apprenant suivant. Dans ce cadre, la *stacked generalization* (Wolpert, 1992) est une méthode très générale dans laquelle différents traitements sont enchaînés : chaque traitement modifie la représentation des exemples, et le nouveau jeu de données enrichi est utilisé par le niveau suivant. La *cascade generalization* (Gama & Brazdil, 2000) est un cas particulier de stacked generalization : à chaque niveau, un classifieur est appliqué sur chaque exemple $\vec{x}$, fournissant la probabilité $p(c|\vec{x})$ que $\vec{x}$ appartienne à la classe c ; ces probabilités sont ensuite ajoutées à la description des exemples et utilisées par le classifieur suivant. La cascade generalization permet de combiner plusieurs classifieurs mais nécessite que chacun soit capable de fournir une probabilité de distribution sur les exemples. En pratique, seulement deux classifieurs sont utilisés.

Enfin, nous considérons également le cas où un apprenant supervisé et un apprenant non supervisé sont combinés comme proposé dans (Apte *et al.*, 2002). Dans ce cas, plusieurs clusterings sont exécutés en faisant varier les paramètres d'entrée de la méthode (par exemple le nombre de groupes recherchés), menant ainsi à différentes partitions de l'ensemble des objets. Pour chaque partition, plusieurs apprentissages supervisés sont menés indépendemment sur les différents groupes d'objets formés. Et finalement, la partition ayant mené au taux d'erreur le plus faible est conservée.

Basé sur ce principe d'enchaîner en cascade un apprenant non supervisé et un apprenant supervisé, puis de calculer la diminution du taux d'erreur de l'apprenant supervisé lorsqu'il est aidé par l'information issue de l'apprenant non supervisé, la nouvelle méthode d'évaluation d'algorithmes de clustering que nous proposons s'appelle *évaluation en cascade*. Nous détaillons tout d'abord cette méthode dans la section 2. Puis nous présentons dans la section 3 les expérimentations que nous avons menées avec cette méthode. Enfin, nous concluons ce papier dans la section 4, et proposons des pistes pour la poursuite de cette recherche.

2 Évaluation en cascade

Étant donné un jeu de données initial contenant des informations de classes, les étapes principales de la méthode que nous proposons sont les suivantes :

1. apprentissage 1
   – apprentissage supervisé sur le jeu de données initial
2. apprentissage 2
   – clustering sur le jeu de données en ignorant les informations de classes
   – enrichissement du jeu de données à partir des résultats du clustering
   – apprentissage supervisé sur le jeu de données enrichi
3. comparaison des résultats des 2 classifieurs appris

Comme nous l'avons évoqué précédemment, nous envisageons deux façons d'enrichir les jeux de données à partir des résultats du clustering. La première consiste à créer de nouveaux attributs représentant l'information capturée par le clustering, et à ajouter ces nouveaux attributs dans le jeu de données initial. La seconde est de créer des sous-jeux de données en fonction des groupes ciblés par le clustering.

Concernant les nouveaux attributs créés depuis les résultats du clustering dans le cas de la première méthode, différents types d'informations peuvent être ajoutés.

1. Étant donné que la plupart des algorithmes de clustering fournissent en sortie une partition de l'ensemble des objets, nous pouvons utiliser comme nouveaux attributs l'appartenance des objets aux clusters. Cette information serait ainsi représentée par un nouvel attribut catégoriel, chaque objet étant associé à un identifiant du cluster auquel il appartient.
2. On peut également associer à chaque objet un ensemble d'attributs représentant le centre du cluster auquel il est associé. Le nombre d'attributs du jeu de données serait ansi doublé.
3. Récemment ont émergé de nombreux algorithmes de *subspace clustering* (Parsons *et al.*, 2004) capables d'associer à chaque attribut de chaque cluster un poids spécifiant l'importance de l'attribut pour déterminer l'appartenance des objets au cluster. Dans ces cas, nous pouvons donc associer à chaque objet un nouvel attribut numérique par attribut initial, correspondant au poids de cet attribut pour le cluster auquel l'objet appartient. Un tel nouvel attribut permettrait de différencier les objets pour lesquels un attribut donné est pertinent des autres objets pour lesquels cet attribut n'est pas pertinent (selon les résultats du subspace clustering).
4. La probabilité d'appartenance de chaque objet à chaque cluster peut également constituer une nouvelle information à ajouter lorsque des modèles probabilistes sont utilisés. Si ce n'est pas le cas, alors la distance de chaque objet à chaque cluster peut être calculée puis ajoutée à leur description.
5. Les valeurs minimum et maximum des coordonnées des membres de chaque cluster sur chaque attribut peuvent aussi être utilisées pour enrichir les jeux de données. À chaque objet serait alors associées ces informations en fonction du cluster auquel il appartient.

De plus, comme la plupart des algorithmes de clustering nécessitent de spécifier différents paramètres en entrée, nous pouvons exécuter ces algorithmes pour différentes valeurs de paramètres, et enrichir les jeux de données pour chaque résultat de clustering. Par exemple, de nombreuses méthodes de clustering nécessitent de spécifier le nombre de clusters recherchés. Dans ce cas, nous pouvons exécuter la méthode plusieurs fois en faisant varier le nombre de clusters entre 2 et 10. L'algorithme supervisé utilisé ensuite pourrait alors choisir quel(s) attribut(s) utiliser parmi tous ceux générés.

Dans le cas de la seconde méthode de combinaison proposée, nous générons dans un premier temps plusieurs partitions avec différents paramètres d'entrée. Puis nous calculons les erreurs en validation croisée d'apprenants supervisés exécutés indépendamment sur chaque groupe d'objet créé par le clustering. Et finalement, la partition ayant mené au taux d'erreur le plus faible est conservée.

La figure 1 résume les étapes principales de cette méthodologie :

1. diviser le jeu de données en deux parts égales : un ensemble d'apprentissage Ap^{-} et un ensemble de test Te^{-}

2. exécuter plusieurs fois le clustering, avec différents paramètres d'entrée et en ignorant les informations de classes, et produire les ensembles enrichis d'apprentissage Ap^+ et de test Te^+
3. lancer l'apprentissage supervisé sur l'ensemble d'apprentissage Ap^- et produire le classifieur C^-
4. lancer l'apprentissage supervisé sur l'ensemble d'apprentissage enrichi Ap^+ et produire le classifieur C^+
5. lancer la classification sur l'ensemble test Te^- à partir du classifieur C^-
6. lancer la classification sur l'ensemble test Te^+ à partir du classifieur C^+
7. et comparer les erreurs de classification

Pour évaluer l'amélioration des résultats avec ou sans les nouvelles informations fournies par un algorithme de clustering évalué, nous testons les deux méthodes (l'apprentissage supervisé sur les données initiales et l'apprentissage supervisé sur les données enrichies) sur différents jeux de données indépendants. Sur chaque jeu de données, nous faisons cinq validations croisées avec découpage du jeu de données en deux, comme proposé dans (Dietterich, 1998). Pour chaque validation croisée, nous calculons les taux d'erreur pondérés des deux méthodes. Puis nous utilisons quatre mesures pour comparer les résultats :

1. *nb vict* : le nombre de victoires de chaque méthode
2. *vict sign* : le nombre de victoires significatives, en utilisant le *5×2cv F-test* (Alpaydin, 1999) pour vérifier si les résultats sont significativement différents
3. *wilcoxon* : le *wilcoxon signed rank test*, qui indique si une méthode est significativement meilleure qu'une autre sur un ensemble de problèmes indépendants
4. et *moyenne* : l'erreur pondérée moyenne

Pour effectuer ces comparaisons, C4.5 (Quinlan, 1993) comme apprenant supervisé est bien adapté car, comme il fait de la sélection d'attributs pendant l'apprentissage et fournit en sortie un arbre de décision où l'on peut observer quels attributs ont été utilisés et à quels niveaux ils ont été utilisés, il met clairement en avant si les nouvelles informations issues du clustering sont utiles ou non. De plus, il a également l'avantage d'être rapide et d'être capable de considérer des attributs catégoriels aussi bien que numériques. Enfin, afin de vérifier si les résultats sont dépendants de l'algorithme supervisé utilisé, nous mènerons également des expérimentations avec deux autres algorithmes supervisés : C5 boosté (Quinlan, 2004) et DLG (Webb & Agar, 1992).

3 Expérimentations

Les expérimentations conduites dans cette section abordent deux problèmes complémentaires :

1. comment un algorithme supervisé peut-il bénéficier des informations issues de méthodes de clustering ?

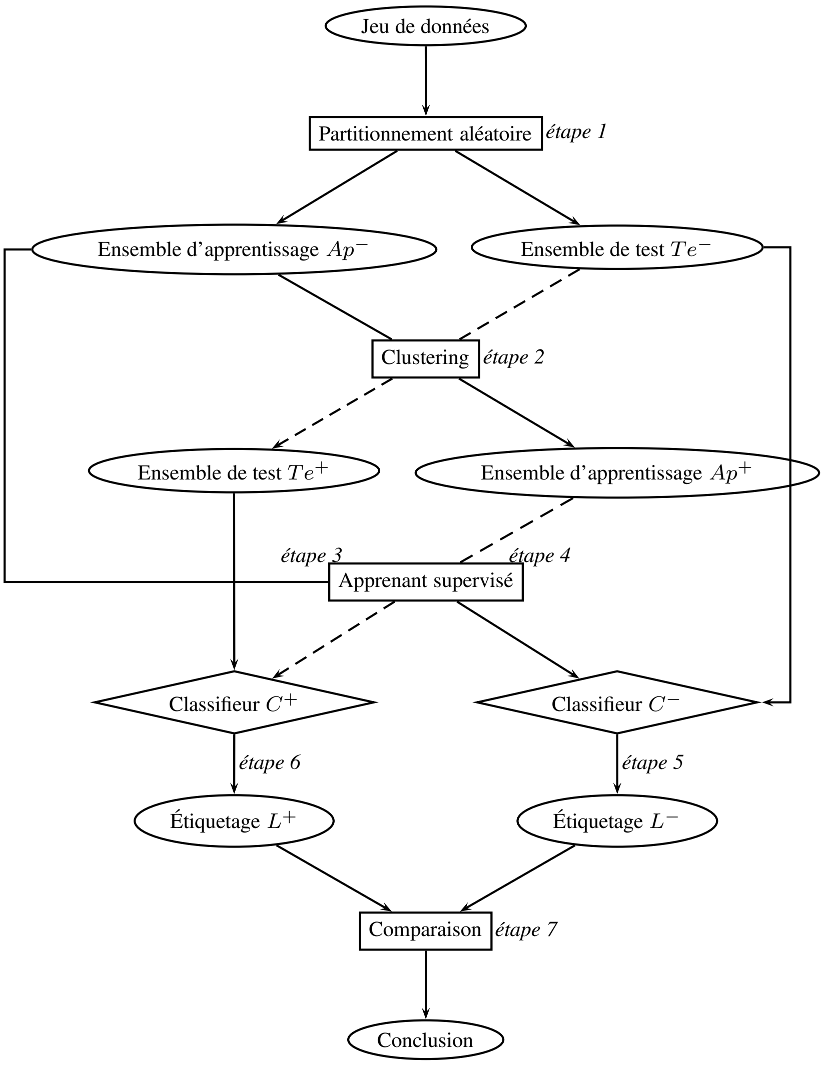

FIG. 1 – Méthodologie d'évaluation de l'intérêt d'ajouter à un jeu de données de l'information provenant d'un algorithme de clustering.

2. quelle méthode de clustering fournit les informations les plus utiles aux algorithmes supervisés ?

Nous présentons d'abord une application où nous montrons l'intérêt d'utiliser le clustering en prétraitement d'un apprentissage supervisé. Puis nous présentons les résultats des comparaisons de plusieurs algorithmes de clustering :

- Aléa, un algorithme qui génère des partitions aléatoires, prenant en entrée le nombre de clusters recherchés, et servant de référence
- l'algorithme très connu K-means, basé sur l'évolution de K centres mobiles représentant les K clusters à trouver
- LAC (Domeniconi *et al.*, 2004), une adaptation de K-means qui associe à chaque centre de cluster un poids sur chaque dimension, inversement proportionnel à la dispersion des objets du cluster sur la dimension
- EM1, basé sur l'utilisation de modèles probabilistes et de l'algorithme EM sous l'hypothèse que les données ont été générées par des distributions gaussiennes indépendantes sur chaque dimension (Ye & Spetsakis, 2003)
- et EM2, une adaptation de EM1 qui effectue de la sélection d'attributs pendant l'apprentissage, en ne considérant pour chaque cluster qu'un sous-ensemble des dimensions sur lesquelles la déviation standard est minimisée

Les algorithmes comparés ici utilisent donc des modèles ayant des niveaux de complexité différents. En effet, K-means utilise uniquement un centroïde pour représenter un cluster. LAC ajoute à chaque centroïde un vecteur de poids sur chaque dimension de l'espace de description. EM1 définit une probabilité d'appartenance de chaque objet à chaque cluster et utilise un modèle gaussien. Et EM2 considère également un sous-ensemble des dimensions pertinentes associées à chaque cluster.

Tous ces algorithmes nécessitent de spécifier le nombre K de clusters recherchés. Comme nous en avons discuté précédemment, nous exécuterons donc plusieurs fois ces méthodes en faisant varier K entre 2 et 10.

Parmi les cinq ensembles d'attributs qui peuvent être ajoutés aux jeux de données initiaux par la première méthode de combinaison proposée, seuls les trois premiers sont utilisés dans ces expérimentations, parce que le quatrième ensemble dégrade les résultats et que le cinquième ne semble pas ajouter d'information supplémentaire. Or en n'utilisant pas ces deux ensembles d'attributs, nous réduisons la taille des jeux de données enrichis, ce qui permet de fournir aux apprenants supervisés des données moins complexes.

Enfin, les jeux de données utilisés sont ceux issus de l'UCI Machine Learning Repository (Blake & Merz, 1998) qui ne contiennent que des données numériques.

3.1 Exemple sur la base *wine*

Le jeu de données *wine* issue de l'UCI Machine Learning Repository contient la description de 178 vins définis par 13 attributs. 3 types de vins, qui représentent les classes, sont présents.

Lorsque C4.5 est appliqué sur ce jeu de données, l'arbre de décision obtenu est celui présenté figure 2. On observe alors que les attributs sélectionnés par C4.5 sont *ColorIntensity*, *Flavanoids* et *Proline*.

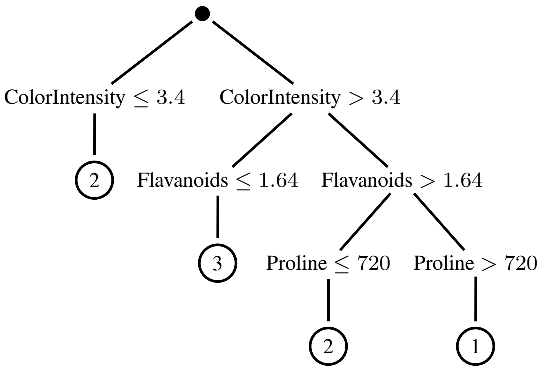

FIG. 2 – Arbre de décision appris par C4.5 sur wine.

Lorsque l'on utilise EM2 pour ajouter de l'information au jeu de données tel que décrit dans la section 2, nous obtenons l'arbre de décision présenté figure 3. On observe alors que C4.5 a utilisé l'un des attributs créés par notre méthode : celui qui se réferre à l'appartenance des objets aux clusters lorsque le nombre de clusters est positioné à 3. L'autre attribut utilisé est *ColorIntensity*, également utilisé par C4.5 sur le jeu de données initial.

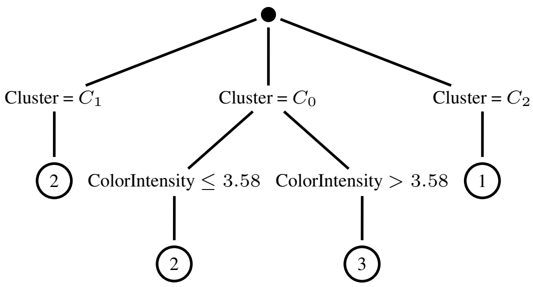

FIG. 3 – Arbre de décision appris par C4.5 après ajout d'information par EM2 sur wine.

Si l'on reprend la méthode de visualisation graphique des règles associées aux clusters présentée dans (Candillier *et al.*, 2005), on s'aperçoit que la meilleure projection 2D sélectionnée concerne les attributs *Flavanoids* et *Proline*, les deux autres attributs utilisés dans l'arbre de décision appris par C4.5 initialement. La figure 4 montre cette projection.

Finalement, nous observons donc sur cet exemple une corrélation forte entre l'arbre de décision appris par C4.5 sur le jeu de données wine initial, et l'arbre de décision appris par C4.5 sur le jeu de données wine enrichi par les résultats du clustering appris par EM2. De plus, nous pouvons remarquer que l'attribut ajouté par notre méthode est utilisé par C4.5 dès le début de la construction de l'arbre de décision. Nous sommes donc là en présence d'un exemple où C4.5 a utilisé l'information provenant des résultats du

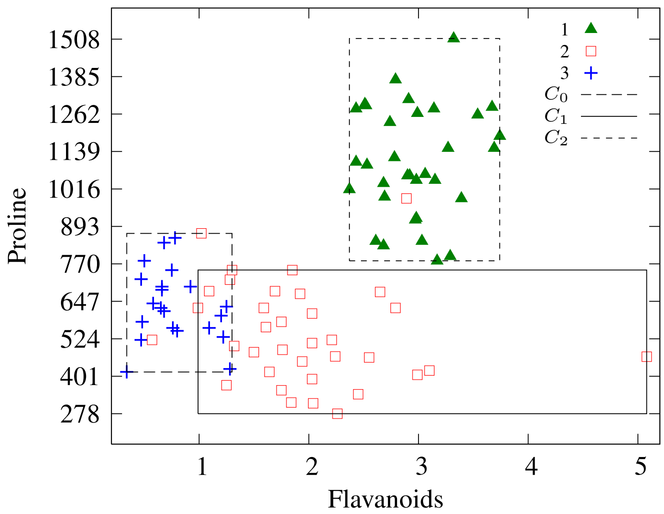

FIG. 4 – Meilleure projection 2D de EM2 sur wine.

clustering à un haut niveau dans l'arbre de décision qu'il construit. Il semble donc qu'il spécialise ici son traitement selon les différentes zones créées par EM2 dans l'espace de description des objets. Une autre interprétation possible est que notre nouvel attribut créé aide C4.5 à définir des zones de décision plus complexes. En effet, utiliser le nouvel attribut ajouté par EM2 correspond en fait à effectuer un test sur plusieurs attributs simultanément, alors que C4.5 seul ne peut utiliser à un nœud de l'arbre qu'un attribut à la fois.

Ensuite, lorsque l'on calcule l'erreur de classification des deux arbres sur les données de test, l'arbre initial a un taux d'erreur de 5/89 alors que l'arbre construit avec l'aide de EM2 a une erreur de seulement 3/89. Les erreurs de l'arbre initial correspondent à 1 erreur où un élément de la classe 1 a été associé à la classe 2, et 4 erreurs où des éléments de la classe 2 ont été associés à la classe 3. D'autre part, les erreurs associées au second arbre correspondent à 1 erreur où un élément de la classe 1 a été associé à la classe 2, et 2 erreurs où des éléments de la classe 2 ont été associés à la classe 3. La méthode a donc aidé à différencier les classes 2 et 3. Ceci semble donc confirmer notre intuition que les nouveaux attributs ajoutent au jeu de données des informations de plus haut niveau.

3.2 Comparaison d'algorithmes de clustering

La table 1 présente les taux d'erreurs pondérés de C4.5 sur les jeux de données initiaux de l'UCI, puis sur les jeux de données enrichis par les algorithmes de clustering correspondants. Chaque mesure correspond à une moyenne sur cinq validations croisées avec découpage du jeu de données en deux. À chaque instant, toutes les méthodes sont exécutées sur le même ensemble d'apprentissage et évaluées sur le même ensemble de test.

	C4.5 seul	C4.5 + Aléa	C4.5 + K-means	C4.5 + LAC	C4.5 + EM1	C4.5 + EM2
ecoli	48.5	48.3	42.8	**40.3**	42	43.1
glass	**32.6**	40.8	35.7	37	40.4	34.9
image	4.8	6	4.8	**4.6**	**4.6**	**4.6**
iono	14.1	15.8	14.2	13.1	**9.8**	11.2
iris	7.3	7.9	6.7	**3.7**	5.1	4.7
pima	31	35	32.1	32.1	30.8	**30**
sonar	31	35.2	30	28.8	28.8	**27.2**
vowel	29.5	38.5	25	26.4	24.1	**22.2**
wdbc	5.9	6.8	4.6	3.9	5.1	**3.1**
wine	8.7	8.8	10.4	9.6	**2.7**	3.6

TAB. 1 – Taux d'erreur pondéré (en %) de C4.5 seul et C4.5 enrichi par les algorithmes de clustering. Les valeurs en gras correspondent au taux d'erreur minimum obtenu sur chaque jeu de données.

On peut ainsi observer que la plupart du temps, les résultats de C4.5 sont améliorés lorsque des informations sont ajoutées à partir des résultats d'algorithmes de clustering *réels* alors qu'ils se dégradent lorsque le clustering aléatoire est utilisé. De plus, les résultats obtenus à l'aide de EM1 et de EM2 sont souvent meilleurs que les résultats obtenus avec K-means et LAC.

Nous utilisons ensuite le *5×2cv F-test* (Alpaydin, 1999) pour évaluer si les résultats de deux algorithmes supervisés (l'un sur le jeu de données initial et l'autre sur le jeu de données enrichi) sont significativement différents. La table 2 reporte ces tests entre C4.5 seul et C4.5 enrichi par les algorithmes de clustering correspondants. Elle utilise la p-value associée au *5×2cv-F test*. Mais la mesure reportée est 1 moins la p-value. Ainsi, une valeur de 0.01 pour EM2 sur le jeu de données wine signifie que la probabilité que C4.5 aidé par EM2 surpasse C4.5 seul *par hasard* est de seulement 1%.

Enfin, la table 3 présente un résumé des comparaisons entre C4.5 seul et C4.5 enrichi par les algorithmes de clustering correspondants. Rappelons les mesures utilisées :

- *nb vict* est le nombre de victoires de chaque méthode
- *vict sign* est le nombre de victoires significatives : le nombre de fois où l'inégalité $(1 - pvalue \leq 0.05)$ est vérifiée
- *wilcoxon* est le *wilcoxon signed rank test* : s'il est supérieur à 1.96, alors cela signifie que la méthode est significativement meilleure que C4.5 seul
- et *moyenne* est l'erreur pondérée moyenne (en %)

EM2 est le seul algorithme qui améliore significativement les résultats de C4.5 sur l'ensemble des jeux de données, selon le test de wilcoxon. Il est significativement meilleur sur 3 jeux de données selon le *5×2cv-F test*. Mais comme EM2, EM1 améliore les résultats de C4.5 neuf fois sur dix, contrairement à K-means et LAC. Tous les algorithmes améliorent les résultats de C4.5 en moyenne, sauf le clustering aléatoire. Et lorsque C4.5 est combiné avec des algorithmes de clustering basés sur des modèles

	C4.5 + Aléa	C4.5 + K-means	C4.5 + LAC	C4.5 + EM1	C4.5 + EM2
ecoli	0.76	0.43	0.20	0.40	0.20
glass	0.12	0.33	0.57	0.24	0.33
image	0.24	0.61	0.61	0.54	0.67
iono	0.67	0.32	0.62	**0.02**	0.09
iris	0.53	0.81	0.65	0.43	0.22
pima	0.19	0.43	0.50	0.53	0.27
sonar	0.37	0.57	0.39	0.33	0.09
vowel	0.01	0.33	0.44	0.23	**0.03**
wdbc	0.46	0.25	**0.04**	0.63	**0.02**
wine	0.53	0.55	0.60	**0.01**	**0.01**

TAB. 2 – 1 - pvalue associée au 5×2cv F-test entre C4.5 seul et C4.5 enrichi par les algorithmes de clustering. Les valeurs en gras correspondent aux différences considérées comme très significatives (inférieures à 0.05).

	C4.5 seul	C4.5 + Aléa	C4.5 + K-means	C4.5 + LAC	C4.5 + EM1	C4.5 + EM2
nb vict	-	1/9	5/4	7/3	**9/1**	**9/1**
vict sign	-	0/1	0/0	1/0	2/0	**3/0**
wilcoxon	-	-2.67	-0.05	1.31	1.83	**2.56**
moyenne	21.3	24.3	20.6	20	19.3	**18.5**

TAB. 3 – Comparaison de C4.5 seul avec C4.5 enrichi par les algorithmes de clustering.

plus complexes, alors les taux d'erreur sont plus faibles et les améliorations sont plus significatives que lorsqu'il est combiné avec des algorithmes de clustering utilisant des modèles moins complexes.

Des expérimentations ont également été menées en utilisant deux autres algorithmes supervisés : C5 boosté 10 fois, pour observer si les informations ajoutées par le clustering aident aussi les méthodes qui combinent déjà plusieurs classifieurs, et DLG, une méthode qui utilise les *moindres généralisés*, et non des arbres de décision.

Les tables 4 et 5 présentent les taux d'erreur pondérés calculés dans ces deux cas. Il est alors très intéressant de remarquer que, malgré l'utilisation de différents algorithmes supervisés, les méthodes qui minimisent le taux d'erreur en validaton croisée sur les différents jeux de données restent les mêmes. En particulier, EM1 et EM2 surpassent encore K-means et LAC dans la grande majorité des cas.

	C5 seul	C5 + Aléa	C5 + K-means	C5 + LAC	C5 + EM1	C5 + EM2
ecoli	44.9	45.2	42.6	**39.9**	43.5	43.1
glass	**33.9**	43.8	35.7	37.8	35.6	34.4
image	3.3	3.9	3.2	**3**	**3**	**3**
iono	8.3	9.2	8.5	9.2	**6.6**	7.8
iris	5.6	6.3	5.3	**4.1**	4.8	4.4
pima	31.1	32.1	31.2	30.5	30.8	**29.7**
sonar	25.8	28.2	25.7	24.3	21.6	**21.3**
vowel	16	21	14.9	15	15.3	**14.5**
wdbc	4.6	4.8	3.8	4.2	4.1	**3.6**
wine	7	7.7	6.8	7.2	**2.9**	3.7

TAB. 4 – Taux d'erreur pondéré (en %) de C5 boosté seul et C5 boosté enrichi par les algorithmes de clustering.

Les tables 6 et 7 résument les comparaisons d'algorithmes de clustering selon l'algorithme supervisé utilisé. Il est de nouveau intéressant de noter que l'ordre dans lequel les méthodes de clustering sont rangées reste le même quelle que soit la méthode supervisée utilisée. Par contre, d'autres méthodes que EM2 deviennent significativement meilleures que l'algorithme supervisé correspondant seul. C'est par exemple le cas de EM1 lorsque C5 est utilisé.

Les tests ont également été menés en utilisant la seconde méthode de combinaison d'algorithmes supervisés et non supervisés, c'est-à-dire en exécutant C4.5 sur chaque groupe de données identifié par les algorithmes de clustering. La table 8 reporte les taux d'erreur obtenus dans ce cas, et la table 9 résume les résultats.

Nous observons donc encore dans ce cas que, malgré l'utilisation d'une autre méthode de combinaison d'algorithmes supervisés et non supervisés, l'ordre dans lequel les méthodes de clustering sont rangées reste le même.

	DLG seul	DLG + Aléa	DLG + K-means	DLG + LAC	DLG + EM1	DLG + EM2
ecoli	60.8	69.7	53.3	**51.5**	54.3	54.8
glass	47.1	60.1	49.6	47.5	47.2	**46.5**
image	9.8	14.2	9.7	9.1	9.1	**8.9**
iono	22.5	32	18.9	21.2	**12.9**	15.4
iris	9.1	15.6	8	**5.1**	6.4	6.5
pima	40.6	43.7	39.1	39.9	38.6	**38.4**
sonar	45.5	45.9	43.5	42.9	40.8	**38**
vowel	50.5	64.5	44.5	45	44	**42.7**
wdbc	9.2	10.7	7.7	7.8	7.1	**6.9**
wine	18.5	22.1	18.1	16.3	**6.9**	8

TAB. 5 – Taux d'erreur pondéré (en %) de DLG seul et DLG enrichi par les algorithmes de clustering.

	C5 seul	C5 + Aléa	C5 + K-means	C5 + LAC	C5 + EM1	C5 + EM2
nb vict	-	0/10	7/3	7/3	**9/1**	**9/1**
vict sign	-	0/0	0/0	0/0	0/0	0/0
wilcoxon	-	-2.88	1.41	1.31	**2.04**	**2.67**
moyenne	18	20.2	17.8	17.5	16.8	**16.5**

TAB. 6 – Comparaison de C5 boosté seul avec C5 boosté enrichi par les algorithmes de clustering.

	DLG seul	DLG + Aléa	DLG + K-means	DLG + LAC	DLG + EM1	DLG + EM2
nb vict	-	0/10	9/1	9/1	9/1	**10/0**
vict sign	-	0/2	1/0	**2/0**	**2/0**	**2/0**
wilcoxon	-	-2.88	**2.15**	**2.77**	**2.77**	**2.88**
moyenne	31.4	37.9	29.2	28.6	26.7	**26.6**

TAB. 7 – Comparaison de DLG seul avec DLG enrichi par les algorithmes de clustering.

	C4.5 seul	C4.5 + Aléa	C4.5 + K-means	C4.5 + LAC	C4.5 + EM1	C4.5 + EM2
ecoli	48.5	51.1	39.3	**32.1**	39.3	43.4
glass	**32.6**	46.4	33.9	32.8	33.1	33.1
image	4.8	6.8	4.8	4.8	4.3	**4**
iono	14.1	15.9	13.6	11.2	**10.1**	11.1
iris	7.3	7.6	5.5	**3.1**	4.9	4
pima	31	31.4	30.3	32.2	30.2	**28.9**
sonar	31	32.9	26.5	24.5	25.4	**22.3**
vowel	29.5	41.8	26.1	26.3	26	**25**
wdbc	5.9	6.5	3.5	4	3.5	**2.7**
wine	8.7	7.5	10.4	10	**2.2**	3

TAB. 8 – Taux d'erreur pondéré (en %) de C4.5 seul et C4.5 enrichi par les algorithmes de clustering selon la seconde méthode de combinaison.

	C4.5 seul	C4.5 + Aléa	C4.5 + K-means	C4.5 + LAC	C4.5 + EM1	C4.5 + EM2
nb vict	-	1/9	7/2	6/3	**9/1**	**9/1**
vict sign	-	0/3	0/0	0/0	1/0	**3/0**
wilcoxon	-	-2.46	1.1	1.2	**2.72**	**2.77**
moyenne	21.3	24.8	19.4	18.1	17.9	**17.8**

TAB. 9 – Comparaison de C4.5 seul avec C4.5 enrichi par les algorithmes de clustering selon la seconde méthode de combinaison.

4 Conclusion

Nous avons présenté dans ce papier une nouvelle méthode d'évaluation d'algorithmes de clustering objective et quantitative qui consiste à comparer les résultats d'un algorithme supervisé lorsqu'il est (ou pas) aidé par de l'information issue des résultats d'un algorithme de clustering. Nous avons considéré différents algorithmes supervisés et différentes façons de combiner les résultats d'algorithmes supervisés et non supervisés. Les expérimentations ont ainsi mis en avant que l'ordre dans lequel les méthodes de clustering sont rangées reste le même quels que soient l'algorithme supervisé et la méthode de combinaison utilisés. Ceci montre donc la stabilité de la nouvelle méthode d'évaluation que nous proposons.

Les expérimentations ont également mis en avant que les méthodes de clustering basées sur l'utilisation de modèles plus complexes surpassent les méthodes utilisant des modèles moins complexes. Ce résultat n'est pas surprenant mais montre le comportement cohérent de notre méthode d'évaluation. Il serait maintenant intéressant de valider ces résultats en menant d'avantage d'expérimentations en utilisant d'autres algorithmes supervisés, non supervisés, et aussi d'autres jeux de données.

Nous avons également montré que les résultats des algorithmes supervisés sont améliorés lorsqu'ils utilisent de l'information issue d'algorithmes *réels* de clustering. Nous postulons que les algorithmes supervisés peuvent bénéficier de cette information car elle est de nature très différente. En particulier, les algorithmes de clustering peuvent aider les algorithmes supervisés à spécialiser leurs traitements en fonction de différentes zones de l'espace spécifiques. Ils peuvent également aider les algorithmes supervisés à décrire des zones de décision plus complexes. Il semble donc intéressant de poursuivre cette recherche dans le cadre plus général de la combinaison de classifieurs lorsque l'un des classifieurs est non supervisé.

Comme nous l'avons vu dans nos expérimentations, il semble qu'utiliser le clustering pour partitionner l'espace des objets puis exécuter des apprentissages supervisés indépendants sur chaque groupe d'objets créé produise de meilleurs résultats que d'enrichir les jeux de données en créant de nouveaux attributs puis exécuter un apprentissage supervisé sur le jeu de données enrichi (comparaison des résultats des tables 3 et 9). Mais cela peut être la conséquence de la méthode que nous avons utilisée pour créer de nouveaux attributs, qui accroît de manière significative la taille du jeu de données. Il serait donc intéressant de creuser ce point dans de futures recherches.

Enfin, dans des travaux futurs, il semble également intéressant de trouver d'autres tâches aussi objectives que l'apprentissage supervisé et pour lesquelles le clustering serait un prétraitement intéressant, afin de tester notre méthode d'évaluation dans un tel cadre.

Références

ALI K. M. & PAZZANI M. J. (1996). Error reduction through learning multiple descriptions. *Machine Learning*, **24**(3), 173–202.

ALPAYDIN E. (1999). Combined 5x2cv F-test for comparing supervised classification learning algorithms. *Neural Computation*, **11**(8), 1885–1892.

APTE C. V., NATARAJAN R., PEDNAULT E. P. D. & TIPU F. A. (2002). A probabilistic estimaton framework for predictive model analytics. *IBM Systems Journal*, **41**(3).

BLAKE C. & MERZ C. (1998). UCI repository of machine learning databases [http ://www.ics.uci.edu/∼mlearn/MLRepository.html].

BREIMAN L. (1996a). Bagging predictors. *Machine Learning*, **24**(2), 123–140.

BREIMAN L. (1996b). Bias, variance, and arcing classifiers. Technical Report 460, Statistics Department, University of California.

CANDILLIER L., TELLIER I., TORRE F. & BOUSQUET O. (2005). SSC : Statistical Subspace Clustering. In P. PERNER, Ed., *Machine Learning and Data Mining in Pattern Recognition*, LNCS, p. 100–109, Leipzig, Germany : Springer Verlag.

DIETTERICH T. G. (1998). Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, **10**(7), 1895–1923.

DIETTERICH T. G. (2000). Ensemble methods in machine learning. In J. KITTLER & F. ROLI, Eds., *1st Int. Workshop on Multiple Classifier Systems*, volume 1857 of *LNCS*, p. 1–15 : Springer-Verlag.

DOMENICONI C., PAPADOPOULOS D., GUNOPOLOS D. & MA S. (2004). Subspace clustering of high dimensional data. In *SIAM Int. Conf. on Data Mining*.

FREUND Y. & SCHAPIRE R. E. (1996). Experiments with a new boosting algorithm. In *Int. Conf. on Machine Learning*, p. 148–156.

GAMA J. & BRAZDIL P. (2000). Cascade generalization. *Machine Learning*, **41**(3), 315–343.

MEIR R. & RÄTSCH G. (2003). An introduction to boosting and leveraging. In S. MENDELSON & A. SMOLA, Eds., *Advanced Lectures on Machine Learning*, number 2600 in LNAI, p. 119–184. Springer-Verlag.

PARSONS L., HAQUE E. & LIU H. (2004). Evaluating subspace clustering algorithms. In *Workshop on Clustering High Dimensional Data and its Applications, SIAM Int. Conf. on Data Mining*, p. 48–56.

QUINLAN J. R. (1993). *C4.5 : Programs for Machine Learning*. KAUFM.

QUINLAN R. (2004). Data mining tools see5 and c5.0.

WEBB G. I. & AGAR J. W. M. (1992). Inducing diagnostic rules for glomerular disease with the DLG machine learning algorithm. *Artificial Intelligence in Medicine*, **4**, 419–430.

WOLPERT D. H. (1992). Stacked generalization. *Neural Networks*, **5**, 241–259.

YE L. & SPETSAKIS M. (2003). *Clustering on Unobserved Data using Mixture of Gaussians*. Rapport interne, York University, Toronto, Canada.