ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 267-272

T opological Map for Binary Data

Mustapha. LEBBAH[a], Fouad. BADRAN[b], Sylvie. THIRIA[a,b]
a- CEDERIC, Conservatoire National des Arts et Métiers,
292 rue Saint Martin, 75003 Paris, France
b- Laboratoire LODYC, Université Paris 6, Tour 26-4^{e}étage,
4 place Jussieu 75252 Paris cedex 05 France

Abstract

We propose a new algorithm using topological map on binary data. The usual Euclidean distance is replaced by binary distance measures, which take into account possible asymmetries of binary data. The method is illustrated on an example taken from literature. Finally an application from chemistry is presented. We show the efficiency of the proposed method when applied to high-dimensinal binary data.

1 Introduction

The algorithm of topological maps proposed b y Kohonen [5] is a self-organization algorithm. The formalism of the dynamic clustering pro vides a general framework which leads to the batch version of the Kohonen's map. In this paper we present a new neuronal method of unsupervised clustering with self-organizing map dedicated to binary data. This method proposes a clustering of the individual data set $App = \{(z_i, p_i), i = 1..I\}$, where each individual $z_i = (z_i^1, z_i^2, ..., z_i^d)$ is a binary vector of $\beta^d = \{0,1\}^d$ and has a corresponding weight p_i. In our approach the dimension d can be very large (for example about 1000). The proposed learning algorithm is similar to the batch version of Kohonen, it uses the Hamming distance in order to determine the topological order. Hierarchical Clustering (HC) reduces the number of classes suggested by the topological map. W e also use an appropriate index of similarity to carry out HC, the so called Tanimoto index.

2 General information about a binary data

Very often binary vector is a coding of discrete features which have a finite, usually small, number of possible values. Some of these features, called ordinal variables, have an implicit order, the others are just nominal variables. The general coding used in order to obtain binary data are: (a) *The additive binary coding:* this coding respect the order existing between modalities. (b) *The disjunctive complete coding.* (see table 1).

Euclidean distance is not adapted to binary data, it is often much more interesting to use an appropriate similarity index [6, 3]. In this paper we use for the auto organization process two different similarity index: the Hamming

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 267-272

Modalities	Additive coding	Disjunctive coding
1	1 0 0	1 0 0
2	1 1 0	0 1 0
3	1 1 1	0 0 1

Table 1: Coding of modalities

z_2/z_1	1	0
1	a	b
0	c	d

Table 2: Contingency table

distance $\mathcal{H}$ and the index of Tanimoto $\mathcal{T}$. The Hamming distance between two vectors z_1 and z_2 is equal to: $\mathcal{H}(z_1, z_2) = \sum_{j=1}^{d} |z_1^j - z_2^j| = b + c$ where b and c are defined in table 2, it measures the number of mismatch between the two vectors z_1 and z_2, (a represents the number of coincident occurrence of 1 between the two vectors z_1 and z_2). The index of Tanimoto which is defined by $\mathcal{T}(z_1, z_2) = \frac{a}{a+b+c}$ represents the ratio of coincident occurrence of one to the number informative components of z_1 and z_2.

A set of individuals App in β^d can be caracterized by the central value, called the median center defined by the Hamming distance. The median center of binary vectors is itself a binary vector which has the same interpretation as the individuals. By definition the median center of App is any point $\omega = (\omega^1, \omega^2, ..., \omega^d)$ included in β^d minimizing the inertia of App: $\sum_{i=1}^{I} p_i \mathcal{H}(z_i, \omega)$. In this expression each component ω^j minimize $\sum_{i=1}^{I} p_i |z_i^j - \omega^j|$. When the weights are set to 1 ($p_i = 1, \forall i$) [7], ω^j can be easily computed, it is the value 0 or 1 most often chosen by the individuals on the variable j.

3 Binary Topological Map

Now we show how the use of the median can define a model of self-organizing map adapted to binary data. As for the traditional topological maps, we use a network of two layers (the input layer and the topological grid with k cells) and a neighbourhood function $\mathcal{K}(\delta(c, r))$ with maximum value centred at the winning unit c and becoming zero as the distance between c and neighbouring units r increases. In the following we define $\delta(c, r)$ as the length of the shortest path on the grid between the cells c and r and we take as neighbourhood function the smooth function: $\mathcal{K}(x) = \begin{cases} 1 & \text{if } x \leq \lambda(t) \\ 0 & \text{else} \end{cases}$ where $\lambda(t)$ controls the width of the neighbourhood with the time-decreasing function: $\lambda(t) = \lambda_0 (\frac{\lambda_0}{\lambda_f})^{\frac{t}{t_f}}$ (λ_0 is an initial width of the neighbourhood and λ_f is a final width of neighbourhoud at the final iteration t_f). Other smooth functions as $\mathcal{K}(x) = \exp(-\frac{x^2}{\lambda(t)^2})$ could be considered later. Each cell c of the grid is represented by a binary vector W_c of dimension d, $\mathcal{W}$ denotes the set of weights or referents associated to the neurons and C the set of neurons of the grid. We present in the next section the self-organizing process which uses a dedicated cost function.

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 267-272

3.1 Minimization of the cost function

The minimization process uses the general form of the cost function [1]. In the following this function will be set to:

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

Where the summation is taken on the neurons of the grid C. As mentioned in section 2, the cost function $\mathcal{E}(\mathcal{W})$ has to be adapted to binary data and the traditional Euclidean distance has been exchanged for the Hamming distance. In formula 1, c represents the neuron assigned to the example z_i by the assignment function Φ, $\Phi(z_i) = argmin_c \mathcal{H}(z_i, W_c)$. We introduce now some extra notations which allow to simplify formula 1:

- $P_c = \{z_i, \Phi(z_i) = c\}$ represents the set of individuals affected to a neuron c and $P_\Phi = \{P_c, c = 1..k\}$ the associated partition of App.
- $V_c = \{r, \mathcal{K}(\delta(c,r)) = 1\}$ represents the set of neurons which constitute the neighbourhood of c.
- $R_c = \bigcup_{r \in V_c} P_r$ represents the subset of App linked to cell c.

Using these notations the formula 1 becomes $\mathcal{E}(\mathcal{W}) = \sum_{r \in C} \sum_{z_i \in R_r} \mathcal{H}(z_i, W_r)$. The minimization of this function, which leads to the topological order, is made using dynamic clusters [2]. We develop an iterative batch algorithm ,named *BinBatch*, operating in two steps: an assignment step which assigns each observation z_i to one cell c using the assignment function, followed by an optimisation step which computes for each cell c the median center of R_c. The minimization of $\mathcal{E}(\mathcal{W})$ is run by iteratively performing the two steps until stabilization. At the end of the minimization, the referents which have the same code as the individuals (additive or disjunctive) can be decoded, allowing a symbolic interpretation of the topological map.

3.2 Compression of the number of classes by hierarchical clustering

At the end of the training, we choose to use a hierarchical clustering associated with Tanimoto index defined in paragraph 2. The similarity between two subset $\mathcal{A}$ and $\mathcal{B}$ of C ($\mathcal{R}_\mathcal{A} = \{z_i, z_i \in P_c, c \in \mathcal{A}\}$, $\mathcal{R}_\mathcal{B} = \{z_i, z_i \in P_c, c \in \mathcal{B}\}$) is defined as follows:

$$\Delta(\mathcal{R}_\mathcal{A}, \mathcal{R}_\mathcal{B}) = \frac{1}{card(\mathcal{R}_\mathcal{A}) card(\mathcal{R}_\mathcal{B})} \sum_{z_a \in \mathcal{R}_\mathcal{A}} \sum_{z_b \in \mathcal{R}_\mathcal{B}} \mathcal{T}(z_a, z_b) \quad (2)$$

This index allows to understand the order of the different referents and to appreciate the quality of the topological grid.

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 267-272

4 Example

4.1 Application of BinBatc h algorithm on dog database

This example is taken from [8]. The data consist in characterization of 27 races of dogs by the 7 following variables: **S**ize(**S**mall, **A**verage, **B**ig), **W**eight (**S**mall, **A**verage, **B**ig), **V**elocity (**S**mall, **A**verage, **B**ig), **I**ntelligence (**S**mall, **A**verage, **B**ig), **A**ffection (**A**fectionate, **N**on-**A**ffectionate), **A**ggressiveness(**A**gressive, **N**on-**A**gressive), **F**unction (**A**ssistance, **H**unting, **C**ompany).

0 SS,SW,SV,AF, NAG,CH Poodle, Chihuahua Pekinese, Dachshund	**1** SS,SW,SV, AI,AF,AG,CM Bull Dog, Cocker Fox-Terrier	**2** AS,AW,AV,AI,AF NAG,CM Boxer, Collie Dalmatien	**3** AS,AW,AV, AI,AF,NAG,H Labrador	**4** AS,AW,AV, BI,AF,NAG,H Breton Spaniel
5	**6**	**7**	**8**	**9** BS,AW,SI,NAF, AG,H Fox-Hound, Gascogne
10 BS,BW,SV,AI, NAF,NAG,A Newfoundland	**11**	**12** BS,AW,BV,BI, AF,AG,A Beauceron, Alsatian Doberman	**13**	**14** SS,SW,SV,SI, NAF,AG,H Basset
15 BS,BW,SV,NAF, AG,A Bull-Mastiff Saint Bernard	**16**	**17**	**18** BS,AW,BV, NAF,NAG,H Greyhound Pointer, Setter	**19**
20 BS,BW,SV,SI, NAF,AG,A Mastiff	**21** BS,BW,BV,SI NAF,AG,A German Dog	**22**	**23** BS,AW,AV,AI NAF,NAG,H French Spaniel	**24**

Table 3: Learning done with BinBatch algorithm. Each cell of the table is a neuron of the grid. The first two lines of a neuron c presents the number of this neuron and the referent W_c given by *BinBatch* followed by the set P_c of individuals collected by it.

Table 3 presents the grid obtained at the end of the *BinBatch* algorithm. W e see that the dogs with **S**mall **S**ize, **S**mall **W**eight, **S**mall **V**elocity, **A**fectionate and **C**ompany belong to neighbouring neurons in the left corner of the map. The only difference between the two neurons being that the dogs collected by neuron 1 (Bulldog, Cocker, Fox-Terrier) are **A**ggressive compared to those of neuron 0 (Poodle,Chihuahua, Pekinese, Dachshund) which are **N**ot **A**gressive. We can make the same analysis for the remaining clusters.

The same dog database was used with another method named Kohonen-ACM (KACM) [4] which uses Euclidean distance to calculate the similarity between the individuals. W e compared the grid obtained by *BinBatch* with the grid obtained using KACM. The clusters are similar but *BinBatch* provides referents which have a symbolic interpretation. This characteristic is not always verified when using KACM method.

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 267-272

4.2 Application of hierarchical clustering (HC)

We carry out hierarchical clustering to the topological grid given by *BinBatch*, using formula 2. Figure 1 presents the dendrogram corresponding to the HC.

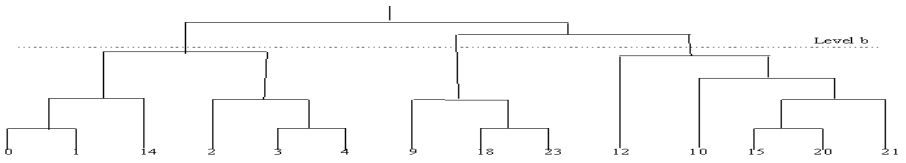

Figure 1: Hierarchical clustering applied to the grid represented in table 3. Each leaf of the tree represents a neuron of the grid which collects an individuals

We compare our results with the results provided by the multiple correspondence analysis [8]. In this example, Tenenhaus uses MCA in order to explain the link between the first six qualitative variables and the seventh variable. Clearly, the three clusters obtained by HC if we cut off a tree in level b (see figure 1) correspond to the three clusters given by MCA (see figure 2). The results given by MCA is one of the possible solution provided by HC and *BinBatch*.

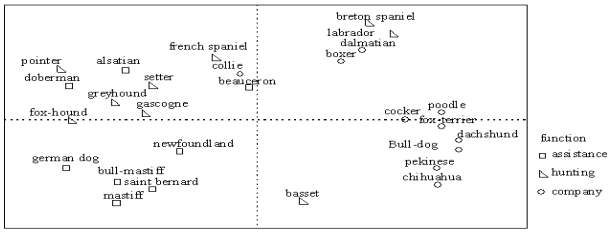

Figure 2: Representation of three clusters (assistance, hunting, company)

4.3 Application of BinBatch on high-dimensional vectors: an application in chemistry

BinBatch is well-suited in order deals with such dimension. Molecular databases have to face the problem of high-dimensional features. A molecule belonging to a molecular databases is coded by 988 binary data. We train *BinBatch* on a databases with a grid of 7 x 13 neurons. For each neuron c we compute the means Tanimoto index T_{P_c} from the subset P_c using this formula: $T_{P_c} = \frac{2\sum_{i=1}^{n_c}\sum_{j=1}^{n_c}\mathcal{T}(z_i,z_j)}{n_c(n_c-1)}$ (n_c is a number of molecules z_i collected by neuron c). If the value of Tanimoto is near 1 the molecules share the same chemical proprieties. Figure 3 presents the means Tanimoto for each neurons. All the value of the grid are near or greater than 0.85. So *BinBatch* provides homogeneous molecule clusters.

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 267-272

	0	1	2	3	4	5	6	6	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

Figure 3: The means of Tanimoto for each neuron

5 Conclusion

The results of *BinBatch* are very satisfactory and promising. We quickly obtained a very good representation of the input data. This algorithm presents advantage to have referents with the same coding as the initial data. The use of the HC allows us to limit the number of cluster.

Acknowledgement: A Part of this work was made in L'ORÉAL-research-laboratory in France. The authors would like to acknowledge Nadia DAMI and Roger ROZOT from L'ORÉAL for providing us molecular databases, and making useful comments about the results. The authors would like to thanks Younes Bennani from LIPN for his constructive comments and suggestions.

References

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

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

[3] Dolinicar, Weingessel , A. Buchta, C. Dimitriadou, E.*A Comparaison of several cluster algorithms on artificial binary data, scenarios from travel market segmentation.* Working paper series 19, SFB (adaptive information systems and modelling in economics and managemen t science). (1998).

[4] Ibbou, S. Cottrell. *Multiple correspondense Analysis crosstabulation matrix using the Kohonen algorithm.* In verlaeysen, M. Editor proc of ESANN'95, pages 27-32. Dfacto Bruxelles. (1995).

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

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

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

[8] Tenenhaus, M. *La régression PLS, théorie et pratique.* Edition Technip. (1998).