ESANN'2007 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2007, d-side publi., ISBN 2-930307-07-2.

Clustering a medieval social network by SOM using a kernel based distance measure

Nathalie Villa[1] and Romain Boulet[2] *

1- Laboratoire de Statistique et Probabilités - Université Toulouse III
118 route de Narbonne, 31062 Toulouse cedex 9 - France

2- Institut Mathématique de Toulouse - Université Toulouse II
5 allées Antonio Machado, 31058 Toulouse cedex 9 - France

Abstract. In order to explore the social organization of a medieval peasant community before the Hundred Years' War, we propose the use of an adaptation of the well-known Kohonen Self Organizing Map to dissimilarity data. In this paper, the algorithm is used with a distance based on a kernel which allows the choice of a smoothing parameter to control the importance of local or global proximities.

1 Introduction

Social networks have been intensively studied through graphs during the past years: examples of such studies are given in [1] for World Wide Web, scientific networks or P2P networks. Most of these graphs come from modern social networks whereas we propose here to analyse the social organization of a medieval peasant community before the Hundred Years' War. A first study ([2]) investigates this problem by the use of the algebraic properties of a non-weighted graph. We propose here a new approach, using an automatic neuronal method and, more precisely, one of the numerous adaptations of the Kohonen Self Organizing Map (SOM). The SOM algorithm, first introduced by Kohonen ([3]), is an unsupervised method which allows both clustering and visualization. The original data, usually living in a high dimensional space, are projected non linearly in a low dimensional space (generally, the projection dimension is set to 1 or 2) called a map; they are partionned into several clusters while preserving their initial topology. The need for adaptations of classical data analysis methods to non vectorial data is important: [4], for example, described neural based unsupervised classification methods for structured data such as text and trees. But SOM has also recently been adapted to a more general type of data, only described by a dissimilarity measure: this *Dissimilarity SOM* (or median SOM) has been first described in [5]; a variant of it has been introduced and used for Web Usage Mining in [6] and a faster version is then described in [7]. As the data set has to be described by its dissimilarity measure, the choice of a good dissimilarity is critical: we propose here to investigate the combination of the algorithm described in [6], using a distance defined on a weighted graph by the diffusion kernel ([8]).

*This project is supported by "ANR Non Thématique 2005 : Graphes-Comp".

The paper is organized as follows. In section 2, we recall the Dissimilarity SOM algorithm (section 2.1) and describe how distances based on a kernel can be used to produce an unsupervised classification algorithm for weighted graphs (section 2.2). In section 3, we focus on the medieval data set: after describing it, we explain how we apply our method (section 3.1) and we compare this classification with algebraic or historical prior knowledge (section 3.2): some similarities prove that the results are consistent with previous work.

2 Relevant theory

2.1 SOM for dissimilarity data

Following [6], let us consider n input data, $(x_i)_{i=1,\ldots,n}$, from an arbitrary input space, $\mathcal{G}$ (here $\mathcal{G}$ is a weighted graph and the x_i are the vertices of the graph). Suppose that we are given a dissimilarity between those data; we will note $\delta(x_i, x_j)$ the value of this measure between data x_i and x_j: δ is symmetric, positive and for all $i = 1, \ldots, n$, $\delta(x_i, x_i) = 0$. As the standard SOM, input data are mapped into a low dimensional space of M neurons, called a map. The neurons $\{1, \ldots, M\}$ are arranged via a prior structure (like a grid) which provides a neighborhood relationship between them; we will note $h(i, j)$ the distance between neurons i and j on the grid. Finally, we are given a decreasing function, R, from $\mathbb{R}^+$ to $\mathbb{R}^+$ with $R(0) = 1$ and $\lim_{s\rightarrow+\infty} R(s) = 0$. By the calculation of $R(h(i, j))$, this function defines how the neighborhood relationship has to be taken into account during the training. R is adapted during the training in order to favour the closest neurons more and more.

At the begining of the algorithm, each neuron i is randomly associated with an element of the data set, its prototype, denoted by m_i. The Dissimilarity SOM is then divided into two steps:

1. the **Affectation step**: $\forall\, i = 1, \ldots, n$, x_i is affected to the neuron $f(x_i) \in \{1, \ldots, M\}$ such that: $f(x_i) = \arg\min_{j=1,\ldots,M} \delta(x_i, m_j)$;

2. the **Representation step**: $\forall\, j = 1, \ldots, M$, the prototypes are re-computed: $m_j = \arg\min_{k=1,\ldots,n} \sum_{i=1}^n R(h(f(x_i), j))\delta(x_i, x_k)$.

These two steps are repeated iteratively until a stabilization is observed. As this iterative algorithm uses, at each step, the whole data set for the optimization procedure, it is closely related to the batch version of the classical SOM algorithm (see [3]).

In [6], the authors use the affinity similarity or the Jacard coefficient to compute a dissimilarity measure on the usage of a web site. In the next section, we propose to use a distance computed from a kernel which is able to take into account both the local and global structures of a graph.

2.2 Using a kernel

We will now precise how a dissimilarity measure, based on a kernel, can be constructed and applied with the previous algorithm. First of all, let us describe more precisely the data set: as we have already said, $\mathcal{G}$ is a graph with a set of vertices $V = \{x_1, \ldots, x_n\}$ and a set of edges, E, having positive weights, $(w_{i,j})_{i,j=1,\ldots,n}$ such that for all $i, j = 1, \ldots, n$, $w_{i,j} = w_{j,i}$ and for all $i = 1, \ldots, n$, $\sum_{j=1}^{n} w_{i,j} = d_i$ where d_i is the degree of the vertex x_i ($w_{i,j} = 0$ is then equivalent to the fact that $(x_i, x_j) \notin E$).

In [9], the authors investigate a family of kernels on graphs based on the notion of regularization operators. These kernels generalize the Green function (which is of common use for real valued functions in machine learning community) and are all based on the *Laplacian* of a graph which is the positive matrix $L = (L_{i,j})_{i,j=1,\ldots,n}$ such that $L_{i,j} = \left\{\begin{array}{ll} -w_{i,j} & \text{if } i \neq j \\ d_i & \text{if } i = j \end{array}\right.$ (see [10] for an entire review of the properties of this operator).

Applying regularization functions to the discrete Laplacian, we obtain a family of matrices that are also kernels on $V \times V$: these kernels are operators that strongly penalize the functions defined on V that vary a lot on close neighborhood. Some of them have became famous: the regularized Laplacian, the diffusion kernel, the von Neumann diffusion kernel (see [9] for further details). All these kernels are symmetric and positive definite and can then be interpreted as inner products of an unknown feature space. It is then easy to define a dissimilarity measure on the graph as the distance in this feature space: for all $x, x' \in V$, $\delta(x, x') = \sqrt{k(x,x) + k(x',x') - 2k(x,x')}$ where k denotes the kernel. In [11], the authors compare several such distances in order to rank the nodes of a weighted graph. They show the good performances of these methods in comparison with standard ones.

In this paper, we use the dissimilarity measure built from the diffusion kernel: $K^\beta = e^{-\beta L}$ is the diffusion matrix and for all $i, j = 1 \ldots, n$, $k^\beta(x_i, x_j) = K^\beta_{i,j}$ defines the diffusion kernel. First studied by [8], the diffusion kernel is the discrete version of the solution of heat equation. It can also be interpreted as the discrete version of the gaussian kernel or as the continuous time limit of a lazy random walk on graph $\mathcal{G}$. It is easy to compute for graphs having less than a few hundred vertices thanks to standard eigenvectors decompositions (see [12], chapter 8). This makes it an attractive tool that becomes very popular in the computational biology area where it has been used with success to extract pathway activity from gene expression data through a graph of gene (see [12], [13]). Here, we propose to use the dissimilarity measure endoved by the diffusion kernel with Dissimilarity SOM: this approach is tested on the medieval social network for a clustering purpose.

3 Application

The graph on which we tested our approach has been obtained from a data base of approximately 10 000 agrarian contracts from four seignories of the Lot and the Tarn-et-Garonne (South West of France), established between 1240 and 1520. In this paper, we focus on a part of the data base, based at the Castelnau-Montratier seignory (Lot) between 1240 and 1350 (before the Hundred Years'

War). Based on this data base, we constructed a weighted graph having 226 vertices (the peasants) which are linked together if they appeared in the same contract. The weights were simply the number of common contracts in which two peasants appeared together. We cleaned the graph by deleting the nobilities because they were mentionned in almost every contract (as the legal authorities). Historians are mainly concerned with the analysis of country sociability during the Middle-Ages; for example, they want to know if geographical links or family links are the most important, if we can find central people having a main social role and who are these people, ... In our case, the data base is too large for an exhaustive study so that data mining tools are required.

3.1 Methodology and results

We used the methodology described in the previous section to extract social tendencies in the medieval graph. More precisely, we computed a Dissimilarity SOM with an hexagonal grid which is close to the star structure of the non-weighted graph found by [2]. The dimension of the grid was set to 3 × 3 as the number of vertices to be clustered is not very high (226); moreover, the non-weighted graph diameter is only 5 and 90 % of the vertices couples have a distance which is smaller than 3 (see [2]), so that a bigger grid would be inappropriate. We used for R a gaussian kernel which was weighted by a linearly decreasing function of the number of steps. Finally, the algorithm was run until the stabilization of the classification was obtained.

One of the main points of the methodology is that a flexibility is allowed, in the classification, by the choice of the kernel parameter, β. The smaller β is, the more prefered the local proximities are. We chose to explore three different values of this kernel: 0.05, 0.1 and 0.2. Larger parameters lead to numerical instability and smaller ones failed to provide an interesting classification: a large class gathered almost every vertices whereas the other classes were really tiny (less than 5 vertices each). In order to minimize the influence of the initialization step, we perform the algorithm 50 times for each parameter β; the best classification is the one which minimizes the map energy $\mathcal{E} = \sum_{j=1}^{M} \sum_{i=1}^{n} R(h(f(x_i), j))\delta(x_i, m_j)$ obtained at the end of the algorithm. The map energies of the classifications constructed by two different dissimilarities are, of course, not comparable so that we finally obtained three classifications, one for each selected parameter β.

We observed many common points between these three classifications, especially with the ones obtained by the two smallest parameters: 5 prototypes were similar and 7 classes were easily recognizable between the 3 classifications. We provide in Figure 1, a synthesis map, quoting the 7 classes with vertices (peasants which are denoted by a number) appearing in two or sometimes all of the three classifications. Common prototypes are emphasized by bold type and positions in the map are determined by a kind of “majority vote” law (the 4 black classes are the easiest to place as they have common positions in two classifications on three).

Class 6 31, 35, 69, 70, 79, 118, 155, 156, 159 161, 175, 176, 198, 246, 249, 278 281, 284, 285, 291, 316, 352, 354 384, 389, 405, 432, 433, 434, 435 442, 447, 449		**Class 7** 3, 61, 64, 74, 191, 204, 220, 228, 230 232, 233, 236, 247, 255, 257, 259 272, 283, 376, 378, **412**
Class 4 8, 10, 119, **127**, 128, 131, 134, 135 153, 181, 193, 200, 237, 258, 264 276, 377, 386, 390, 416, 420	**Class 3** 133,**139**, 150, 262	**Class 2** 9, 67, 129, 130, **136**, 137, 140, 141 146, 148, 151, 152, 217, 260, 263 413
	Class 5 6, 30, 37, 72, 120, 197, 226, 269, 274 282, 357, 379, 399, 421	**Class 1** 2, 22, 23, 29, 34, 36, 51, 52, 71, 73 147, 192, 199, 202, 203, 207, 216 218, 219, 221, 223, 224, **225**, 229 231, 238, 248, 250, 252, 265, 275 279, 315, 356, 358, 366, 367, 369 371, 383, 395, 401, 403, 410, 417 440, 443, 444

Fig. 1: Synthesis map of three classifications

3.2 Comparison with previous works

In this section, we give some elements of comparison with previous work: on one hand, the historical knowledge ([14]) and on the other hand, an algebraic study of the non-weighted graph which is partially exposed in [2]. This comparison emphasizes some interesting common points.

For example, historians can easily explain some of the classes: Class 1 and 2 have homogeneous geographical settings. A majority of the peasants of class 1 comes from "Castrum de Flaugnac" and a majority of the peasants of class 2 comes from "Castrum de la Graulière". Some exceptions interest the historians: "Combelcau family", for example, is not in the class 1 although some of them live in "Castrum de Flaugnac" but "Combelcau family" is also present in "Castrum de la Graulière" and constitutes a large part of class 2. This emphasizes the fact that family links are more important than geographical ones. Moreover, class 3 is also a "Combelcau family" class which shows, by the central situation of this class on the map, that this family played a great social role at this time.

The comparison with algebraic study is also interesting: this method divides the graph into several *communities* which are (not maximal) cliques having the same neighbors. Every class found by the dissimilarity SOM corresponds to one or several connected communities. Moreover, the central class (class 3) is a part of a "rich club" (see [15]) which is a small number of vertices having large numbers of links and being very well connected to each other. The algebraic study of the non-weighted graph leads to a star-shaped structure around this rich club in which the "Combelcau family" plays a very important role; the dissimilarity SOM provides an organization of the weighted graph which is close to the social relations found by the algebraic study.

4 Conclusion

We demonstrate that a kernel based distance can provide a coherent clustering of a weighted graph by the use of Dissimilarity SOM. Moreover, the choice of parameter provides more flexibility and allows to extract the most important tendencies in the graph structure. The differences between this approach and algebraic or historical ones have now to be further investigated on the whole data set to explore the social organization of this medieval peasant community deeply.

References

[1] S. Bornholdt and H.G. Schuster. *Handbook of Graphs and Networks - From the Genome to the Internet.* Wiley-VCH, Berlin, 2002.

[2] R. Boulet and B. Jouve. Partitionnement d'un réseau de sociabilité à fort coefficient de clustering. In *7èmes Journées Francophones "Extraction et Gestion des Connaissances"*, pages 569–574, 2007.

[3] T. Kohonen. Self-Organizing Maps, 3rd Edition. In *Springer Series in Information Sciences*, volume 30. Springer, Berlin, Heidelberg, New York, 2001.

[4] B. Hammer, A. Micheli, M. Strickert, and A. Sperduti. A general framework for unsupervised processing of structured data. *Neurocomputing*, 57:3–35, 2004.

[5] T. Kohohen and P.J. Somervuo. Self-Organizing maps of symbol strings. *Neurocomputing*, 21:19–30, 1998.

[6] A. El Golli, F. Rossi, B. Conan-Guez, and Y. Lechevallier. Une adaptation des cartes auto-organisatrices pour des données décrites par un tableau de dissimilarités. *Revue de Statistique Appliquée*, LIV(3):33–64, 2006.

[7] B. Conan-Guez, F. Rossi, and A. El Golli. Fast Algorithm and Implementation of Dissimilarity Self-Organizing Maps. *Neural Networks*, 19(6-7):855–863, 2006.

[8] R.I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th International Conference on Machine Learning*, pages 315–322, 2002.

[9] A.J. Smola and Kondor R. Kernels and Regularization on Graphs. In M. Warmuth and B. Schölkopf, editors, *Proceedings of the Conference on Learning Theory (COLT) and Kernel Workshop*, 2003.

[10] F. Chung. *Spectral Graph Theory.* Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society, 1997.

[11] F. Fouss, L. Yen, A. Pirotte, and M. Saerens. An experimental investigation of graph kernels on a collaborative recommendation task. In *IEEE International Conference on Data Mining (ICDM)*, pages 863–868, 2006.

[12] B. Schölkopf, K. Tsuda, and J.P. Vert. *Kernel methods in computational biology.* MIT Press, London, 2004.

[13] J.P. Vert and M. Kanehisa. Extracting active pathways from gene expression data. *Bioinformatics*, 19:238ii–244ii, 2003.

[14] F. Hautefeuille. Espace juridique, espace réel : l'exemple de la châtellenie de Castelnau-Montratier (Lot) aux XIIIe et XIVe siècles. In *Habitats et territoires du sud, 126e congrès national des sociétés historiques et scientifiques*, 2001.

[15] S. Zhou and R.J. Mondragon. The rich-club phenomenon in the Internet topology. *IEEE Communications Letters*, 8(3):180–182, 2004.