Mining a medieval social network by kernel SOM and related methods

Nathalie Villa, Fabrice Rossi, Quoc-Dinh Truong

HAL Id: hal-00278196

https://hal.archives-ouvertes.fr/hal-00278196

Submitted on 9 May 2008

Mining a medieval social network by kernel SOM and related methods

Nathalie Villa[1,2], Fabrice Rossi[3] & Quoc Dinh Truong[4]

[1]IUT de Carcassonne, Université de Perpignan, Perpignan. France

[2]Institut de Mathématiques de Toulouse, Université de Toulouse, Toulouse, France

nathalie.villa@math.univ-toulouse.fr

[3]INRIA, Projet AxIS, Rocquencourt, France

fabrice.rossi@inria.fr

[4]IRIT, Université de Toulouse, Toulouse, France

truong@univ-tlse2.fr

Abstract - *This paper briefly presents several ways to understand the organization of a large social network (several hundreds of persons). We compare approaches coming from data mining for clustering the vertices of a graph (spectral clustering, self-organizing algorithms...) and provide methods for representing the graph from these analysis. All these methods are illustrated on a medieval social network and the way they can help to understand its organization is underlined.*

Key words - social network, large graphs, SOM algorithm, graph drawing, clustering, spectral clustering, heat kernel

1 Introduction

A large number of practical applications can be modeled through what is commonly called a "complex network". Complex networks are relational data, that appear in World Wide Web studies, in social networks or in biological studies (genes, proteins, metabolites interaction networks) for example.

This work is based on a historical database built from the archives of Lot, a small region in South West of France. This database has already been presented in [2]: in a tiny geographical location around Castelnau-Montratier, a large documentation has been collected (see [9] for a complete presentation). This documentation, made of about 1000 agrarian contracts (available at `http://graphcomp.univ-tlse2.fr`) is a very precious source of information about the peasants' usual life in the middle ages where most of the written documents were concerned by the well-educated part of the population. All the contracts are agrarian transactions: they mention the name of the involved peasant (or the peasants), the names of the lord and the notary to whom the peasants are related, some of the neighbors of the peasants and various other informations (such as the type of transaction, the location, the date, and so on). All the studied transactions were written between 1260 and 1340 that is, just before the Hundred Years' War but others concerned the period just after this War.

From this database, a relational network is built following the advices provided by the historians (see [4]). This social network is described by a *weighted graph* with 615 vertices (the

peasants) and 4193 edges standing for the relations between them. The edges are weighted by the number of relations found between two given peasants. The obtained graph is described in [4] where it is analyzed through the comparison of an algebraic study and of a SOM algorithm. The collaboration between mathematicians, computer scientists and historians intends to provide several tools for historians to understand this complex and large network. A part of the methods developed are coming from statistics and data mining and will be reviewed and illustrated in this paper. Complementary material could be found in [4] and complementary studies of the database are available in [3, 2].

The paper is organized as follows: Section 2 presents the problem of clustering the vertices of a large graph and explains how this problem can help to understand the structure of the graph. Several methods are reviewed and some of them, coming from what is called *spectral clustering*, are illustrated on the medieval database. Section 3 explains how this clustering can be used to provide a simplified representation of the graph. This leads us to use organization algorithms designed for graphs in order to classify and organize simultaneously the vertices of the graph: kernel SOM, described in section 4, targets such a dual objective. Finally, Section 5 intends to represent the whole graph from this final organizing map. Examples of insights on the data obtained by the reviewed methods are given at each step of the analysis.

2   Clustering the vertices of the graph

Large graphs representing complex networks are not easy to understand. One way to simplify them, in order to underline the main tendances of their structure, is to find dense subgraphs that have few connections to each others. As emphasized by [13],

> *reducing [the] level of complexity [of a network] to one that can be interpreted readily by the human eye, will be invaluable in helping us to understand the large-scale structure of these new network data.*

But such a clustering of the vertices of a graph is facing the problem of relational data: there is no *a priori* distance between two vertices and thus classical clustering algorithms, such as, e.g., k-means, cannot be directly used. A recent survey on clustering methods adapted to graphs is provided in [14]. Clustering the vertices of a graph is commonly addressed by the use of a dissimilarity between vertices or by mapping the graph on an euclidean space; then usual data mining tools can be used to find a convenient clustering. Recently, *spectral clustering* became a successful method among this kind of methodologies (see [24] for a very exhaustive tutorial on this subject): spectral clustering uses the properties of the *Laplacian* of the graph to understand its structure. Given a weighted graph $\mathcal{G}$ with vertices $V = \{x_1 \ldots, x_n\}$ and edges weighted by $(w_{i,j})_{i,j=1\ldots,n}$ ($w_{i,j} = w_{j,i}$ and $w_{i,i} = 0$), the Laplacian is the matrix L such that $L_{i,j} = \begin{cases} -w_{i,j} & \text{if } i \neq j, \\ d_i = \sum_{j=1}^{n} w_{i,j} & \text{if } i = j. \end{cases}$ The Laplacian appears as a very convenient tool for understanding the graph as its eigenvalue decomposition is directly related to the min cut problem ("How to find a partition of the vertices that minimizes the number of cuts in the graph ?", see [24]) and to the problem of finding perfect communities, i.e., complete subgraphs those vertices have exactly the same neighbors (see [4, 21]). More precisely, spectral clustering uses the eigenvectors associated with the smallest eigenvalues of the Laplacian to map the graph on an Euclidean space where a k-means algorithm is performed.

But [4] notes that the spectral clustering method gives equal weights to the first p eigenvectors of the Laplacian, whereas the smaller the eigenvalue is, the more important the corresponding eigenvector is. Moreover, only the first p eigenvalues are used and, hence, this approach does not use the entire structure of the graph. To avoid these problems, one can use a regularized version of the Laplacian: the heat kernel (also called the diffusion kernel). The diffusion matrix of the graph $\mathcal{G}$ for the parameter $\beta > 0$ is $D^\beta = e^{-\beta L}$ and the diffusion kernel is the function $K_\beta : (x_i, x_j) \in V \times V \to D^\beta_{i,j}$. This diffusion kernel has been intensively studied and used through the past years (see [5, 10, 17, 15, 22], among others). One of its main desirable properties comes from Aronszajn's Theorem [1] that states that there is a reproducing kernel Hilbert space (RKHS), $\mathcal{H}_\beta$, called the feature space, and a mapping function, $\phi_\beta : V \to \mathcal{H}_\beta$ such that for all i, j,

$$\langle \phi_\beta(x_i), \phi_\beta(x_j) \rangle_{\mathcal{H}_\beta} = K_\beta(x_i, x_j). \tag{1}$$

This last equation is commonly known as *kernel trick* and means that K_β is simply a scalar product between images by ϕ_β of the vertices of the graph. Similarly as spectral clustering, a k-means algorithm can be performed on this mapping. This method is known under the name *kernel k-means* (see [16, 7, 6]).

On a practical point of view, partitions coming from spectral clustering and kernel k-means cannot be directly compared by the way of k-means error because the mapping of the graph is not the same: the underlined metrics are not comparable. The same occurs for partitions coming from kernels with different values of β. To enable a comparison between all these partitions, we use a quality measure introduced by [12], the *q-modularity*, $Q_{\text{modul}} = \sum_{j=1}^{k}(e_j - a_j^2)$ where k is the number of clusters, e_j is the fraction of edges in the graph that connect two vertices in cluster j and a_j is the fraction of the edges in the graph that connect to one vertex in cluster j. This criterion does not depend on a mapping or a dissimilarity on the graph and is easily interpretable in terms of probability of having in/between-clusters edges: a high q-modularity means that vertices are clustered into dense subgraphs having few edges between them.

Table 1 summarizes the main characteristics of the partitions into 50 clusters of the medieval graph obtained by these two approaches. Obviously, both partitions share common properties: the q-modularity is similar and the vertices are concentrated in a few number of large clusters. More than three fourth of the vertices belong to a cluster having less than 7 vertices and the largest cluster contains more than one third of the vertices of the whole graph. In conclusion, the partitions provided by these two simple tools have to be improved.

	Spectral clustering	Kernel k-means ($\beta = 0.05$)
q-modularity	0.4195	0.4246
Number of clusters of size 1	16	17
Maximum size of the clusters	268	242
Median of the clusters' size	2	2
$3rd$ quartile of the clusters' size	7	7

Table 1: Details about the partitions obtained by spectral clustering and kernel k-means

3   Drawing the graph

From any partition obtained by clustering the graph, a simplified representation can be obtained by assigning a given glyph to each cluster where the surface of the glyph is proportional to the number of vertices of the given cluster. At the same time, glyphs are connected to each others by edges whose width is also proportional to the total number of weights between the vertices belonging to the two corresponding clusters. Glyphs can be spatially positioned by a *force directed algorithm* that aims at providing an aesthetic representation of a graph by assigning forces amongst the edges and nodes (see [8]). Examples of such a representation are given in Figure 1 for the two partitions described in section 2[1].
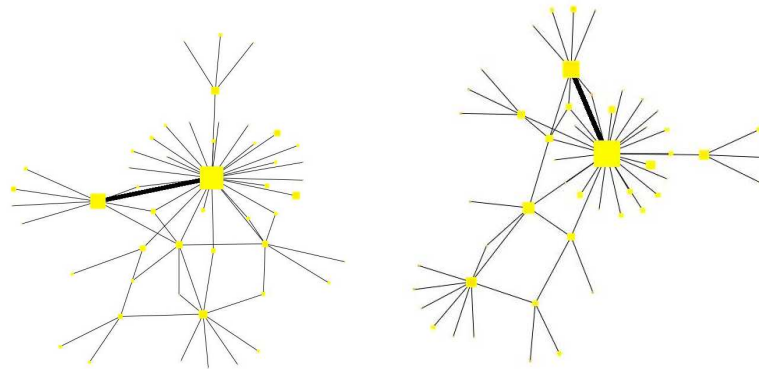

Figure 1: Force-directed algorithms used for a simplified representation coming from spectral clustering (left) and kernel k-means (right)

Both representations share common properties that can be seen as main structural properties of the graph: the network has a star shaped structure with two main groups of central people that can be seen as a kind of "rich club" (see [4]). Some tiny groups are totally isolated from this two main central clusters and linked to other secondary clusters. The two major clusters are strongly linked to each others. These pictures seem to give understandable representations of the structure of the graph but, unfortunately, the two main clusters respectively contain about 250 and 100 vertices, that is, more than half of the vertices of the graph: then, these two clusters are almost as complex as the initial graph.

4   A clustering and organizing algorithm

Several kernelized versions of SOM algorithm, that can perform simultaneously the objectives described in sections 2 (clustering) and 3 (representation), has been described in [11]. The present paper uses a batch version of the kernel SOM (that generally converges much faster) proposed in [4, 23].

The aim of self-organizing algorithms is to project the initial data on a prior structure that is generally a grid consisting in M neurons. A neighborhood relationship is defined on the

---

[1]All the graph figures have been made with the free software Tulip, available at `http://www.tulip-software.org/`

grid and the projection intends to preserve the initial topology of the data on this grid. The batch kernel SOM is simply a batch SOM performed on data that have been mapped on a RKHS; the algorithm is rewritten by the way of the kernel trick (Equation (1)).

This algorithm has been applied to the medieval graph with a rectangular grid of size 7×7. The main characteristics of the obtained partition is summarized in Table 2. It is compared to the partition obtained by using the batch SOM on the rows of the k eigenvectors associated to the smallest eigenvalues (this last approach has been named "spectral SOM").

	Spectral SOM	Kernel SOM $(\beta = 0.05)$
q-modularity	0.433	0.551
Final number of clusters	29	35
Number of clusters of size 1	11	13
Maximum size of the clusters	325	255
Median of the clusters' size	2	3
$3rd$ quartile of the clusters' size	10	10

Table 2: Details about the clustering obtained by spectral SOM and batch kernel SOM

The corresponding simplified representations, respecting the topology of the map, are provided in Figure 2[2] where an additional information is given by the U-matrices (see [20]) that smoothly represent the mean distances (respectively in the $\mathbb{R}^k$ space generated by k eigenvectors associated to the smallest eigenvalues and in the feature space) between the prototypes of each cluster. Clearly, kernel SOM provides better clustering than spectral SOM (larger q-modularity, much less vertices in the largest cluster). It also seems to be a little bit better than the spectral clustering and the kernel k-means (larger q-modularity, smaller number of tiny clusters - with less than 5 vertices).

In both cases, the simplified representations are well organized and easy to understand. Compared to Figure 1, the representation provided by kernel SOM is very close to the one provided by a simple clustering followed by a force directed representation of the clusters: a large cluster has a central position and is surrounded by smaller clusters. But looking at the u-matrix, the map is clearly divided into three main part (top left, top right and bottom right) which, according to color levels, are distant to each others.

This fact is clearly explained by Figure 3 (left) where the significance of each cluster clearly appears: the top left part of the map is the oldest cluster whereas the top right part is the youngest, with a continuous connexion of the dates on the map. Figure 3 (right) also provides some interesting informations about the social network: in particular, the top left part of the map has an homogeneous geographical setting which is the small village of Divilhac. This part of the map is only linked with the large clusters at the bottom right by a single peasant. This peasant doesn't live in this village but in the dominant village of the clusters to which he is linked at the bottom right of the map (St Julien[3]). Then, it seems that generational

---

[2]Colored and high quality images can be found at `http://nathalie.vialaneix.free.fr/maths/article-normal.php3?id_article=8`

[3]Readers interested by the location of the villages named in this paper will find a approximate map at `http://maps.google.com/maps/ms?ie=UTF8&hl=fr&msa=0&msid=100355826667676777753.000001134e74760eae6cd&z=10`
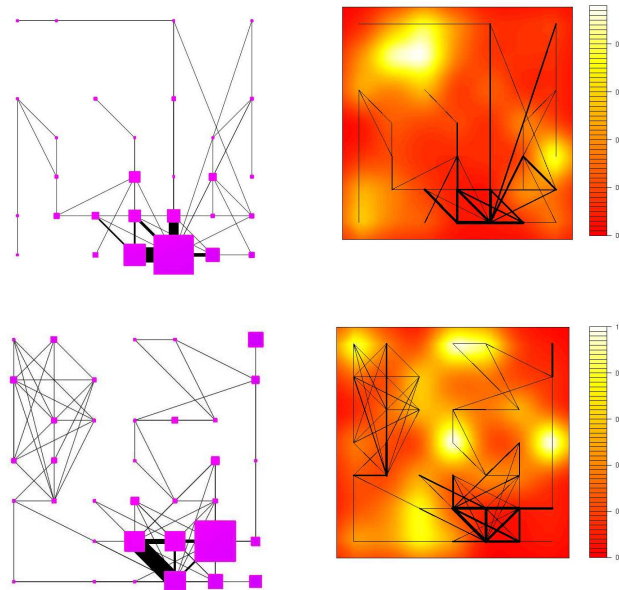
Figure 2: Final map obtained by spectral SOM (top left) and corresponding smoothed u-matrix (top right) and final map obtained by batch kernel SOM (bottom left) and corresponding smoothed u-matrix (bottom right)

relationships and geographical ones are very important in this network. Moreover, it is suprising to see how even large clusters have a good homogeneity of their geographical settings (see the top right and bottom right clusters for example).

5 Representing the whole graph from the kernel Self-Organizing Map

As the layout used for Figure 2 has been built to be well organized, it provides an interesting starting point for a readable presentation of the whole graph. In [19, 18], Truong *et al.* developed an energy model, in the spirit of force directed algorithms, but under location constraints. This model intends to represent graphs that are already clustered. By applying this algorithm to the self-organizing map presented in Figure 2, we obtain the representation of Figure 4 where the names of some peasants in the smallest clusters have been added.

If it is obvious that the representation of the bottom right part of the map still has to be improved, some important facts that seem to be of interest for historians have been emphasized: first of all the peasant that links the top left part of the map to the bottom right one is "Pierre Fornie". This man is already known by historians to be a major character. This name also appear a bit appart from the main clusters and was identified by historians as the same people (and not a namesake) which means that some ambiguities still exist in the database (database correction is currently underway, partly due to this first analysis). Moreover, the persons having a geographical setting different from the rest of the top left part of the map (Pern and Ganic instead of Divilhac) are named Trapas and Tessendier.
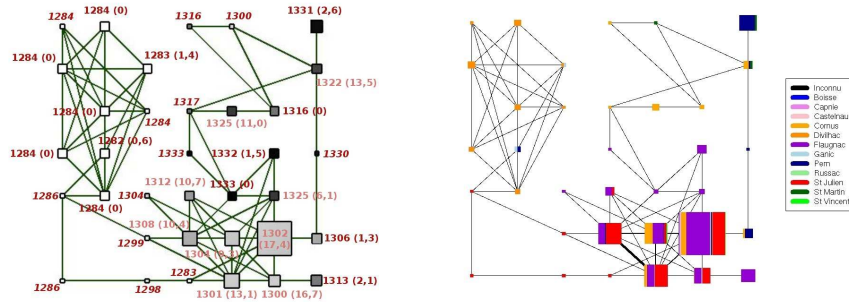
Figure 3: Mean dates of each clusters with standard deviation in parenthesis (left) and geographical settings distribution of each cluster (right)
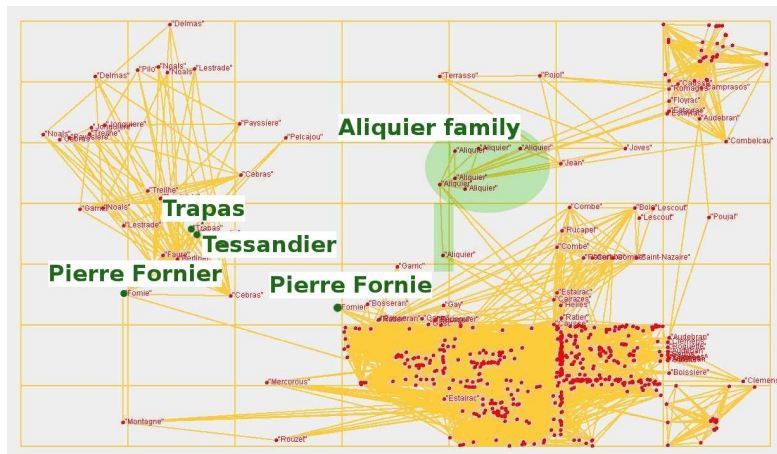

Figure 4: Representation of the whole medieval graph coming from kernel SOM

They also belong to families known for their dominant positions. Then, some clusters roughly homogeneous on the geographical point of view are connected to similar clusters via important families that do not live in the same area and that can be seen as important links between villages. Finally, the top right part of the map is linked to the bottom right one by a single family named Aliquier family, this leads to identify this family as being very important for the social cohesion of the network.

All these remarks have helped historians to understand the organization of the social network. Moreover, dominant families have been identified through this first study: the next objective of this project is to understand how they structured the society and also how they have evolved through the hard break of the Hundred Years' war.

TRACES (University of Toulouse Le Mirail), who provides us helpful comments and analysis.

References

[1] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

[2] R. Boulet, F. Hautefeuille, B. Jouve, P. Kuntz, B. Le Goffic, F. Picarougne, and N. Villa. Sur l'analyse de réseaux de sociabilité dans la société paysanne médiévale. In *MASHS 2007*, Brest, France, 2007.

[3] R. Boulet and B. Jouve. Partitionnement d'un réseau de sociabilité à fort coefficient de clustering. *Revue des Nouvelles Technologie de l'Information*, 9:569–574, 2007.

[4] R. Boulet, B. Jouve, F. Rossi, and N. Villa. Batch kernel SOM and related laplacian methods for social network analysis. *Neurocomputing*, 71(7-9):1257–1273, 2008.

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

[6] I.S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means, spectral clustering and normalized cuts. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 2004.

[7] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta. A survey of kernel and spectral methods for clustering. *Pattern Recognition*, 41:176–190, 2008.

[8] T. Fruchterman and B. Reingold. Graph drawing by force-directed placement. *Software-Practice and Experience*, 21:1129–1164, 1991.

[9] F. Hautefeuille. *Structures de l'habitat rural et territoires paroissiaux en bas-Quercy et haut-Toulousain du VIIème au XIVème siècle*. PhD thesis, University of Toulouse II (Le Mirail), 1998.

[10] 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.

[11] K.W. Lau, H. Yin, and S. Hubbard. Kernel self-organising maps for classification. *Neurocomputing*, 69:2033–2040, 2006.

[12] M.E.J. Newman. Mixing patterns in networks. *Physical Review, E*, 67:026126, 2003.

[13] M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review, E*, 69:026113, 2004.

[14] S.E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, August 2007.

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

[16] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, Cambridge, UK, 2004.

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

[18] Q.D. Truong, T. Dkaki, and P.J. Charrel. An energy model for the drawing of clustered graphs. In *Proceedings of Vème colloque international VSST*, Marrakech, Maroc, 21-25 octobre 2007.

[19] Q.D. Truong, T. Dkaki, and P.J. Charrel. Clustered graphs drawing. In *Proceedings of Stimulating Manufacturing Excellence in SME*, Hammamet, Tunisie, 14-16 février 2008.

[20] A. Ultsch and H. P. Siemon. Kohonen's self organizing feature maps for exploratory data analysis. In *Proceedings of International Neural Network Conference (INNC'90)*, pages 305–308, 1990.

[21] J. van den Heuvel and S. Pejic. Using Laplacian eigenvalues and eigenvectors in the analysis of frequency assignment problems. *Annals of Operations Research*, 107(1-4):349–368, 2001.

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

[23] N. Villa and F. Rossi. A comparison between dissimilarity SOM and kernel SOM for clustering the vertices of a graph. In *Proceedings of the 6th Workshop on Self-Organizing Maps (WSOM 07)*, Bielefield, Germany, September 2007.

[24] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.