CARTE TOPOLOGIQUE ET DONNEES BINAIRES

Mustapha. LEBBAH[a], Fouad. BADRAN[b], Sylvie. THIRIA[a,b]

a- CEDERIC, Conservatoire National des Arts et Mtiers,

292 rue Saint Martin, 75003 Paris, France

b- Laboratoire LODYC, case 100, Université Paris 6, Tour 26-4^{e}étage,

4 place Jussieu 75252 Paris cedex 05 France

1 Introduction

L'algorithme des cartes topologiques proposé par Kohonen [3] fait parties des méthodes dites d'auto-organisation. Le formalisme des nuées dynamiques fournit un cadre général permettant de formuler ce problème en un problème d'optimisation. Nous présentons dans cet article une nouvelle méthode neuronale de classification non supervisée basée sur les cartes auto-organisatrices, dédiée aux variables qualitatives. La méthode proposée recherche une classification automatique d'un nuage de points $App = \{(z_i, p_i), i = 1..I\}$ où l'individu $z_i = (z_i^1, z_i^2, ..., z_i^d)$ muni de la pondération p_i appartient àl'ensemble des données binaires $\beta^d = \{0, 1\}^d$. Cette méthode s'inspire de la version nuées dynamiques de l'algorithme de Kohonen [1] mais utilise pour déterminer l'ordre topologique un nouveau critère spécialement adapté au traitement des données binaires.

2 généralités sur les données binaires

Il existe de nombreuse variables, dites discrètes, ne pouvant prendre par nature qu'un nombre restreint de valeurs. Elle se repartissent en deux groupes : les variables qualitatives ordinales et les variables qualitatives nominales. Les codages utilisés le plus souvent pour avoir des variables binaires sont : (a)*Le codage binaire additif :* ce codage permet de rester cohérent avec la notion d'ordre entre les modalités d'une variable. (b)*Le codage disjonctif complet* (voir table 1).

L'espace des données binaires peut être muni de la distance euclidienne mais il est souvent plus intéressant de le munir de distances adaptées permettant de mieux traduire ses particularités [4]. Dans cet article nous utiliserons deux types de distance : la distance de Hamming appelé $\mathcal{H}$ et l'indice de Tanimoto appelé $\mathcal{T}$. Le calcul de dissimilarites entre deux individus z_1 et z_2 se fait alors à partir de la table de contingence établie à partir des vecteurs binaires qui leurs sont associés (voir table 2).

La distance binaire de Hamming entre z_1 et z_2 est le nombre de composantes différentes entre ces deux points, $\mathcal{H}(z_1, z_2) = b + c = |z_1 - z_2|$. L'indice de Tanimoto est égal à $\mathcal{T}(z_1, z_2) = \frac{a}{a+b+c}$.

Le nuage de points *App*, peut maintenant être caractérisé par sa caractéristique de valeur centrale associée à la distance de Hamming. Le nuage étant inclus dans l'espace β^d, il admet pour centre médian un point de ce même espace. Par définition le centre

Modalities	Codage Additif	Codage Disjonctif
1	1 0 0	1 0 0
2	1 1 0	0 1 0
3	1 1 1	0 0 1

TAB. 1 – Codage des Modalites

z_2 / z_1	1	0
1	a	b
0	c	d

TAB. 2 – Table de contingence

médian du nuage *App* est tout point $\omega^j = (\omega^1, \omega^2, ..., \omega^d)$ de β^d minimisant l'inertie du nuage définit par la distance de Hamming ,$\sum_{i=1}^{I} p_i \mathcal{H}(z_i, \omega)$, ce qui signifie que, pour tout j, ω^j minimise : $\sum_{i=1}^{I} p_i |z_i^j - \omega^j|$.

Les données étant binaires, ω^j est la médiane binaire de l'ensemble des valeurs prises par la variable j sur l'ensemble des individus. Lorsque les z_i sont munis d'une même pondération ($p_i = 1, \forall i$) la règle fournit une médiane ayant une interprétation particulièrement simple [5], ω^j est alors la valeur 0 ou 1 la plus souvent choisie par les individus sur la variable j.

3 Une carte topologique binaire

Nous montrons maintenant comment l'utilisation de la médiane peut permettre de définir un modèle de carte auto-organisatrice adapté aux données binaires. Comme pour le modèle classique des cartes topologiques, nous utilisons un réseau de neurones avec une couche d'entrée pour les entrées et une carte possédant un ordre topologique de k cellules. A chaque cellule c de la grille est associée un vecteur de poids binaire W_c de dimension p. Nous définissons la topologie de la carte C à l'aide d'un graphe non orienté et la distance $\delta(c, r)$ entre deux cellules c et r étant la longueur du chemin le plus court qui sépare la cellule c et r. Afin de modéliser la notion d'influence d'un neurone r sur un neurone c, qui dépend de leur proximité, on utilise comme fonction noyau $\mathcal{K}(\delta(c, r))$ la fonction indicatrice $\mathcal{K}(x) = \begin{cases} 1 & \text{si } x \leq \lambda(t) \\ 0 & \text{sinon} \end{cases}$, tel que $\lambda(t)$ contrôle le rayon du voisinage avec la fonction suivante : $\lambda(t) = \lambda_0 (\frac{\lambda_{max}}{\lambda_0})^{\frac{t}{t_{max}}}$,($\lambda_0$ est le voisinage initial et λ_{max} est le rayon du voisinage final au temps t_{max})

L'auto-organisation de la carte va maintenant se faire par l'intermédiaire de la minimisation d'une fonction de coût.

3.1 Minimisation d'une fonction d'énergie

Le processus de minimisation utilise la forme générale d'une fonction de coût [1]. Cette fonction sera adaptée aux données binaires comme suit :

$$\mathcal{E}(\mathcal{W}) = \sum_{z_i \in App} \sum_{r \in C} \mathcal{K}(\delta(c, r)) \mathcal{H}(z_i, W_r) \tag{1}$$

tel que c est le neurone affecté à l'exemple z_i par la fonction d'affectation Φ, $\Phi(z_i) = argmin_c \mathcal{H}(z_i, W_c)$. A chaque cellule c de la grille est associée une partie définie par : $P_c = \{z_i, \Phi(z_i) = c\}$. L'ensemble de toutes les parties constituent une partition notée $P_\Phi = \{P_c, c = 1..k\}$. Etant donné un voisinage noté $V_c = \{r, \mathcal{K}(\delta(c, r)) = 1\}$, on définit le recouvrement $R_c = \bigcup_{r \in V_c} P_r$. L'expression $\mathcal{E}(\mathcal{W})$ devient

$$\mathcal{E}(\mathcal{W}) = \sum_{r \in C} \sum_{z_i \in R_r} \mathcal{H}(z_i, W_r) \tag{2}$$

La minimisation de cette fonction qui va faire apparaître l'ordre topologique s'effectue en mettant en œvre un algorithme itératif basé sur les nuées dynamiques [2] appelé *Bin-Batch* qui fonctione en deux phase : une phase d'affectation qui associe a chaque exemple z_i le référent correspondant suivi d'une phase d'optimisation qui calcule pour chaque cellule de la carte le centre médian relatif à son recouvrement. Ces référents sont du même genre que les données initiales : Le décodage (additif ou exclusif) de différent vecteur permet l'interprétation symbolique des référents trouvés.

4 Exemple

4.1 Application de l'algorithme BinBatch

On realise une classification d'une base de données de 179 molécules en procédant à l'apprentissage d'un réseau de 7×13 neurones. Chaque molécule est décrite par un vecteur de dimension 988. Après 500 itérations, on obtient la carte de la figure 1.

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	6	10	2	8	1	7	4	1	2	2	2	0	2
1	3	3	3	0	1	1	2	0	0	0	1	0	2
2	14	0	0	4	2	1	3	3	0	0	2	1	0
3	0	1	2	0	1	0	3	4	0	2	0	1	1
4	4	0	2	4	3	2	0	0	4	0	3	1	0
5	3	5	0	5	1	3	1	2	0	0	3	1	1
6	4	0	6	1	0	2	2	2	1	3	0	1	1

FIG. 1 – la carte résultat du *BinBatch*. On affiche dans chaque neurone le nombre d'exemples (molécules) captés par celui-ci.

On remarque que les molécules sont bien réparties sur la carte. Il reste maintenant à voir l'homogénéité de ces partitions. On utilise un indice de similarité fréquemment utilisé en chimie, c'est l'indice de Tanimoto définit à la section 2. La moyenne de Tanimoto T_{moy} par rapport aux individus captés par un neurone c de cardinalité égale à n_c est calculée en utilisant la formule suivante :

$$\mathcal{T}_{moy} = \frac{2 \sum_{i=1}^{n_c} \sum_{j=1}^{n_c} \mathcal{T}(z_i, z_j)}{n_c(n_c - 1)} \tag{3}$$

Dans la littérature, chimiquement parlant, on considère que 2 molécules sont similaires lorsque l'indice de Tanimoto est proche ou supérieur à 0.85. La figure 2 représente la carte des moyenne de l'indice de Tanimoto.

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0.81	0.85	0.98	0.94	1.00	0.92	0.96	1.00	0.87	0.85	0.94	-	0.98
1	0.92	0.92	0.90	-	0.90	1.00	0.91	-	-	-	1.00	-	0.93
2	0.86	-	-	0.96	0.98	1.00	1.00	0.95	-	-	0.98	1.00	-
3	-	1.00	0.93	-	1.00	-	0.80	0.76	-	0.91	-	1.00	1.00
4	0.89	-	0.87	0.89	0.91	0.93	-	-	0.76	-	0.96	1.00	-
5	0.97	0.83	-	0.76	1.00	0.86	1.00	0.87	-	-	0.90	1.00	1.00
6	0.88	-	0.88	1.00	-	0.95	0.87	0.86	1.00	0.90	-	1.00	1.00

FIG. 2 – Moyenne de Tanimoto pour chaque neurone. On constate que la moyenne de l'indice de Tanimoto est proche ou supérieure à 0.85 pour la majorité des neurones à l'exception des molécules captées par les neurones (3,7) et (4,8).

L'intérêt de cette carte est de mettre en évidence une relation éventuelle entre la structure (empreinte) et les propriétés ou activités des molécules.

5 Conclusion

Les résultats de l'algorithme *BinBatch* sont très satisfaisants et prometteurs, en effet, sur l'exemple pour lequel on a appliqué l'algorithme, on a obtenu une très bonne représentation des relations entre les variables. Cet algorithme présente l'avantage de générer des référents du même type que les données initiales.

Références

[1] Anouar, F. Badran, F. Thiria, S.(1998). Probabilistic self-organizing map and radial basis function networks. Neurocomputing 20, 83-96.

[2] Diday, E. C,Simon (1996).*Clustering Analysis*, in :K.S. Fu(ED), Digital pattern recognition, Springer,New York.An Introduction to Symbolic Data.

[3] Kohonen, T (1994). *Self-Organizing Map*. Springer, Berlin

[4] Leich, F. Weingessel, A. Dimitriadou, E (1998). *Competitive Learning for Binary Data*. Proc of ICANN'98, septembre 2-4. Springer Verlag.

[5] Marchetti, F (1989). *Contribution à la classification de données binaires et qualitatives*, thèse de l'université de Metz.