ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

The Sum-over-Forests Clustering

Mathieu Senelle, Marco Saerens & François Fouss

Université catholique de Louvain (UCL), Belgium
LSM & ICTEAM

Abstract. This work introduces a novel way to identify dense regions in a graph based on a mode-seeking clustering technique, relying on the Sum-Over-Forests (SoF) density index [1] (which can easily be computed in closed form through a simple matrix inversion) as a local density estimator. We first identify the modes of the SoF density in the graph. Then, the nodes of the graph are assigned to the cluster corresponding to the nearest mode, according to a new kernel, also based on the SoF framework. Experiments on artificial and real datasets show that the proposed index performs well in nodes clustering.

1 Introduction

General introduction. Density is an important concept in graph analysis and has been proven to be of particular interest in various areas such as, for example, social networks, biology and World-Wide-Web [2–4]. The task of identifying dense regions on a graph can be based on various concepts (degree of a node, cliques, cores, etc.) leading to various approaches (see Section 1). The key concept on which our approach is based is forest enumeration and, in particular, the matrix-forest theorem [5,6], an extension of the well-known matrix-tree theorem (see, e.g., [7]), defining the **Sum-over-Forests** (SoF) **density index** [1].

A new clustering algorithm based on a mode-seeking procedure using the SoF density index is developed. Indeed, this index is used to identify local peaks of density in a graph, which are then considered as center of clusters (modes). The clustering in itself is performed assigning each node to its "closest" mode, according to a certain similarity (or distance) measure.

Brief related work. Clustering on graphs, also called community detection, is a topic that received a lot of attention recently, and extensive reviews exist on the subject (see for instance [8, 9]). Our work is more precisely based on mode-seeking methods, like the Mean Shift algorithm [10], which compute the modes of a probability density function to find high density areas. These methods were originally intended to be used in the feature space of the data, but adaptations to graph data were recently proposed [11–13]. Our work is quite similar to [13], as they use random walks to define modes (nodes most visited by a random walker), coupled with a steepest ascent procedure to form the clusters.

2 Background and notation

Consider a weighted directed graph or network without self-loops, G, not necessarily strongly connected, with a set of n nodes V (or vertices) and a set of arcs

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

E (or edges). To each arc linking node k and node k', we associate a positive number $c_{kk'} > 0$ representing the **immediate cost** of following this arc. The **cost matrix C** is the matrix containing the immediate costs $c_{kk'}$ as elements. If, instead of $\mathbf{C}$, we are given an adjacency matrix with elements $a_{kk'} \geq 0$ indicating the affinity between node k and node k', the corresponding costs could be computed from $c_{kk'} = 1/a_{kk'}$. The adjacency matrix containing the elements $a_{kk'}$ is denoted by $\mathbf{A}$, while the Laplacian matrix of a graph having adjacency matrix $\mathbf{A}$ is $\mathbf{L}(\mathbf{A}) = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \mathbf{Diag}(\mathbf{A}^{\mathrm{T}}\mathbf{e})$ is a diagonal matrix containing the column sums of $\mathbf{A}$. Here, $\mathbf{e}$ is a column vector full of 1's. Moreover, if the graph is undirected, it is assumed that, for each arc, there exist directed links in the two directions $k \to k'$ and $k' \to k$.

3 The Sum-over-Forests density index

For completeness, the present section summarizes developments made in [1] where the **Sum-over-Forest density index** is introduced, essentially based on [5, 6] and [14]. The idea is to define a **bag of forests** from which forests are sampled according to a Boltzmann probability distribution, so that large (high-cost) forests have a low probability of being sampled while short (low-cost) forests are sampled with a high probability. Then, the SoF density index of a node is defined as the expected outdegree of this node when sampling forests according to the Boltzmann distribution, thus providing a smoothed measure of density around that node.

Let us define the set of rooted forests φ in the graph G as $\mathcal{F} = \{\varphi_1, \varphi_2, \dots\}$. The *total cost* of such a forest φ is defined as the *sum* of the individual costs of the existing arcs belonging to φ, $C(\varphi)$. A forest with no arc (containing only individual nodes without any connection and thus no cost) has a 0 total cost. A **Boltzmann probability distribution** is defined on the set $\mathcal{F}$:

$$\mathrm{P}(\varphi) = \frac{\exp\left[-\theta C(\varphi)\right]}{\sum_{\varphi' \in \mathcal{F}} \exp\left[-\theta C(\varphi')\right]} \tag{1}$$

where the denominator $\mathcal{Z} = \sum_{\varphi \in \mathcal{F}} \exp\left[-\theta C(\varphi)\right]$ is called the **partition function**. Now, the **expected number of times** a link $k \to k'$ is present in a forest can easily be computed through

$$\overline{\eta}(k, k') = \sum_{\varphi \in \mathcal{F}} \mathrm{P}(\varphi)\, \delta(\varphi; k, k') \tag{2}$$

where $\delta(\varphi; k, k')$ is a Kronecker delta indicating if the link $k \to k'$ is present in forest φ. As shown in [1], this quantity can easily be computed in terms of $\mathbf{C}$ thanks to $\overline{\eta}(k, k') = -\frac{1}{\theta}\partial(\log \mathcal{Z})/\partial c_{kk'}$, which follows from the matrix-forest theorem [5, 6]. Then, the **expected outdegree** of node k on a forest, which

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

defines the **SoF density index**, is

$$\mathrm{dens}(k) = \sum_{\varphi \in \mathcal{F}} \mathrm{P}(\varphi) \left(\sum_{k'=1}^{n} \delta(\varphi; k, k') \right) = \sum_{k'=1}^{n} \overline{\eta}(k, k') \tag{3}$$

and corresponds to the sum of the contributions of the arcs issued from node k (notice that for the **expected weighted outdegree**, we would have $\mathrm{dens}(k) = \sum_{\varphi \in \mathcal{F}} \mathrm{P}(\varphi) \left(\sum_{k'=1}^{n} a_{kk'} \delta(\varphi; k, k') \right)$ instead). It can be computed in closed form for all nodes at once and requires a matrix inversion (see [1] for details):

$$\mathbf{dens} = \mathbf{W}\,\mathbf{diag}(\mathbf{Z}) - \mathbf{diag}(\mathbf{WZ}) \tag{4}$$

with $\mathbf{W} = \exp[-\theta \mathbf{C}]$ (elementwise exponential), $\mathbf{Z} = (\mathbf{I} + \mathbf{L}(\mathbf{W}))^{-1}$, $\mathbf{L}(\mathbf{W}) = \mathbf{D}_{\mathrm{w}} - \mathbf{W}$ is the Laplacian matrix computed from $\mathbf{W}$ and $\mathbf{D}_{\mathrm{w}} = \mathbf{Diag}(\mathbf{We})$ (for the expected weighted outdegree, we obtain $\mathbf{dens} = (\mathbf{A} \circ \mathbf{W})\,\mathbf{diag}(\mathbf{Z}) - \mathbf{diag}((\mathbf{A} \circ \mathbf{W})\mathbf{Z})$ where $\circ$ is the elementwise product).

4 The SoF mode-seeking clustering

Our algorithm is based on a simple mode-seeking procedure [10], as used in the Mean-Shift method, with the difference that it operates directly on graphs instead of the feature space.

The first step of our SoF clustering algorithm is to identify the modes of the density on the graph. The density estimation is provided by the Sum-over-Forests density index computed on each node (see Equation (4)). Then, starting from any node, a steepest ascent procedure is used, jumping from the original node to the one of its neighbour having the highest density score. This procedure ends when a local density peak (mode) is reached, i.e., when a node has the highest local SoF density score. The different modes are then considered as the prototypes or centers of the different clusters on the graph.

In order to form the different clusters, each node is associated to its corresponding mode. This is done by computing a similarity measure between each node and the different modes, and then associating the node to the most similar (closest) mode. The similarity measure showing the best results (detailed in Section 5) is a SoF extension of the forest similarity ($\mathbf{K}_{\mathrm{FS}}$) developed in [6], where we replaced the adjacency matrix $\mathbf{A}$ by the $\mathbf{W}$ matrix defined above. This choice is particularly consistent with our SoF density index, as it uses the same formalism, and can also be obtained from the cost matrix $\mathbf{C}$ using $\mathbf{K}_{\mathrm{SoF}} = (\mathbf{I} + \mathbf{L}(\mathbf{W}))^{-1}$.

This similarity measure between two nodes i and j introduced here is a valid kernel and has a nice probabilistic interpretation: it is the probability for node i to be present on a tree rooted in j (and therefore connected to j, see [6]), knowing that forests are sampled according to a Boltzmann distribution (Equation (1)). It therefore corresponds to the *a posteriori probability* that the root node of i is j, given the leaf node i of interest.

This clustering method works naturally in an unsupervised way (the natural number of clusters does not need to be known a priori), but can also work in

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

a supervised way. Indeed, when the set of modes has been identified, it can be restrained to the m clusters truly present in the data, by selecting the m largest modes to where most of the nodes converge, and then applying the clustering procedure. Obviously, this can only be done if the number of modes found is greater than m. In summary, this clustering method has two main benefits : it is unsupervised (even if it can also be used in a supervised way) and it works directly on graphs.

5 Experiments

Datasets. Five small to moderately sized networks are used to assess the performance of the clustering algorithm : Zachary's Karate Club [15], Dolphins [16], Football [17], Sampson's Monastery [18], and Political Books [19].

Evaluation methods. As the ground-truth clustering for each of the network used is known, the clustering obtained by the SoF method to the true clustering can be easily compared, using two different criteria: the adjusted rand index and the normalized mutual information. We compare our clustering method (1) in an unsupervised way to the Louvain method [20], and (2) when the number of clusters is given as an input to the Kernel K-Means and the Kernel Hierarchical clustering developed in [9]. Kernel K-Means is derived from the standard K-Means clustering algorithm and is able to identify nonlinearly separable clusters with the help of a kernel matrix. Kernel Hierachical clustering is a kernel version of Ward's hierarchical clustering [21]. As these two latter methods need a kernel matrix as input, we use the sigmoid commute-time kernel, with a parameter $\gamma = 7$, which proved to give good results [9].

Results and discussion. The results for the adjusted rand index (ARI) and the normalized mutual information criterion (NMI) are shown in Table 5. The parameter θ for the SoF clustering method is fixed to 0.1, as this value provides in all cases the best results. Moreover, we also observed that in order to identify dense areas, the larger a graph is, the higher the value of θ should be. This may be related to the forests' size, which are "scaled" according to the graph. The results for Kernel K-Means, Kernel Hierarchical clustering and Louvain method are averaged on 100 runs.

On the Zachary dataset, the SoF clustering obtains a perfect score of one in the two criteria. Kernel K-means and Hierarchical clustering obtain good results, with knowledge of the number of classes. The Louvain method finds three clusters (instead of two) and its result is quite below the other methods. For the Dolphins dataset, results without providing the number of clusters are essentially the same for the SoF and Louvain method. When the number of clusters is known, the SoF method's results jump to the ones of Kernel K-Means and above the ones of Hierarchical clustering. In the Football case, where 12 clusters are to be found, the SoF method identifies 11 clusters, and the Louvain method 6, which results in higher score for the SoF. In the supervised case, the

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

		SoFnbClust	K-Means	Hierarchical	SoF	Louvain
Zachary	ARI	1	0.9252	0.8823	1 (2)	0.5943 (3)
(2)	NMI	1	0.9067	0.8365	1 (2)	0.6360 (3)
Dolphins	ARI	0.9348	0.9294	0.7537	0.4080 (4)	0.4293 (3)
(2)	NMI	0.8889	0.8833	0.7014	0.6168 (4)	0.5985 (3)
Football	ARI	0.5874	0.7709	0.8893	0.5874 (11)	0.3572 (6)
(12)	NMI	0.7505	0.8669	0.9269	0.7505 (11)	0.6337 (6)
PolBooks	ARI	0.6679	0.6703	0.6559	0.6679 (2)	0.5974 (4)
(3)	NMI	0.6102	0.5706	0.5522	0.6102 (2)	0.5218 (4)
Monastery	ARI	0.5352	0.4223	0.1088	0.2475 (8)	0.4686 (5)
(4)	NMI	0.6548	0.5756	0.3579	0.6247 (8)	0.6332 (5)

Table 1: Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) values for five networks. SoFnbClust represents the SoF clustering method with number of clusters provided in input. For unsupervised methods (SoF and Louvain), the number of identified clusters is given in parenthesis.

results of the SoFnbClust are the same since the number of clusters found is below the true number and we cannot add artifically new clusters (in the Dolphins case, we only keep the modes which attract the most nodes). Hierarchical clustering gives the best results on this dataset. For the Political Books dataset, the SoF clustering performs better than the Louvain method, is slightly above the Hierarchical clustering and very close to the Kernel K-means. Finally, the results for the Monastery dataset show an ARI score inferior to the Louvain method, but a NMI score almost equal for these two methods. When the number of clusters is provided, the SoF clustering performs better than all other methods.

6 Conclusion and perspectives

This work introduces a new clustering algorithm: the SoF clustering. It is based on a mode-seeking procedure, identifying the modes as peaks of SoF density, and clustering the dataset using a forest similarity between the nodes. This method, which can be used in an unsupervised way or by providing the number of clusters, performs well on the five networks shown here. In the unsupervised case, the SoF clustering matches, and in some cases, outperforms the Louvain Method, and even the Kernel K-Means and the Hierarchical clustering. Without surprise, when the number of clusters is provided, the performance can greatly increase. In the future, the mode-seeking procedure detailed in this paper, using the SoF density index, could be coupled to another clustering algorithm. Indeed, the modes identified could serve as cluster prototypes given as input to, for instance, the K-Means method. The main drawback of the procedure is its computational complexity. We will therefore investigate the adaptation of the proposed technique to large graphs as, e.g., in [22].

References

[1] M. Senelle, S. Garcia-Diez, A. Mantrach, M. Shimbo, M. Saerens, and F. Fouss. The sum-over-forests density index: identifying dense regions in a graph. *arXiv:1301.0725 [cs.LG]; to appear in the IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

[2] D. Luce and A. Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116, 1949.

[3] X. Li, C. Foo, and S. Ng. Discovering protein complexes in dense reliable neighborhoods of protein interaction networks. *Computational Systems Bioinformatics Conference*, 6:157–168, 2007.

[4] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, 2000.

[5] P. Chebotarev and E. Shamis. The matrix-forest theorem and measuring relations in small social groups. *Automation and Remote Control*, 58(9):1505–1514, 1997.

[6] P. Chebotarev and R. Agaev. Forest matrices around the Laplacian matrix. *Linear Algebra and its Applications*, 356:253–274, 2002.

[7] W. Tutte. *Graph theory*. Cambridge University Press, 2002.

[8] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.

[9] L. Yen, F. Fouss, C. Decaestecker, P. Francq, and Marco Saerens. Graph nodes clustering with the sigmoid commute-time kernel: A comparative study. *Data and Knowledge Engineering*, 68:338–361, 2009.

[10] W.L.G. Koontz, P.M. Narendra, and K. Fukunaga. A graph-theoretic approach to nonparametric cluster analysis. *IEEE Transactions on Computers*, pages 936–944, 1976.

[11] S. Jouili, S. Tabbone, and V. Lacroix. Median graph shift : A new clustering algorithm for graph domain. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*, 2010.

[12] H. Liu and S. Yan. Robust graph mode seeking by graph shift. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010.

[13] M. Cho and K.M. Lee. Mode-seeking on graphs via random walks. In *Proceedings of the 25th Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[14] A. Mantrach, L. Yen, J. Callut, K. Francoisse, M. Shimbo, and M. Saerens. The sum-over-paths covariance kernel: A novel covariance measure between nodes of a directed graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1112–1126, 2010.

[15] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.

[16] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, and S.M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54:396–405, 2003.

[17] M. Girvan and M. E. Newman. Community structure in social et biological networks. *Proceedings of the National Academy Science of the United States of America*, 99(12):7821–7826, 2002.

[18] S.F. Sampson. A novitiate in a period of change: An experimental and case study of social relationships. *In Computational Intelligence, Theory and Applications*, 1968.

[19] V. Krebs. New political patterns. *Unpublished*, 2004.

[20] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 10:P10008, 2008.

[21] J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of American Statistical Association*, 58:236–244, 1963.

[22] A. Mantrach, N. van Zeebroeck, P. Francq, M. Shimbo, H. Bersini, and M. Saerens. Semi-supervised classification and betweenness computation on large, sparse, directed graphs. *Pattern Recognition*, 44(6):1212 – 1224, 2011.