Amélioration de la classification dans les sous espaces

Amel Boulemnadjel* , Fella Hachouf *

*Département d'électronique, Laboratoire d'Automatique et de robotique
Université Mentouri de Constantine.
Algérie, Route d'Ain El-Bey, 25000 Constantine.
amel_boulemnadjel@yahoo.fr, fhachouf@wissal.dz

RÉSUMÉ. Ce papier présente un nouvel algorithme pour la classification des données de grande dimension dans les sous espaces. C'est un algorithme itératif basé sur la minimisation d'une fonction objective. Cette nouvelle fonction est développée par l'intégration de la séparabilité et de la compacité des classes dans le quel nous avons introduit leur densité. Les résultats expérimentaux confirment que l'algorithme proposé donne de bons résultats sur différents types d'images en optimisant le temps d'exécution.

ABSTRACT. This paper presents a new algorithm for subspace clustering for high dimensional data, it is an iterative algorithm based on the minimization of objective function, this new function is developed by integrating the séparability and compactness of clusters in which we introduced the density. The experimental results confirm that the proposed algorithm gives good results on different types of images by optimizing the execution time.

MOTS-CLÉS : classification, sous espaces, séparabilité, compacité, classes, densité.

KEYWORDS: classification, subspace, séprability, compactness, clusters, density.

1. Introduction

La classification automatique des données consiste à diviser un jeu de données en sous-ensembles de données appelés classes pour que tous les individus dans une même classe soient similaires et les individus de classes distinctes soient dissimilaires. Typiquement, chaque classe est représentée par un individu qui s'appelle le centre de la classe ou par certaines informations dérivées de tous les individus de la classe qui sont suffisantes de décrire la classe. Les objets utilisés dans la classification peuvent disposer de certains attributs. La classification dans un tel espace de haute dimensionnalité est extrêmement difficile. Le calcul de similarité devient très coûteux. Un problème très connu sous le nom de malédiction de dimensionnalité (dimensionnality curse), qui est le manque de propriétés de données dans un espace de haute dimension. Dans le cas de données de grande dimensionnalité, les groupes peuvent être caractérisés uniquement par certains sous-ensembles de dimensions. Ces dimensions pertinentes peuvent être différentes d'un groupe à l'autre. Sur de tels problèmes, les techniques classiques de clustering fonctionnent mal car, fondées sur une distance entre objets définie globalement dans l'espace de description. Elles ne peuvent pas appréhender le fait que la notion de similarité varie d'un groupe à l'autre. De plus, il est assez naturel de penser que les données provenant de classes différentes vivent dans des sous-espaces différents. Une nouvelle problématique a donc émergé récemment, celle du subspace clustering [7]. C'est une extension de la classification traditionnelle qui recherche un ensemble de clusters qui peuvent être définis dans différents sous-espaces. L'intérêt de telles techniques est important dans le cadre de données contenant un nombre important de dimensions car elles permettent de faire face au problème de la dimensionnalité. De plus, elles permettent de fournir une description réduite des clusters obtenus car les clusters sont alors définis par un nombre restreint de dimensions. Les méthodes de subspace clustering peuvent être divisées en deux grandes familles de méthodes : d'une part, les méthodes heuristiques qui recherchent les dimensions permettant d'obtenir le meilleur clustering et, d'autre part, les méthodes basées sur des modèles de mélange qui modélisent le fait que les données vivent dans des sous-espaces. De nombreuses méthodes de subspace clustering utilisent des techniques heuristiques de recherche pour identifier les sous-espaces des classes. Parmi ces méthodes, on peut distinguer deux types d'algorithmes de recherche des sous-espaces : les méthodes de recherche dites bottom-up qui utilisent des histogrammes pour sélectionner les dimensions permettant de séparer efficacement les groupes. CLIQUE [1] fut l'un des premiers algorithmes bottom-up proposés pour rechercher des groupes dans des sous-espaces de l'espace original. L'idée est basée sur le fait que si une collection de points S est une classe dans un espace de k dimensions, alors S est ainsi une partie d'une classe dans n'importe quelle projection en (k-1) dimensions de cet espace. D'autre part, les techniques de recherche de type top-down sont des méthodes itératives qui, partant de l'espace entier comme solution initiale, pondèrent à chaque itération les dimensions ne semblant pas

contenir de groupe. La première méthode de ce type fut Proclus [2] qui évalue à chaque itération la qualité de la classification en calculant la distance moyenne entre les centres des groupes. Dans le même temps, de nombreuses méthodes basées sur les mélanges ont vu le jour. Ces méthodes génératives proposent de se placer dans les espaces propres des classes afin de prendre en compte le fait que les données vivent dans des sous-espaces de faible dimension. Pour cela, elles se basent sur le modèle de l'analyse factorielle [3]. D'autres méthodes sont basées sur la densité [4,6 ,8]. Dans ces méthodes, les classes sont considérées comme des régions de haute densité séparée par des régions de faible densité. La densité est représentée par le nombre d'objets de données dans le cluster. C'est pourquoi ces méthodes sont capables de chercher des classes de forme arbitraire. Un algorithme très populaire dans ce type d'algorithme de classification est DBSCA [4]. Toutes ces méthodes sont capables de retrouver efficacement les clusters et leur sous-espaces spécifiques mais elles nécessitent souvent des paramètres difficiles à régler par l'utilisateur et influant sur leurs performances (seuil de densité, nombre moyen de dimensions caractéristiques des clusters, distance minimale entre clusters, etc.). Des travaux plus récents basés aussi sur la densité règlent quelques problèmes comme ISA [8] et SUBCLU [6]. Un autre type de classification dite douce existe (soft subspace clustring) , parmi ces méthode on peut citer un algorithme nommé ESSC (Enhanced soft subspace clustering) [10] qui utilise deux informations concernant l'inter et l'intra classes, où la seule information utilisée dans ce type de classification est seulement la distance inter-classes. Cette modification introduit par ESSC apporte une grande amélioration sur la classification des images de texture. Dans ce type de textures, la densité est la même pour tous les motifs. Le problème est posé quand la densité des classes varie. Bien que la densité est un outil très important pour la classification des données de grande dimension qui donne de bon résultats [9], mais ESSC l'a ignorée. Les méthodes de classification utilisant la densité donnent de bons résultats avec un temps exécution assez coûteux. Dans ce travail, nous proposons une méthode de classification basée sur l'optimisation d'une fonction d'objective. Celle-ci est composée de deux termes. Le premier terme est la compacité dans lequel nous avons introduit la densité des classes sans augmenter le coût informatique. Ceci est réalisé par un prétraitement adapté pour l'extraction des attributs des données à classifier, le second terme est la séparabilité. Le papier est organisé comme suit : après une introduction, la section deux sera dédiée à la fonction objective proposée que nous expliciterons. Comme nous testons notre approche sur des images de textures, la section trois présente un aperçu des méthodes d'extraction des paramètres. Les résultats expérimentaux seront exposés et commentés dans la section quatre. Le papier se termine par une conclusion.

2. Fonction objective proposée

La fonction proposée est une extension de la fonction d'objective de l'algorithme ESSC [10] donnée par l'équation.1. Elle contient trois termes : le terme de compacité, le terme de l'entropie pondérée et le terme de séparabilité.

$$J_{ESSC}(v,u)=\sum_{i=1}^{c}\sum_{j=1}^{N}u_{ij}^{m}\sum_{k=1}^{D}w_{ik}(x_{jk}-v_{ik})^2+\gamma\sum_{i=1}^{c}\sum_{k=1}^{D}w_{ik}\ln w_{ik}-\eta\sum_{i=1}^{c}(\sum_{j=1}^{N}u_{ij}^{m})\sum_{k=1}^{D}w_{ik}(v_{ik}-v_{0k})^2 \quad \text{.....(1)}$$

Où
c : le nombre des classes
D : le nombre de dimension
N : taille de données
u : le degré d'appartenance
w : le poids de l'entropie de chaque dimension
v : centre des classes
v_0: centre initiale des classes
La modification que nous apportons à cette fonction est basée d'une part sur l'introduction de la densité n(i) dans le premier terme de compacité de l'équation .1 qui sera remplacé par l'équation.2 :

$$j_{cfm}(v,u)=\sum_{i=1}^{c}\sum_{j=1}^{N}u_{ij}^{m}\sum_{k=1}^{D}w_{ik}(x_{jk}-v_{ik})^2/n(i)+1 \quad \text{.....(2)}$$

Et d'autre part la suppression du terme de l'entropie pondérée qui sera remplacé par le calcul local de l'entropie dans la phase d'initialisation ; afin de diminuer le coût informatique en gardant toujours l'information de l'entropie avec la mise a jour de sa valeur à chaque itération. Donc la fonction d'objective modifiée est donnée par l'équation.3:

$$J_{fm}(v,u)=\sum_{i=1}^{c}\sum_{j=1}^{N}u_{ij}^{m}\sum_{k=1}^{D}w_{ik}(x_{jk}-v_{ik})^2/(n(i)+1)-\eta\sum_{i=1}^{c}(\sum_{j=1}^{N}u_{ij}^{m})\sum_{k=1}^{D}w_{ik}(v_{ik}-v_{0k})^2 \quad \text{.....(3)}$$

En utilisant le multiplicateur de Lagrange, nous déduisant les expressions de u et v qui minimisent la fonction d'objective équation.3, données respectivement par les équations 4 et 5.

$$u_{ij}=\frac{[\sum_{k=1}^{D}w_{ik}(x_{jk}-v_{ik})^2/(n(i)+1)-\eta(v_{ik}-v_{0k})^2]^{-1/m-1}}{\sum_{i=1}^{c}[\sum_{k=1}^{D}w_{ik}(x_{ik}-v_{ik})^2/(n(i)+1)-\eta(v_{ik}-v_{0k})^2]^{-1/m-1}} \quad \text{.....(4)}$$

Avec

$$\sum_{i=1}^{c}u_{ij}-1=0$$

$$v_{ik} = \frac{\sum_{j=1}^{N} u_{ij}(x_{jk} /(n(i)+1) - \eta v_{0k})}{\sum_{j=1}^{N} u_{ij}((1/(n(i)+1)) - \eta)} \quad \ldots\ldots(5)$$

On peut résumer les étapes de cette méthode dans l'algorithme suivant :

Algorithme

Etape 1 : Initialisation

- Entrées : le nombre de classe c, paramètres η, ε , initialisation arbitraire des centre des classe v_0 et les poids des entropies w_0
- Extraire les attributs de l'image : obtention de la matrice ***G*** de dimension D xN

Etape 2 : Traitement

Tant que u(t+1)-u(t) $\leq \varepsilon$ **faire**

- Calculer u selon l'équation (4)
- Calculer v selon l'équation (5)
- Mise à jour de la densité ***n*** des classes
- Calculer l'entropie ***Entp*** de la matrice ***G*** **.**
- Calculer ***w*** selon l'équation suivante :

w(c,k)= ***Entp*** (j,k) / ∑ (***Entp****(k)*) k=1…D; j=1…N ….….(6)

Où **c** est la classe correspondant au pixel **j** après une itération

Fin tant que

Etape 3 : classification : utilisant l'algorithme du centre de gravité pour la défuzzification et donc attribution de chaque pixel de l'image à une classe.

3. L'extraction d'attributs de textures

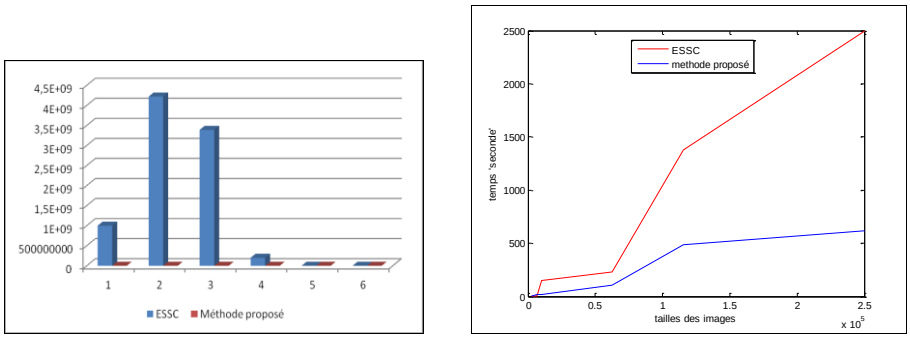
L'analyse de texture est très utile dans la vision par ordinateur, elle a plusieurs application dans la vie réelle, par exemple, l'analyse d'image médicale, l'analyse de document, l'analyse d'empreinte digitale, . . . Les textures différentes nous aident à distinguer différentes surfaces, en conséquence, elles facilitent la distinction des objets dans les images. Du fait de leur richesse en information de texture, nous avons opté pour les matrices de cooccurrence pour l'extraction des attributs. Ces dernières contiennent une masse très importante d'informations difficilement manipulable. C'est pour cela

qu'elles ne sont pas utilisées directement mais à travers des mesures dites caractéristiques de texture. En 1973, Haralick et al. [6] ont en proposé quatorze. Parmi les quatorze indices proposés par Haralick, nous avons choisi pour notre étude les plus utilisés, à savoir : le contraste, l'entropie, l'homogénéité, la corrélation, le second moment angulaire et la directivité. Nous citons les six paramètres considérés comme étant les plus utilisés et les plus pertinents et qui ont donné les meilleurs résultats : L'énergie qui mesure l'uniformité de la texture, le contraste est autant plus élevé que la texture est plus contrastée, l'entropie est un indicateur de désordre dans l'image. La corrélation décrit les corrélations entre les lignes et les colonnes de la matrice de cooccurrence ou de corrélogramme , la variance mesure l'hétérogénéité de la texture. Le moment inverse mesure l'homogénéité de l'image. Bien sûr, les contours constituent une caractéristique très importante que nous avons rajoutée. Ces sept caractéristiques sont les attributs sélectionnés, Les filtres sont aussi très utilisés pour extraire des caractéristiques de texture. L'algorithme ESSC [10] utilise le filtre Gabor, il a utilisé trente dimensions.

4. Résultats expérimentaux

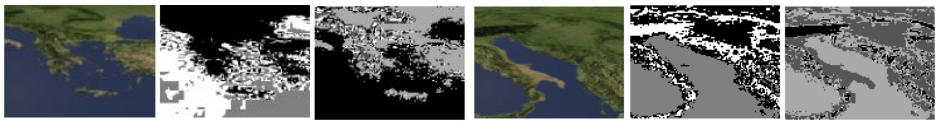
Nous avons calculé la compacité en utilisant la fonction proposée et celle exposée dans [10] sur différentes types d'images ayant différents textures. Les calculs ont été exécutés sur un pc CORE Duo CPU 2,1 GHz, en utilisant la plate forme de Matlab version 7. Les résultats sont présentés dans la figure .2.a qui montre clairement que la valeur de la compacité par la fonction proposée est plus faible que la fonction de l'algorithme ESSC pour différentes tailles d'images. Sachant qu'une petite valeur de ce terme signifie que les classes sont compactes, donc les classes obtenues après notre modification sont plus compactes. Les différentes images obtenues par l'implémentation des deux fonctions sont données par les figures 2 et 3. Dans la figure 2.b.1 et 2.b.2 une partie de la terre n'est pas présentée contrairement dans la figure 2.c.1 et 2.c.2. Dans la figure.3.c.3, il est facile de séparer la rate(5), le foie(2) l'estomac(7) et le pancréas (8) ; les cotes (1) sont bien détectées. Les contours sont bien limités, les poumons(6) sont clairs et homogènes, le liquide (4) autour de la rate est bien détecté. Il prend une classe différente de la rate. Dans cette image, d'après le médecin radiologue il est facile de tirer le diagnostic avec une précision élevée. Cependant dans la figure.3.b.3 on peut aussi séparer quelques organes comme la rate et l'estomac mais on remarque que le corps vertébral(3) ne prend pas sa vraie forme que sur la figure.3.c.3. Il est en contact avec le foie. Les poumons sont détectés mais ils ne sont pas homogènes. Le pancréas n'est pas détecté complètement ce qui signifie que le foie constitue une même classe avec la rate. Cette fausse anatomie obtenue peut créer de faux diagnostics comme elle peut cacher d'autres anomalies donc le médecin radiologue confirme que cette image ne peut

apporter aucune information concernant le diagnostic. Non seulement l'analyse visuelle de la classification nous a montré que notre méthode est la plus adaptée mais aussi les calculs peuvent juger facilement de la qualité de notre modification ainsi que de son avantage il a été démontré au dessus pour la valeur de la compacité. Le coût informatique aussi est l'un des facteurs les plus intéressants pour évaluer la qualité d'une méthode de classification sur une base de données contenant différents types d'images de différentes tailles. Nous avons calculé le temps d'exécution pour notre méthode et la méthode ESSC. Ces calculs sont présentés par la courbe donnée par la figure.1.b. Ces courbes montrent que le temps d'exécution augmente en fonction de la taille des images pour les deux méthodes mais avec une pente plus faible pour la méthode proposée.



a b

Figure 1. *a / compacité en fonction de la taille des images, b /temps d'exécution pour les deux méthodes.*



a .1 *b.1* *c .1* *a.2* *b.2* *c .2*

Figure 2.*a/image originale ;b / résultat utilisant ESSC ;c /résultat utilisant la méthode proposée.*

a.3 *b.3* *c.3*

Figure 3.*a/image originale ;b / résultat utilisant ESSC ;c /résultat utilisant la méthode proposée.*

5. Conclusion

Dans ce papier nous avons proposé une méthode de classification dans les sous espaces. La modification apportée dans la fonction objective produit de bons résultats, accompagnée de la réduction du coût de calcul. La densité des classes que nous avons introduite dans la fonction d'objective nous a permis de détecter différentes formes des classes. Le choix des attributs sélectionné est important pour les résultats de la classification. L'implémentation de cet algorithme a donné des résultats intéressants comparés à d'autres algorithmes.

6. Bibliographie

[1] Agrawal. R et al, 1998, *Automatic subspace clustering of high dimensional data for data mining applications*. In Proceedings of the 1998 ACM SIGMOD international conference on Management of data, ACM Press , pp 94-105.
[2] Aggarwal.C, et al ,1999, *Fast algorithms for projected clustering.* In Proceedings of the 1999 ACM SIGMOD international conference on Management of data, ACM Press, pp 61-72.
[3] Bouveyron .c, 2006, Modélisation et classification des données de grande dimension application à l'analyse d'images, thèse 2006 doctorat de l'université joseph fourier specialite : mathematiques appliquees.
[4]Ester. M, et all ,1996, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise ;* In Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR, pp 291-316.
[5]Haralick. R et al. 1973, *Textural features for image classification.* IEEE Transactions on Systems, Man and Cybernetics, 3(6),pp 610–621.
[6]Kailing. K, et al 2004, *Density-Connected Subspace Clustering for High-Dimensional Data* , In Proc. 4th SIAM Int. Conf. on Data Mining, pp 246-257.
[7] Parsons. L, et al, 2004. *Evaluating subspace clustering algorithms. In Workshop on Clustering High Dimensional Data and its Applications,* SIAM Int. Conf. on DataMining, pp 48–56.
[8]Sunita. J, Parag K, 2009, *Intelligent Subspace Clustering, A Density based Clustering approach for High Dimensional Dataset*, World Academy of Science, Engineering and Technology, 55, pp 69-73
[9] Widia Sembiring. R, Jasni. M, 2010, *Cluster Evaluation of Density Based subspace Clustering, journal of computing,* vol 2, issue 11, pp 2151-9617
[10] Zhaohong. D et al, 2010, *Enhanced soft subspace clustering integrating within-cluster and between-cluster information,* Pattern Recognition 43, pp 767–781