Une approche ensembliste inspirée du boosting en classification non supervisée

Romain Billot (*,**,***), Henri-Maxime Suchier (*,****)
Stephane Lallich (*)
,

* Université Lyon 2, Laboratoire ERIC, 5 avenue Pierre Mendès-France,
69676 Bron Cedex, France
** Laboratoire d'Ingénierie Circulation Transports (LICIT),INRETS-ENTPE
25 Avenue François Mitterand Case 24, 69675 Bron Cedex, France
*** Laboratoire de Mathématiques Appliquées aux Systèmes (MAS), Ecole Centrale Paris,
92295 Châtenay-Malabry- France
**** Laboratoire informatique, Agrocampus Rennes, 65 rue de Saint-Brieuc, CS 84215
35042 Rennes Cedex - France

Contacts : billotro@gmail.com , hmsuchier@gmail.com , stephane.lallich@univ-lyon2.fr

Résumé. En classification supervisée, de nombreuses méthodes ensemblistes peuvent combiner plusieurs hypothèses de base afin de créer une règle de décision finale plus performante. Ainsi, il a été montré que des méthodes comme le *bagging* ou le *boosting* pouvaient se révéler intéressantes, tant dans la phase d'apprentissage qu'en généralisation. Dès lors, il est tentant de vouloir s'inspirer des grands principes d'une méthode comme le *boosting* en classification non supervisée. Or, il convient préalablement de se confronter aux difficultés connues de la thématique des ensembles de regroupeurs (correspondance des classes, agrégation des résultats, qualité) puis d'introduire l'idée du *boosting* dans un processus itératif. Cet article propose une méthode ensembliste inspirée du *boosting*, qui, à partir d'un partitionnement flou obtenu par les c-moyennes floues (fuzzy-c-means), va insister itérativement sur les exemples difficiles pour former une partition dure finale plus pertinente.

1 Introduction

Il est courant de séparer le domaine de l'apprentissage automatique en deux domaines distincts. D'un coté, l'apprentissage supervisé désigne un cadre où les exemples sont reliés à une information relative à leur classe, à un concept. Les méthodes supervisées produisent par la suite, à partir d'une base d'exemples d'apprentissage pour lesquels la classe est connue, une règle de décision visant à prédire la classe de nouvelles observations. Cette règle de décision, appellée aussi classifieur ou hypothèse, peut être considérée géométriquement comme une hypersurface séparant les exemples représentés dans un espace multidimensionnel.

A contrario, cette notion de classe ou de concept est absente dans le cadre de l'apprentissage non supervisé. Aucune information *a priori* n'étant disponible, les techniques non supervisées visent à détecter des structures de groupes fondées sur des notions de distance ou de similarité entre les exemples (Lerman (1970)). C'est précisement dans ce cadre que se place ce travail de recherche qui part d'un constat simple : en classification supervisée, des méthodes dites ensemblistes ont au cours des dernières années montré des performances tout à fait intéressantes. Dans le cas particulier du *boosting*, il est question d'un processus itératif qui va repondérer les exemples en insistant sur ceux mal classés par la méthode d'apprentissage à une itération donnée (Freund et Schapire (1997)).

Dès lors, il est tentant de vouloir s'inspirer d'un processus comme le *boosting* dans le domaine de la classification non supervisée. Cette transposition n'est pas du tout immédiate et soulève un certain nombre de problèmes : tout d'abord, le *boosting* repose sur des justifications théoriques solides et il serait présomptueux de prétendre appliquer rigoureusement une telle méthode. C'est pourquoi la contribution de ce travail doit simplement être considérée comme une approche ensembliste inspirée des grands principes du *boosting*. Dans un premier temps, il faut donc se confronter aux problèmes classiques liés aux ensembles de regroupeurs, présentés dans la section 2. D'autre part, la notion d'exemple "difficile", intuitive en apprentissage supervisé (les exemple difficiles sont par essence les exemples mal classés par la méthode d'apprentissage de base), reste, dans le domaine non supervisé, à définir. Si l'on veut, par analogie avec le *boosting*, insister à chaque itération sur les exemples difficiles, il faut précisement pouvoir les détecter, c'est-à-dire évaluer la qualité individuelle de bonne classification d'un exemple. Dans cet article, l'approche *UBLA* (Unsupervised Boosting-Like Approach) est proposée, qui détecte et repondère les exemples difficiles à regrouper à partir d'une partition floue, pour construire itérativement une matrice de co-association qui permettra de former finalement une partition dure plus pertinente au sens de certains critères de qualité.

2 Les ensembles de regroupeurs

La problématique des ensembles de regroupeurs consiste à combiner les résultats de plusieurs algorithmes de partitionnement (ex : centres mobiles) afin de former une partition plus pertinente des différentes instances. On trouvera une présentation des méthodes les plus populaires dans Hornik (2004). Les applications dans les domaines du rassemblement et de la réutilisation des connaissances sont nombreuses mais les ensembles de regroupeurs peuvent aussi permettre de combiner des partitionnements obtenus à partir de sous-ensembles d'individus ou d'attributs différents. La formation d'un ensemble de regroupeurs se heurte à certaines difficultés. D'une part, l'absence d'information sur la classe des instances pose le réel problème de la correspondance entre les classes construites par les différents partitionnements. D'autre part, la construction de la partition finale doit s'appuyer sur une méthode de consensus efficace, qui tient aussi compte de la qualité des différents partitionnements.

En ce qui concerne la correspondance des classes, Fred (2001) a proposé un index de cohérence qui va calculer la similarité entre deux classes de deux partitions différentes au sens du plus grand nombre de points partagés. La procédure, connue aussi sous le nom de *high matching score*, va donc désigner itérativement les deux classes possédant le plus grand score de correspondance. Strehl et Ghosh (2002) ont quant à eux proposé une fonction objectif qui pourrait assurer de trouver le partitionnement idéal, partageant le plus d'information possible

avec les partitionnements de base. Ils font appel à la notion *d'information mutuelle normalisée*. Dans Dimitriadou et al. (2001), il est question d'une méthode pour laquelle un ensemble de partitionnements durs ou flous peut être combiné afin de former une partition floue du jeu de données. La partition finale est formée par un vote qui minimise une fonction de dissimilarité entre les partitions initiales. Cette approche ne résiste pas à l'obstacle de la correspondance des groupes en considérant toutes les permutations possibles des matrices d'appartenance. Fred (2001) propose de contourner le problème de la correspondance des groupes en utilisant une matrice de co-association individus-individus qui ne considère que la fréquence avec laquelle deux exemples se retrouvent dans un même groupe.

Certaines méthodes ensemblistes supervisées ont déjà été l'objet de transpositions au domaine non supervisé. Leisch (1999) a par exemple appliqué le *bagging* en contexte non supervisé pour proposer l'algorithme *bagged clustering*, que nous utiliserons comme sous-procédure de notre contribution. L'application du *boosting* à la classification non supervisée a déjà été abordée dans Frossyniotis et al. (2004), où la méthode *boost-clust* repondère itérativement les exemples difficiles et forme une partition floue optimale. Toutefois, cet algorithme transpose par analogie et sans réelles justifications des concepts justifiés en contexte supervisé. L'apport le plus significatif des auteurs réside dans l'utilisation des vecteurs d'appartenance des partitions floues pour détecter les exemples sensibles. Les expérimentations proposées, trop succintes, ne permettent malheureusement pas d'évaluer de façon satisfaisante l'efficacité de la méthode et de ses différentes variantes.

3 L'algorithme *UBLA*

L'algorithme *UBLA* (Unsupervised Boosting-Like Approach) constitue une nouvelle approche inspirée du *boosting*, qui va construire itérativement une partition dure des données. L'algorithme se déroule en quatre phases, trois phases sont répétées à chaque itération de la procédure tandis que la quatrième et dernière phase établit la partition finale :

1. Chacune des t itérations de la procédure "boostée" commence par une *phase d'évaluation* qui va permettre la sélection des exemples difficiles. Au cours de cette évaluation les c-moyennes floues (Bezdek (1981)) sont effectuées une seule fois. Le caractère flou est utilisé pour calculer les critères de qualité à partir des degrés d'appartenance et les exemples vont être repondérés directement à la fin de cette phase d'évaluation.

2. Une deuxième phase commence alors, appelée *phase de regroupement/stabilisation*, qui constitue la sous-procédure de bagged clustering, où les c-moyennes floues classiques sont appliquées sur dix échantillons bootstrap obtenus à partir de la nouvelle distribution des poids. Il est important de noter que le fait d'opérer la repondération à la fin de la première phase entraîne la non utilisation du jeu de données initial pendant tout l'algorithme *UBLA*. Ainsi, même à la première itération, la procédure de *bagging* est effectuée à partir d'une nouvelle distribution des poids. Cet aspect implique que toutes les partitions sont obtenues à partir de distributions de poids non uniformes.

3. Le problème de correspondance entre les différents groupes est ici contourné par l'utilisation d'une matrice de co-association (Fred (2001)) qui intervient lors de la phase 3 de l'algorithme. Ainsi, les points sont considérés deux à deux et la matrice A, individus-

individus, est mise à jour lorsque deux points sont dans une même classe pour une partition d'une itération donnée.

4. A la fin des itérations dites de "*boosting*", la partition finale est formée par un vote majoritaire sur la matrice A construite itérativement (phase 4).

Avant de détailler le processus (algorithme 1), il semble indispensable de s'attarder dans les sous-sections suivantes sur quelques points essentiels de la méthode *UBLA*.

3.1 Le critère local de qualité d'un exemple

Une approche inspirée du *boosting* doit mettre l'accent par itérations successives sur les exemples dits difficiles à classer. Comment détecter de tels exemples ? Là où cette notion est très intuitive en apprentissage supervisé (un exemple difficile est par essence mal classé par le classifieur choisi), il convient en apprentissage non supervisé de quantifier la qualité du positionnement de chaque exemple, du fait -dans la grande majorité des cas- de la non connaissance *a priori* de la structure des données. Il est judicieux, tout comme Frossyniotis et al. (2004), d'user du vecteur des degrés d'appartenance des exemples aux différentes classes formées par l'algorithme de base (les c-moyennes floues), à une itération t. Ce vecteur U est construit par l'algorithme des c-moyennes floues. Ainsi, u_{ij} représente le degré d'appartenance de l'exemple i à la classe j. La somme des termes du vecteur des degrés d'appartenance d'un individu quelconque vaut 1. A partir de ce vecteur U, il est possible de calculer l'entropie des degrés d'appartenance de l'individu i, et plus précisement l'entropie de Shannon qui constitue une mesure très utile et bien connue pour quantifier la notion de désordre. Le critère local E_i est défini par l'entropie de Shannon des degrés d'appartenance d'un exemple. Il est aisé de voir que ce critère permet de repérer le degré d'indécision du rattachement d'un exemple à une classe en particulier. Soit l'exemple d'une partition d'un jeu de données en quatre classes. Sachant que la somme des degrés d'appartenance vaut 1, l'exemple suivant sera forcément le plus difficile à classer de par l'égalité de ses degrés d'appartenance :

$$U_1 = (0.25, 0.25, 0.25, 0.25)$$

Dans ce cas, l'entropie de Shannon de U_1 est maximale, soit $E_1 = 2$. Nous sommes dans l'incertitude la plus totale, l'exemple pouvant appartenir avec la même croyance à n'importe quelle classe. Au contraire, un exemple très bien classé, dont l'appartenance à une des quatre classes apparaît clairement, possédera un vecteur U de la forme suivante :

$$U_2 = (0, 0, 0, 1)$$

Ici, l'entropie de Shannon est minimale, soit $E_2 = 0$. Le cas succinctement exposé traduit le fait qu'un exemple devra être d'autant plus repondéré que son critère de qualité local, à savoir l'entropie de ses degrés d'appartenance, est fort. Ainsi, les exemples semblant difficiles à classer seront favorisés (aucun dégré d'appartenance à une classe ne se détache). Ce type de critère n'est d'ailleurs pas sans rappeler la notion de rejet en contexte supervisé (Leray et al. (2000)).

3.2 Le critère global de qualité

Parallèlement à la qualité locale d'un exemple, la repondération doit tenir compte conjointement de la qualité globale du partitionnement. D'aucuns y verront une analogie avec l'erreur

en apprentissage utilisée par Freund et Schapire (1997) dans le *boosting* pour construire un coefficient qui entre en compte dans la favorisation exponentielle des exemples mal classés. Cette qualité globale est aussi et surtout envisagée par souci de logique. En effet, pourquoi repondérer un exemple apparemment difficile si la qualité de la partition dans laquelle il se trouve est médiocre ?

Pour conserver une certaine cohérence dans l'approche, il est logique que le critère global de qualité du partitionnement en cours soit la moyenne des critères locaux. Ainsi, à la b_{ieme} itération, le critère de qualité C_b est la moyenne des E_i courants. Plus précisement, ce critère offre la possibilité de repondérer un exemple en tenant compte non seulement de la qualité de classement intrinsèque au partitionnement (c'est le critère local E_i) mais aussi de la qualité globale du regroupement dans lequel il se trouve (critère C_b).

3.3 La repondération des exemples difficiles

Les performances du *boosting* reposent en partie sur la construction d'un ensemble d'hypothèses faibles construites à partir de distributions statistiques dans lesquelles sont favorisés les exemples dits *difficiles*. C'est ce principe qui est transposé dans notre proposition en prenant pour hypothèse de base l'algorithme classique des c-moyennes floues. L'intérêt du choix d'une partition floue réside bien sûr dans la possibilité de raisonner en terme de degré d'appartenance. L'erreur en apprentissage est donc remplacée ici par deux critères de qualité, local et global, définis ci-dessus. Il s'agit par la suite de mettre à jour les poids de chaque exemple en tenant compte de la qualité globale de la partition puis de la qualité locale de classification d'un point. A chaque itération b, le poids d'un exemple p est ajusté de la façon décrite en (1) (cf algorithme 1). Par conséquent, les poids des exemples dont l'entropie des degrés d'appartenance est supérieure à la moyenne de l'échantillon sont augmentés ($E_i > C_b$), tandis que ceux des autres se voient diminuer par normalisation. L'argument de l'exponentielle comporte deux termes. Le premier terme ($log_2 K - C_b$) se rapproche de zéro lorsque la qualité globale de la partition est mauvaise ($log_2 K$ est la borne supérieure du coefficient d'entropie à minimiser). Le deuxième terme (E_i) représente la qualité individuelle de classification d'un exemple i et est d'autant plus élévé que l'exemple est difficile. Logiquement, les exemples qui prendront exponentiellement le plus d'importance seront les points mal classés dans une partition de bonne qualité.

3.4 Utilisation d'une sous-procédure : le *bagged clustering*

Une autre approche inspirée des méthodes supervisées, en l'occurence le *bagging* (Breiman (1996)), peut être transposée en contexte non supervisé. Cette méthode, appelée *bagged clustering* a été proposée par Leisch (1999). La procédure combine une méthode de regroupement (centres-mobiles, c-moyennes floues) avec une classification hiérarchique. La nouveauté réside dans l'application de la méthode de clustering à plusieurs échantillons bootstrap du jeu de données de départ, puis dans la transposition du problème initial dans l'espace des centres formés par la méthode de base sur ces échantillons bootstrap. Le nombre de classes *a priori* est donc donné suffisament grand, le regroupement (centres-mobiles ou c-moyennes floues) est appliqué sur chaque échantillon bootstrap puis tous les centres finaux sont regroupés dans une matrice. Une classification hiérarchique sur les centres est ensuite effectuée puis chaque point

est assigné à la classe qui contient le centre dont il est le plus proche. La figure 1 résume la procédure.

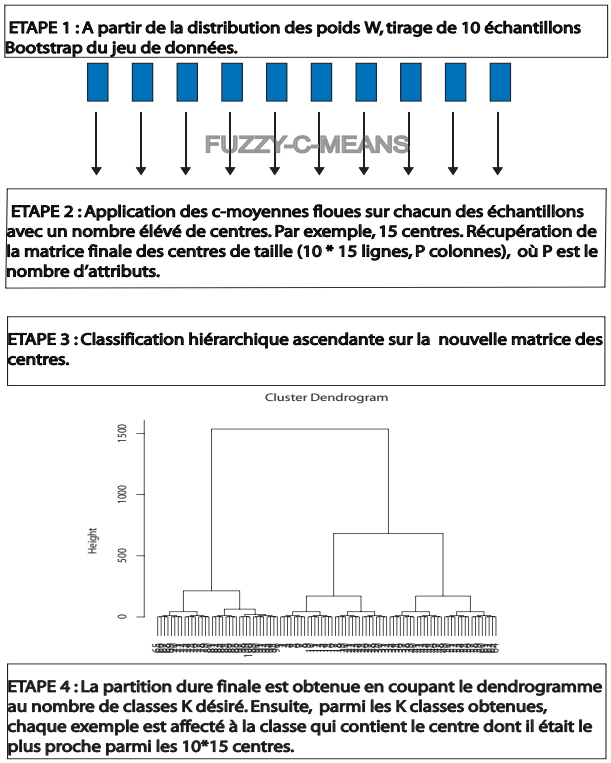

FIG. 1 – *La procédure de bagged clustering.*

Il a semblé intéressant d'introduire ce processus comme une sous-procédure de l'algorithme, qui stabilisera les résultats et donc la partition formée, sans pour autant dénaturer l'idée originale de repondérations successives.

3.5 Complexité de l'algorithme

A l'intérieur des phases 1 et 2, la complexité algorithmique reste linéaire en fonction de N. Lors de la phase 3, la complexité est quadratique (O(N^2)), à une constante près liée au nombre d'itérations T de *boosting*. On remarquera que l'utilisation d'une classification hiérarchique n'augmente pas la complexité car celle-ci est effectuée sur la matrice des centres, dont la taille est liée à des constantes. Enfin, la formation de la partition finale (phase 4) s'effectue en O(N^2) par parcours de la matrice A. Finalement, c'est l'utilisation d'une matrice de co-association qui borne de façon générale la complexité de la procédure *UBLA* en O(N^2).

Algorithme 1 : Pseudo-code de l'algorithme *UBLA*

Entrées : le vecteur des N instances $(x_1, ..., x_N)$ où chaque $x_i \in \Re^p$;
Un nombre K initial de groupes et nombre T d'itérations;
Sorties : Une partition du jeu de données.
début
  Initialisation à zéro des termes de la matrice de co-association A $\in M_{N,N}$;
  Initialisation du vecteur des qualités globales. $C = (0, ..., 0), \in \Re^T$;
  Initialisation des poids. $W_i = \frac{1}{N}\ \forall i \in 1, ..., N$;
  pour *b allant de 1 à T* **faire**
    Z=0 (coefficient de normalisation);
    E=(0,...,0), $\in \Re^N$ le vecteur des critères locaux des exemples. ;
    PHASE 1 : EVALUATION
    Application des c-moyennes floues avec la distribution des poids W;
    Calcul des critères de qualité;
    Repondération des exemples;
    (1) **si** $E_i > C_b$ **alors**
      $W_i^{b+1} = W_i^b * \exp\left[(log_2(K) - C_b) + E_i\right]$
    fin
    Normalisation de tous les poids;
    PHASE 2 : REGROUPEMENT/STABILISATION
    sous procédure de *bagged clustering*
    PHASE 3 : MISE A JOUR DE LA MATRICE A
    pour *i allant de 1 à N* **faire**
      pour *j allant de 1 à N* **faire**
        si *i et j sont dans la même classe* **alors**
          $A_{ij} = A_{ij} + \frac{1}{T}$
        fin
      fin
    fin
  fin
  PHASE 4 : FORMATION DE LA PARTITION FINALE PAR VOTE MAJORITAIRE
  Pour chaque exemple i et j, si $A[i, j] > \frac{1}{2}$, regrouper les deux exemples dans la même classe.
  Les éventuels exemples "seuls" formeront des singletons.
fin

4 Expérimentations

4.1 Choix du critère de qualité externe de la partition finale

Un problème majeur de l'apprentissage non supervisé réside dans l'absence de juge de paix pour départager les différentes méthodes. En classification supervisée, ce juge existe et est aussi bien local (erreur de prédiction) que global (taux d'erreur). Nous avons donc arbitrairement choisi un juge de paix qui n'est ici que global et qui compare la qualité des partitions finales. Les différentes mesures de validation des partitions ont été bien résumées dans Halkidi et al. (2001). Pour valider expérimentalement les performances de l'approche *UBLA*, à savoir la qualité de la partition dure proposée, des indices fondés sur des mesures de compacité (dispersion intra-classe) et de séparation (dispersion inter-classe) semblent bien adaptés

car les algorithmes de regroupement classiques cherchent également à optimiser des critères liés aux mêmes notions. Ainsi, des indices comme l'indice de Dunn, la silhouette moyenne et le ratio intra/inter (*wb*) sont tout à fait pertinents. Une combinaison de ces trois indices peut être intéressante dans les cas où les valeurs seront très serrées. Rappelons que l'indice de Dunn et la silhouette moyenne d'une partition sont à maximiser tandis que le ratio intra/inter est à minimiser. L'indice de qualité suivant :

$$Ind = Dunn + silhouette - ratiowb$$

combinera donc simplement les trois indices et sera lui aussi à maximiser. La comparaison s'est effectuée entre la méthode *UBLA*, les centres-mobiles de MacQueen (1967), les c-moyennes floues de Bezdek (1981) et la procédure de *bagged clustering* classique de Leisch (1999), qui constitue rappelons-le une sous-procédure de notre proposition. Dans la grande majorité des cas, la meilleure partition domine les trois autres au sens des trois indices (maximisation de *Dunn* et de la *silhouette moyenne*, minimisation du ratio *wb* par rapport aux deux autres). Mais pour les rares cas plus incertains, la valeur finale de Ind est prise pour sélectionner la méthode formant la meilleure partition pour une expérimentation précise.

4.2 Résultats

La méthode *UBLA* a été testée dans l'environnement statistique *R* sur onze jeux de données, disponibles dans les bases de données bien connues du web. Comme le rappelait Diday (1974), les algorithmes de regroupement peuvent fournir des solutions satisfaisantes, mais qui ne sont pas forcément optimales. Ainsi, plusieurs répétitions des centres-mobiles sur un même jeu de données peuvent donner des résultats différents. Par conséquent, nous avons pour nos expérimentations rélancé les différents algorithmes dix fois et sélectionné le meilleur résultat pour chaque méthode. Il serait d'ailleurs intéressant d'analyser la stabilité des partitions des algorithmes de regroupement comme dans Bertrand et Bel Mufti (2006). Les caractéristiques des jeux de données (nombre d'individus et de variables) sont rappellées dans le tableau suivant. La valeur de l'indice de qualité Ind, le nombre d'itérations T et le nombre final de groupes formé par notre méthode (qui peut rappelons-le être différent de K), sont également introduits dans le tableau 1. Dans le cas où le nombre K final de classes formées par $UBLA$ était différent du K initial, c'est avec ce K final que le calcul de l'indice de qualité, qui dépend du nombre de classes, s'effectue pour les centres-mobiles et les c-moyennes floues (ceci assure la cohérence des comparaisons des partitions formées).

Les résultats obtenus sont assez intéressants. Sur un total de 55 expérimentations, la méthode *UBLA* surpasse les centres mobiles et les c-moyennes floues dans 47 cas. (Il y a en plus un ex-aequo). En considérant les comparaisons de *UBLA* aux deux méthodes de manière indépendante, il est observé que, toujours au sens de l'indice de qualité Ind, notre approche domine d'une part les centres-mobiles dans 39 cas sur 45, de l'autre améliore les c-moyennes floues dans 41 cas sur 45. La méthode *UBLA* est supérieure au *bagged clustering* dans 37 cas sur 55 et égale dans 8 cas sur 55. Nous pouvons constater que le nombre d'itérations est généralement assez faible, contrairement à ce qui pouvait être attendu d'une procédure de boosting. Un trop grand nombre d'itérations peut même dans certains cas entraîner des chutes de performances, les répondérations successives des exemples déformant trop la structure initiale du jeu de données. Pour conclure, nous pouvons dire que les résultats montrent qu'un processus ensembliste

Jeu de donnees		***UBLA***			**Centres mobiles**	**C-moyennes floues**	***bagged clustering***
Nom	**K**	**nb iter**	**K final**	Ind	Ind	Ind	Ind
	2	3	2	**-0.02**	-0.30	-0.67	-0.20
Breast cancer	3	7	3	**-0.05**	-0.41	-0.67	-0.42
77 individus	4	6	4	**-0.2**	-0.34	NaN	-0.38
9 attributs	5	3	5	**-0.21**	-0.37	NaN	-0.29
	10	4	5	-0.194	-0.190	NaN	**-0.134**
	2	2	2	**1.77**	1.17	1.17	**1.77**
Baviere	3	2	3	**1.52**	0.42	0.36	0.52
89 individus	4	2	4	**1.47**	0.46	0.46	0.51
3 attributs	5	6	5	**1.47**	0.52	0.52	0.52
	10	3	9	**0.54**	0.43	0.52	0.45
	2	2	2	**-0.06**	-0.59	-0.52	-0.08
Diabetis	3	3	3	**0.04**	-0.5	-0.5	-0.42
300 individus	4	4	4	**0.06**	-0.38	-0.65	-0.37
8 attributs	5	6	5	**-0.05**	-0.41	-0.70	-0.38
	10	6	10	**-0.1**	-0.32	-0.79	-0.41
	2	3	2	**0.28**	-0.42	-0.46	-0.08
Heart	3	3	3	**0.133**	-0.5	-0.46	-0.53
100 individus	4	4	4	**0.08**	-0.54	NaN	-0.52
13 attributs	5	5	5	**-0.04**	-0.45	NaN	-0.46
	10	13	10	**-0.28**	-0.31	Nan	**-0.28**
	2	2	2	**0.45**	0.25	0.041	**0.45**
Thyroid	3	5	3	**0.51**	0.38	-0.032	**0.51**
75 individus	4	8	4	**0.50**	-0.04	-0.10	-0.52
5 attributs	5	3	5	0.48	-0.013	-0.10	**0.51**
	10	2	12	**0.38**	-0.02	-0.05	0.19
	2	2	2	**0.8**	-0.54	-0.69	**0.8**
German	3	3	3	**0.39**	-0.51	-0.78	-0.63
300 individus	4	10	5	**0.22**	-0.59	-0.77	-0.02
20 attributs	10	11	5	**0.04**	-0.49	-1.75	-0.58
	15	10	14	**-0.28**	-0.34	-1.65	-0.5
	2	7	2	**0.59**	0.21	0.19	0.56
Indiens PIMA	3	7	3	**0.57**	0.17	0.16	0.20
768 individus	4	1	4	0.15	0.10	0.07	**0.16**
9 attributs	5	10	4	0.0699	0.0683	-0.0297	**0.15**
	10	1	10	0.0278	0.0270	-0.008	**0.08**
	2	6	2	**0.32**	0.13	0.14	0.29
Zurich	3	3	3	0.22	0.14	0.14	**0.24**
235 individus	4	5	4	**0.23**	0.17	0.10	0.13
16 attributs	5	1	5	**0.15**	0.10	0.10	0.13
	10	2	10	**0.11**	0.08	-0.01	0.07
	2	1	2	**0.03**	-0.04	-0.08	-0.21
Wind	3	1	3	**-0.03**	-0.07	-0.09	-0.26
6574 individus	4	2	4	-0.19	**-0.08**	-0.12	-0.19
15 attributs	5	6	5	**0.016**	-0.13	-0.22	-0.2
	10	2	10	-0.35	**-0.11**	-0.25	**0.11**
	5	2	5	**0.58**	**0.58**	**0.58**	**0.58**
X8d5k	10	10	7	**0.26**	0.05	-0.05	0.06
1000 individus	15	5	15	**0.0064**	-0.042	-0.2	-0.04
8 attributs	20	5	11	**0.064**	-0.11	-0.2	-0.11
	30	5	30	**0.04**	-0.07	Nan	-0.26
	2	9	2	-0.93	**0.29**	**0.29**	0.0037
X2d2k	3	1	3	**0.07**	0.02	0.04	0.01
1000 individus	4	10	4	-0.04	**-0.00052**	-0.05	-0.01
2 attributs	5	5	5	-0.02	0.02	**0.03**	-0.04
	10	1	10	-0.01	**0.11**	0.07	0.01

TAB. 1 – *Tableau des résultats*

itératif, qui insiste sur les exemples difficiles à classer, peut grandement améliorer un partitionnement au sens de critères intra/inter classe. Les illustrations graphiques de la sous-section suivante vont permettre de visualiser les performances de la méthode *UBLA* pour d'autres jeux de données, plus propices à la visualisation graphique car comportant peu d'attributs.

4.3 Illustration graphique

La figure 2 compare les partitions formées par l'approche *UBLA* et des c-moyennes floues pour le jeu de données des IRIS de Fisher, bien connu des statisticiens. Il s'agit de 150 individus (des plantes) décrits par 4 variables. Il existe donc six plans de représentation possibles pour ces données. Pour cet exemple, le nombre de classes donné *a priori* était $K = 2$. La figure 3 illustre quant à elle la classification en trois classes du jeu de données DNase (deux dimensions).

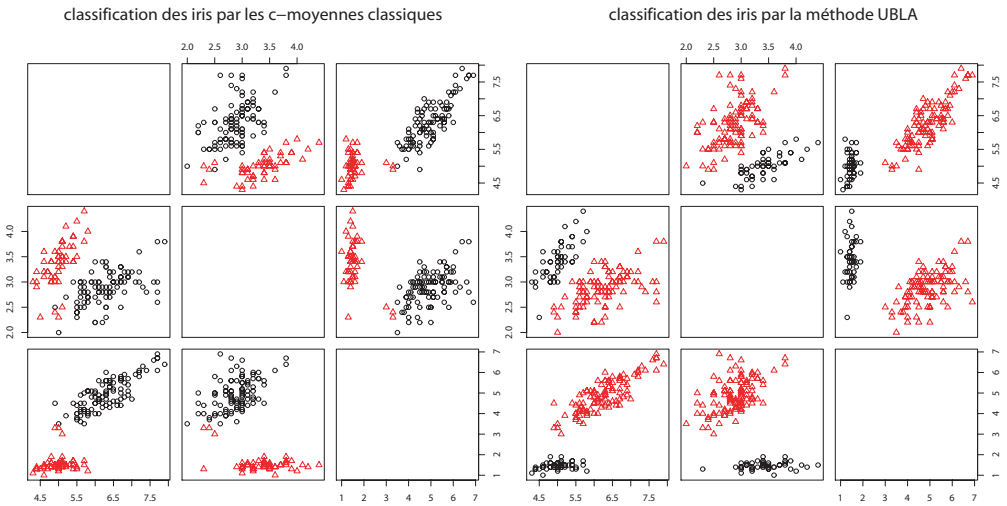

FIG. 2 – *Visualisation de la classification par les c-moyennes (à gauche) du jeu de données IRIS dans les six plans possibles de représentation. Les exemples des deux classes sont symbolisés pas des triangles et des ronds. On remarquera sur les différents plans la mauvaise classification de trois individus. A droite, cette erreur est corrigée par la méthode UBLA qui affecte les trois exemples à la bonne classe.*

5 Conclusion

Dans cet article, nous avons proposé une nouvelle approche ensembliste en apprentissage non supervisé, qui s'inspire des principes du *boosting*. A chaque itération, l'algorithme repère au sens de certains critères les exemples difficiles pour leur donner plus d'importance à l'itération suivante. Cette approche se sert ainsi dans un premier temps d'une partition floue, formée par la méthode des c-moyennes floues pondérées, pour glisser ensuite vers une partition finale dure.

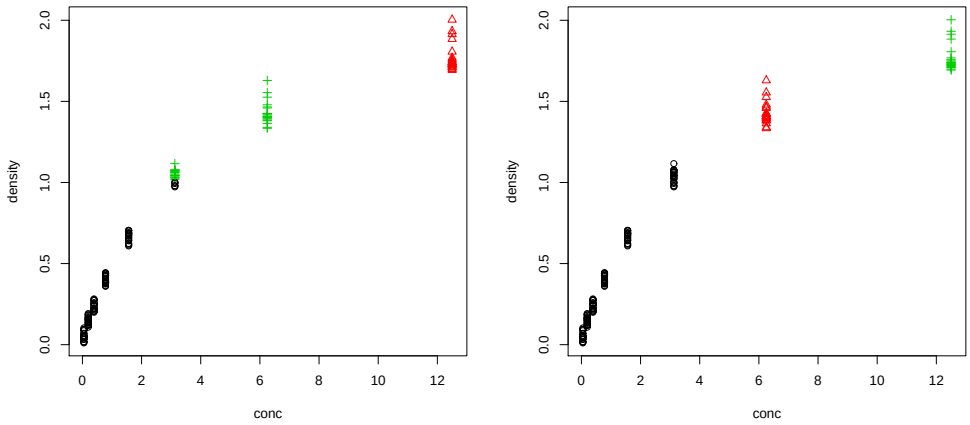

FIG. 3 – *Visualisation de la classification du jeu de données DNase (2D) par les c-moyennes classiques (à gauche) puis UBLA (à droite). Nous remarquons une zone sensible au milieu-gauche de la figure ou deux classes ne sont pas nettement séparées par les c-moyennes floues. A droite, l'incertitude est corrigée.*

Les premiers résultats sont prometteurs et laissent penser qu'une telle approche peut améliorer la qualité des partitions finales formées en termes de compacité et de séparation. A présent, il serait pertinent de comparer les performances de la méthode *UBLA* avec plusieurs autres approches ensemblistes proposées dans le domaine non supervisé (ex : Frossyniotis et al. (2004)). Parmi les perspectives de ce travail, il faudrait trouver une alternative à la matrice de co-association qui est coûteuse et proposer de nouveaux critères de qualité. Par exemple en implémentant des critères qui ne soient pas essentiellement fondés sur des mesures de dispersion intra- et inter-classe. Il serait de plus envisageable de s'affranchir de l'utilisation de la procédure de *bagged clustering* pour implémenter notre propre procédure de stabilité, fondée sur des tirages *bootstrap* ou des modifications de l'échantillon par rapport à la nouvelle pondération.

Pour conclure, nous pouvons dire que la transposition du *boosting* à l'apprentissage non supervisé doit se faire avec la plus grande prudence. En effet, des concepts justifiés théoriquement en contexte supervisé ne s'appliquent pas de façon directe en classification non supervisée. Notre travail se veut donc simplement inspiré de certains concepts du *boosting*, principalement la repondération des exemples difficiles. Il reste maintenant à approfondir le cadre théorique de la méthode proposée.

Références

Bertrand, P. et G. Bel Mufti (2006). Loevinger's measures of rule quality for assessing cluster stability. *Computational Statistics & Data Analysis 50*(4), 992–1015. available at http ://ideas.repec.org/a/eee/csdana/v50y2006i4p992-1015.html.

Une approche ensembliste inspirée du boosting en classification non supervisée.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA, USA : Kluwer Academic Publishers.

Breiman, L. (1996). Bagging predictors. *Maching Learning 24*(2), 123–140.

Diday, E. (1974). Optimization in non-hierarchical clustering. *Pattern Recognition 6*(1), 17–33.

Dimitriadou, E., A. Weingessel, et K. Hornik (2001). Voting-merging : An ensemble method for clustering. In *ICANN '01 : Proceedings of the International Conference on Artificial Neural Networks*, London, UK, pp. 217–224. Springer-Verlag.

Fred, A. L. N. (2001). Finding consistent clusters in data partitions. In *MCS '01 : Proceedings of the Second International Workshop on Multiple Classifier Systems*, London, UK, pp. 309–318. Springer-Verlag.

Freund, Y. et R. E. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences 55*(1), 119–139.

Frossyniotis, D., A. Likas, et A. Stafylopatis (2004). A clustering method based on boosting. *Pattern Recognition Letters 25*(6), 641–654.

Halkidi, M., Y. Batistakis, et M. Vazirgiannis (2001). On clustering validation techniques. *Journal of Intelligent Information Systems 17*(2-3), 107–145.

Hornik, K. (2004). Cluster ensembles. In *Workshop Ensemble Methods FAU. 2004*.

Leisch, F. (1999). Bagged clustering. In *Working Papers SFB Adaptive Information Systems and Modelling in Economics and Management Science, 51, Institut fr Informationsverarbeitung, Abt. Produktionsmanagement, Wien, 1999.*

Leray, P., H. Zaragoza, et F. d'Alché-Buc (2000). Pertinence des mesures de confiance en classification. In *12ème Congrès Francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Articifielle (RFIA 2000)*, Paris, France, pp. 267–276.

Lerman, I. (1970). *Les bases de la classification automatique*. Gauthier-Villars. Paris.

MacQueen, J. B. (1967). Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathemtical Statistics and Probability*, pp. 281–297.

Strehl, A. et J. Ghosh (2002). Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research (JMLR) 3*, 583–617.

Summary

Cluster Ensemble Methods have shown their efficiency to build a better clustering from many base clusterings. In order to form an efficient consensus partition, some problems must be solved such as the correspondence between groups, the results combination. In supervised learning, ensemble methods have produced superior results in learning and generalization. In this article, we propose to transpose the main principles of *boosting* to the clustering. Thanks to a fuzzy clustering, the *UBLA* algorithm (Unsupervised Boosting-Like Approach) aims at recognizing the sensitive examples and reweighting them in a *boosting* process which leads to a crisp clustering of a dataset.