Intégration de contraintes dans les cartes auto-organisatrices

Anouar Benhassena*, Khalid Benabdeslem**, Fazia Bellal**, Alexandre Aussem** et Bruno Canitia***

* IRISA - Projet CORDIAL
6, rue de Kerampont - BP 447, 22305 Lannion Cedex, France
anouar.benhassena@gmail.com
**Université Lyon1, UFR d'Informatique, LIESP
8, Avenue Niels Bohr, 69622 Villeurbanne Cedex, France
{kbenabde, fbellal, aaussem}@bat710.univ-lyon1.fr
*** VISOON
60, Avenue de Rockefeller, 69008 Lyon, France
bruno.canitia@visoon.com

Résumé. Le travail présenté dans cet article décrit une nouvelle version des cartes topologiques que nous appelons CrTM. Cette version consiste à modifier l'algorithme de Kohonen de telle façon à ce qu'il contrôle les violations des contraintes lors de la construction de la topologie de la carte. Nous validons notre approche sur des données connues de la littérature en utilisant des contraintes *artificielles*. Une validation supplémentaire sera faite sur des données réelles issues d'images médicales pour la classification des mélanomes chez l'humain sous contraintes médicales.

1 Introduction

La prise en compte des connaissances additionnelles constitue un problème essentiel et un vrai défi pour la recherche actuelle dans le domaine de la classification automatique. Il s'agit à la fois de l'expression, de la structuration et de la formalisation des connaissances (appelées aussi connaissances *a priori*) pour les intégrer dans le processus de la classification automatique. Les premiers travaux dans ce domaine ont été réalisés par (Wagstaff et Cardie, 2000) en modifiant l'algorithme COBWEB proposé par (Fisher, 1987). Les auteurs ont montré, à partir de résultats expérimentaux, une amélioration claire de la précision de la classification. Les mêmes auteurs ont proposé une autre approche qui intègre les contraintes dans l'algorithme K-means (MacQueen, 1967). L'algorithme proposée est appelé COP-Kmeans (Wagstaff et al., 2001). Son principe consiste à contrôler la violation des contraintes dans la phase de mise à jour des classes. Les auteurs arrivent à démontrer qu'il est possible d'améliorer sensiblement la précision du partitionnement même avec un nombre réduit de contraintes. Les auteurs dans (Davidson et Ravi, 2005) ont étudié le problème de la faisabilité de la classification en présence de plusieurs combinaisons de contraintes dans une approche de type K-means. Récemment, nous avons proposé dans (Elghazel et al., 2007) une nouvelle méthode de classification sous contraintes basée sur la b-coloration de graphes. Convaincus par l'importance de l'intégration

des contraintes dans les méthodes de classification d'une part et de l'aspect de visualisation avec prise en compte de la notion de voisinage assurée par les cartes topologiques d'autre part, nous proposons dans cette article une nouvelle version de ces dites cartes, capable d'intégrer les connaissances *a priori* dans la construction de la topologie finale.

2 Cartes auto-organisatrices sous contraintes

2.1 Les contraintes

Dans le cadre des algorithmes de classification non supervisée, les contraintes liées aux observations peuvent être une manière très utile d'exprimer les connaissances *a priori* et donc indiquer quelles observations doivent ou non être regroupées ensemble. Par conséquent, nous considérons deux types de contraintes à la fois binaires et déterministes : (1) les contraintes positives notées $Con_{=}(z_i, z_j)$, spécifiant que deux observations z_i et z_j doivent être dans le même neurone. (2) les contraintes négatives notées $Con_{\neq}(z_i, z_j)$ spécifiant que deux observations z_i et z_j doivent être placées dans deux neurones distincts.

Les contraintes sur les observations définissent une relation binaire, positive et transitive entre les observations. Ce qui génere des contraintes *sous-entendues* apportées par le calcul de la fermeture transitive. Ces contraintes sont présentées sous la forme suivante :

- $Con_{=}(z_i, z_j)$ et $Con_{=}(z_j, z_k) \Rightarrow Con_{=}(z_i, z_k)$
- $Con_{=}(z_i, z_j)$, $Con_{=}(z_k, z_l)$ et $Con_{\neq}(z_i, z_k) \Rightarrow Con_{\neq}(z_i, z_l)$ et $Con_{\neq}(z_j, z_k)$

2.2 L'algorithme proposé

Notre contribution se situe dans les deux phases de l'étape itérative de l'algorithme de Kohonen (Kohonen, 1994) : la phase d'affectation (compétition) au cours de laquelle une forme z_i est affectée au neurone de la carte le plus proche au sens de la distance euclidienne et qui n'entraîne pas une violation de contraintes (L'ensemble de ces neurones est noté C_{Cond}), et la phase de minimisation (adaptation) qui consiste à mettre à jour les poids des neurones du voisinage du neurone vainqueur dans l'étape de compétition. Ces neurones ne doivent pas entraîner une violation de contraintes. La modification majeure apportée à l'algorithme de Kohonen est essentiellement dans la phase de compétition par l'incorporation de la fonction *ViolateCon(.)* (Algorithme 2). Cette fonction prend en entrée le neurone courant c, l'observation z_i présentée à l'apprentissage, l'ensemble des contraintes positives ($Con_{=}$) et l'ensemble des contraintes négatives ($Con_{\neq}$). Elle retourne en sortie l'ensemble des neurones C_{viol} qui entraînent une violation de contraintes avec l'observation z_i. D'une part, CrTM (Algorithme 1)parcourt les paires (z_i, z_j) satisfaisant les contraintes positives $Con_{=}$ et vérifie si z_j est affecté à un neurone différent de c, dans le cas échéant, il y a violation de contraintes et donc $c \in C_{viol}$ qui n'entrera pas en jeu dans le choix du neurone vainqueur. D'autre part, l'algorithme parcourt les paires (z_i, z_k) qui satisfont les contraintes négatives($Con_{\neq}$) et vérifie si z_k est affecté au neurone courant c, si oui, il y aura une violation de contraintes et donc $C_{viol} = \{c, V(c)\}$ tel que $V(c)$ représente les voisins de c. Cette vérification est nécessaire pour garantir que z_i sera affecté loin du neurone violant c et aussi pour que l'ensemble des neurones C_{viol} ne se rapprochent pas de z_i et donc ils ne l'apprennent pas en conséquence.

Algorithme 1 CrTM

ENTRÉES: A : Ensemble des données,
$Con_{=}, Con_{\neq}$: Ensembles des contraintes positives et négatives, respectivement.
SORTIES: ***CrTM*** : Carte Topologique satisfaisant les contraintes
Initialisation :
Pour t = 0, initialiser les poids w_j de tous les neurones de la carte C à des valeurs aléatoires et N_{iter} le nombre d'itérations.
présenter l'observation z_i choisie aléatoirement et faire :
Compétition :
Sélectionner les neurones qui ne violent pas les contraintes avec z_i :
pour tout neurone $c \in C$ **faire**
  $C_{viol} = ViolateCon(c, z_i, Con_{=}, Con_{\neq})$
  $C_{Cond} = C_{Cond} \cup (C/C_{viol})$
fin pour
Choix du gagnant $w_{c^*}^t$: $\|z_i - w_{c^*}^t\|^2 = min_{j \in C_{Cond}} \left\|z_i - w_j^t\right\|^2$
Adaptation :mise à jour des poids des neurones C_{Cond}

$$\begin{cases} w_j^t = w_j^{t-1} - \mu^t \mathcal{K}^T(\delta(j, c^*)) \left(w_j^{t-1} - z_i\right) & \text{Si } j \in C_{Cond} \\ w_j^t = w_j^{t-1} & \text{sinon} \end{cases}$$

Retour à 2 et répéter les deux étapes de compétition et d'adaptation jusqu'à atteindre N_{iter} ou une stabilisation.
μ^t représente un paramètre d'adaptation , appelé pas d'apprentissage et $T = T(t)$ représente le rayon de voisinage. Ces deux paramètres décroissent en fonction du temps t.
$\mathcal{K}^T(\delta(c, r)) = \exp(\frac{-0.5\delta(c,r)}{T(t)})$.

Algorithme 2 ViolateCon($c, z_i, Con_{=}, Con{\neq}$)

ENTRÉES: c : un neurone donné,
z_i : une observation donnée,
$Con_{=}, Con_{\neq}$: Ensembles des contraintes positives et négatives, respectivement.
SORTIES: C_{viol} : Ensemble des neurones entraînant une violation avec l'observation présentée z_i.
pour tout z_j tel que $Con_{=}(z_i, z_j)$ **faire**
  si $z_j \notin c$ **alors**
    C_{viol} = $\{c\}$
  finsi
fin pour
pour tout z_k tel que $Con_{\neq}(z_i, z_k)$ **faire**
  si $z_k \in c$ **alors**
    C_{viol} = $\{c, V(c)\}$ //$V(c)$: *Voisins de* c//
  finsi
fin pour
return C_{viol}

3 Résultats expérimentaux

3.1 Méthode d'évaluation

La méthode d'évaluation utilisée est basée sur l'indice de Rand (Rand, 1971). Cet indice représente une mesure d'accord entre deux partitions P_1, P_2 d'un même ensemble de données A. P_1 représente la partition correcte produite par les étiquettes des classes prédéfinies. P_2 représente la partition produite par l'algorithme CrTM. Chaque partition est vue comme un ensemble de $N(N-1)/2$ paires de décisions où N est la taille de A. Pour chaque paire d'observations z_i, z_j dans A, P_1 et P_2 les assignent à la même classe ou à deux classes différentes. Nous montrons aussi pour tester notre proposition que les informations apportées par les contraintes peuvent améliorer la performance de la classification même sur les observations qui n'ont pas été concernées par les contraintes ('Held-Out'). Cette dernière mesure représente le taux de bonne classification en ne considérant que les observations qui ne sont pas directement (ou par transitivité) concernées par les contraintes.

3.2 Résultats utilisant les contraintes artificielles

La validation expérimentale est réalisée par la génération des contraintes *artificielles*, c'est-à-dire pour générer une contrainte, nous prélevons aléatoirement deux observations de la base étiquetée et nous comparons leurs étiquettes : si elles ont la même étiquette, une contrainte positive est générée sinon une contrainte négative. Nous avons choisi trois bases de données de la banque UCI (Blake et Merz., 1998) : "Soybean", "Tic-tac-toe" and "Heart Disease". L'application de CrTM sur la première base a fournit une performance de 100% sans aucune contrainte. Ce résultat parfait est dû à la séparation linéaire des 4 classes dans l'espace des données, ce qui n'a pas été un souci pour l'apprentissage de la carte. Nous rappelons que COB-COBWEB et COP-Kmeans atteignent, respectivement, 96% et 98% après l'intégration de 100 contraintes artificielles. COP-b-coloring atteint la performance de 100% avec 30 contraintes. Vue la nature symbolique des variables de la deuxième base, nous avons procédé par codage disjonctif pour générer une base d'apprentissage numérique. CrTM commence avec un taux de 66.6 % sans contraintes. Il atteint une performance totale de 96,70% ("Overall Accuracy") et 91.40% sur le "Held-Out" avec 500 contraintes aléatoires (les résultats du "Held-Out" sont : 49% par COP-COBWEB, 56% par COP-Kmeans et 82% par COP-b-coloring avec le même nombre de contraintes). CrTM a donc permis une amélioration de 24% (Figure 1.b)

En l'absence de contraintes, l'algorithme de Kohonen atteint une performance de 78% sur la troisième base(Figure 1.a). Après l'intégration de 240 contraintes aléatoires, notre algorithme améliore de 13% la performance totale, avec un "Overall Accuraccy" de 91%. Le "Held-Out" quant à lui atteint les 85% avec seulement 150 contraintes intégrées, soit 7% d'amélioration avec seulement 5% d'information *a priori*. L'algorithme COP-b-coloring atteint une performance de 50% sans contraintes puis de 89% avec 500 contraintes et le "Held-Out" avoisine les 66%.

3.3 Application aux données de mélanomes chez l'humain

Nous avons testé l'algorithme CrTM sur une base médicale contenant 226 images de grain de beauté. Chaque image est caractérisée selon la règle ABCD (Stolz, 1994). l'Asymétrie de

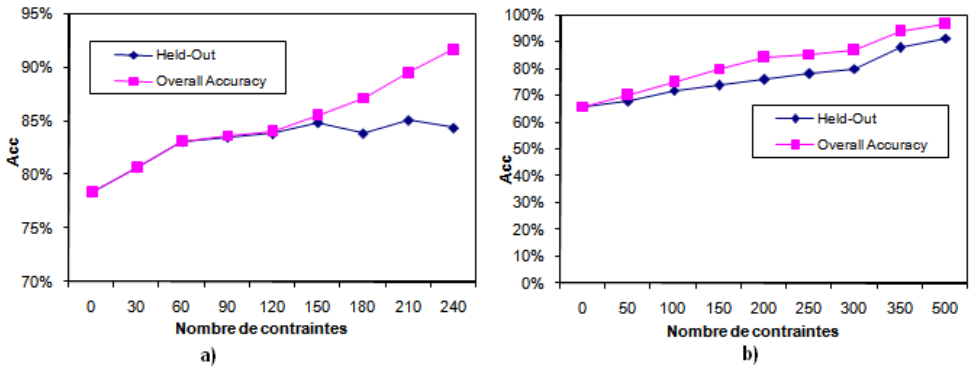

FIG. 1 – *Résultats de CrTM sur les bases "Cleve" (a) et "Tic-Tac-Toe (b)"*

formes, de textures et de couleurs, les Bords abrupts des structures pigmentées, la diversité des Couleurs, et différentes structures Dermatoscopiques. Les contraintes sont représentées par l'information *a priori* apportée par les diagnostics des dermatologues sous forme d'étiquettes associées aux images. Nous avons constaté que CrTM atteint une performance de 78,2% sans contraintes. "Overall Accuracy" augmente jusqu'à 84%, après l'intégration de 210 contraintes, soit un total de 3200 paires de décisions (y compris celles obtenues par transitivité). Ces 17% d'information *a priori* nous ont amélioré la performance de la classification, calculée par "Held Out", pour atteindre 81,2% (voir Figure 2).

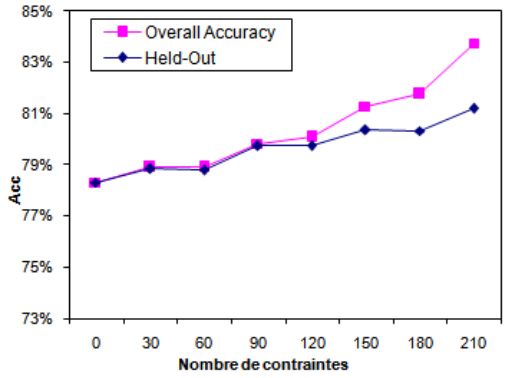

FIG. 2 – *Evaluation de la performance de CrTM sur la base des mélanomes*

4 Conclusion

Nous avons proposé dans cet article une extension de l'algorithme des cartes topologiques pour intégrer les contraintes liées aux données. Ces contraintes sont binaires (entre chaque

paire de données) et déterministes (appartenance ou pas à la même classe). Dans ce cadre, nous avons analysé les propriétés de l'algorithme de Kohonen (SOM) en lui apportant quelques modifications nécessaires pour l'adapter aux contraintes. Les résultats obtenues sont très encourageant aussi bien sur des bases "Benchmark" que sur une base réelle. D'une part, ce travail peut s'étendre au traitement des contraintes complexes (i.e. contraintes probabilistes , groupes de contraintes, contraintes conditionnelles) dans CrTM. D'autre part, nous visons à optimiser cet algorithme par une classification hiérarchique en trouvant un bon compromis entre la satisfaction des contraintes et le nombre de macro classes.

Références

Blake, C. et C. Merz. (1998). Uci repository of machine learning databases. Technical report, University of California.

Davidson, I. et S. S. Ravi (2005). Clustering with constraints: Feasibility issues and the k-means algorithm. In *SDM*.

Elghazel, H., K. Benabdeslem, et A. Dussauchoy (2007). Constrained graph b-coloring based clustering approach. In *DaWaK*, pp. 262–271.

Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning 2*, 139–172.

Kohonen, T. (1994). *Self-Organizing Map*. Berlin: Springer.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the the fifth symposium on Math, Statistics and Probability*, pp. 281–297.

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

Stolz, W. Riemann, A. e. a. (1994). Abcd rule of dermatoscopy: A new practical method for early recognition of malignant melanoma. *Eur J Dermatol 4*, 521–527.

Wagstaff, K. et C. Cardie (2000). Clustering with instance-level constraints. In *ICML*, pp. 1103–1110.

Wagstaff, K., C. Cardie, S. Rogers, et S. Schrödl (2001). Constrained k-means clustering with background knowledge. In *ICML*, pp. 577–584.

Summary

This paper describes a new topological map dedicated to clustering under constraints. In general, traditional clustering is used in an unsupervised manner. However, in some cases, background information about the problem domain is available or imposed in the form of constraints, in addition to data instances . In this context we demonstrate how the popular SOM algorithm can be modified to take these constraints into account during the construction of the topology. We present experiments on some known databases with generated artificial constraints. We also apply the new method to a real problem of clustering melanoma data in health domain.