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.

Agglomerative Hierarchical Kernel Spectral Clustering for Large Scale Networks

Raghvendra Mall[1], Rocco Langone[2] and Johan A.K. Suykens[3]

KU Leuven - ESAT/STADIUS
Kasteelpark Arenberg 10, bus 2446, B-3001 Leuven - Belgium

Abstract. We propose an agglomerative hierarchical kernel spectral clustering (AH-KSC) model for large scale complex networks. The kernel spectral clustering (KSC) method uses a primal-dual framework to build a model on a subgraph of the network. We exploit the structure of the projections in the eigenspace to automatically identify a set of distance thresholds. These thresholds lead to the different levels of hierarchy in the network. We use these distance thresholds on the eigen-projections of the entire network to obtain a hierarchical clustering in an agglomerative fashion. The proposed approach locates several levels of hierarchy which have clusters with high modularity (Q) and high adjusted rand index (ARI) w.r.t. the groundtruth communities. We compare AH-KSC with 2 state-of-the-art large scale hierarchical community detection techniques.

1 Introduction

In the modern era large scale complex networks are predominantly visible in social networks, collaboration networks, financial networks, biological networks etc. These complex networks show community like structures. This means that nodes of one community are more densely connected to nodes within the community and sparsely connected to other nodes. The problem of community detection has received wide attention [1] and two state-of-the-art large scale hierarchical community detection techniques are the Louvain method [2] and the Infomap method [3]. However, *these techniques suffer from a resolution limit i.e. they prevent the detection of high quality clusters of finer granularity which is also shown by our experiments.*

Recently a kernel spectral clustering (KSC) method for big data networks was proposed in [4]. The method works by building a model on a representative subgraph of the large network. This subgraph is obtained by the Fast and Unique Representative Subset (FURS) selection technique as proposed in [5]. This subset is used to build the KSC model. The model requires a kernel function which can have parameters and needs to identify the number of clusters k in the network. A self-tuned KSC model for big data networks was proposed in [6]. The power of the KSC method is that it creates a model which can be used for out-of-sample extensions. Thus, we can infer community affiliation for unseen nodes of the large scale network using this model.

The goal of hierarchical clustering is to locate multiple levels of hierarchy in the network with high quality clusters at each level. In this paper we exploit the structure of the eigen-projections corresponding to the validation set of nodes to obtain a set of distance thresholds ($\mathcal{T}$). These distance thresholds ($t \in \mathcal{T}$)

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.

are used on the projections of the entire network which is obtained by the out-of-sample extensions of the KSC model. We then perform an agglomerative hierarchical clustering using $\mathcal{T}$ to produce good quality clusters at multiple levels of hierarchy. Hence our approach doesn't suffer from resolution limit problem.

2 Kernel Spectral Clustering (KSC)

We briefly describe the KSC method for large scale networks. A network is represented as a graph $G(V, E)$ where V denotes vertices and E the edges. For large scale networks, the training data comprise the adjacency list of all the nodes $v_i, i = 1, \ldots, N_{tr}$. Training, validation and test set of nodes are given by the V_{tr}, V_{valid}, V_{test} with cardinality N_{tr}, N_{valid} and N_{test} respectively. These adjacency lists can efficiently be stored in the memory as real world networks are highly sparse and have limited connections for each node.

For V_{tr} training nodes the dataset is given by $\mathcal{D} = \{x_i\}_{i=1}^{N_{tr}}$, $x_i \in \mathbb{R}^N$. Given $\mathcal{D}$ and a user-defined $maxk$ (maximum number of clusters in the network), the primal formulation of the weighted kernel PCA [8] is given by:

$$\begin{aligned} \min_{w^{(l)}, e^{(l)}, b_l} \quad & \frac{1}{2} \sum_{l=1}^{maxk-1} w^{(l)\intercal} w^{(l)} - \frac{1}{2N_{tr}} \sum_{l=1}^{maxk-1} \gamma_l e^{(l)\intercal} D_\Omega^{-1} e^{(l)} \\ \text{such that} \quad & e^{(l)} = \Phi w^{(l)} + b_l 1_{N_{tr}}, l = 1, \ldots, maxk-1 \end{aligned} \tag{1}$$

where $e^{(l)} = [e_1^{(l)}, \ldots, e_{N_{tr}}^{(l)}]^\intercal$ are the projections onto the eigenspace, $l = 1, \ldots, maxk-1$ indicates the number of score variables required to encode the $maxk$ communities, $D_\Omega^{-1} \in \mathbb{R}^{N_{tr} \times N_{tr}}$ is the inverse of the degree matrix associated to the kernel matrix Ω with $\Omega_{ij} = K(x_i, x_j) = \phi(x_i)^\intercal \phi(x_j)$. Φ is the feature matrix such that $\Phi = [\phi(x_1)^\intercal; \ldots; \phi(x_{N_{tr}})^\intercal]$ and $\gamma_l \in \mathbb{R}^+$ is the regularization constant. We note that $N_{tr} \ll N$ i.e. the number of nodes in the training set is much less than the total number of nodes in the large scale network. Each element of kernel matrix Ω is taken as $\Omega_{ij} = \frac{x_i^\intercal x_j}{\|x_i\| \|x_j\|}$ and is calculated using notions of set intersection and union as shown in [6]. The primal clustering model is then represented by: $e_i^{(l)} = w^{(l)\intercal} \phi(x_i) + b_l, i = 1, \ldots, N_{tr}$, where $\phi : \mathbb{R}^N \to \mathbb{R}^N$ and b_l are the bias terms, $l = 1, \ldots, maxk-1$. For large scale networks we can utilize the explicit expression of the underlying feature map as shown in [6]. The dual problem corresponding to this primal formulation is:

$$D_\Omega^{-1} M_D \Omega \alpha^{(l)} = \lambda_l \alpha^{(l)}, \tag{2}$$

where M_D is the centering matrix which is defined as $M_D = \mathrm{I}_{N_{tr}} - (\frac{(1_{N_{tr}} 1_{N_{tr}}^\intercal D_\Omega^{-1})}{1_{N_{tr}}^\intercal D_\Omega^{-1} 1_{N_{tr}}})$. The $\alpha^{(l)}$ are the dual variables and the positive definite kernel function $K : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ plays the role of similarity function. The corresponding predictive model is $\hat{e}^{(l)}(x) = \sum_{i=1}^{N_{tr}} \alpha_i^{(l)} K(x, x_i) + b_l$ which provides clustering inference for adjacency list x corresponding to each test node $v \in V_{test}$.

3 Agglomerative Hierarchical Clustering

In [6], an affinity matrix A_{valid} was created using the latent variable matrix $E_{valid} = [e_1, \ldots, e_{N_{valid}}]^\intercal$ which is a $N_{valid} \times (maxk-1)$ matrix, as:

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.

$$A_{valid}(i,j) = CosDist(e_i, e_j) = 1 - \cos(e_i, e_j) = 1 - \frac{e_i^\intercal e_j}{\|e_i\|\|e_j\|}. \quad (3)$$

The $CosDist(\cdot,\cdot)$ function takes values between $[0, 2]$. The nodes which belong to the same cluster will have smaller $CosDist(e_i, e_j), \forall i, j$ in the same cluster. It was shown in [6] that a rotation of the A_{valid} matrix has a block diagonal structure. This block diagonal structure was used to identify the ideal number of clusters k in the network using the concept of entropy and balanced clusters.

3.1 Finding Distance Thresholds using Validation Set

In our proposed approach we refer to the affinity matrix at level 0 of hierarchy as $A^{(0)}_{valid}$. After obtaining this matrix as in [6], we perform an agglomerative hierarchical clustering in a bottom up fashion. After several empirical evaluations, we set $t^{(0)} = 0.15$ for lowest level of hierarchy to make the approach tractable to large scale networks. We greedily select the validation node with maximum number of similar nodes in the latent space i.e. we select the projection e_i which has a maximum number of projections e_j satisfying $A^{(0)}_{valid}(i,j) < t^{(0)}$. We put the indices of these nodes in $C^{(0)}_1$ representing the 1^{st} cluster at level 0. We then remove these nodes and corresponding entries from $A^{(0)}_{valid}$ to obtain a reduced matrix. This process is repeated recursively until $A^{(0)}_{valid}$ becomes empty. Thus, we obtain the set $C^{(0)} = \{C^{(0)}_1, \ldots, C^{(0)}_q\}$ where q is the maximum number of clusters at level 0 of the hierarchy. The set $C^{(0)}$ has clusters containing the indices of the nodes belonging to those clusters.

To obtain the clusters at the next level of hierarchy we treat the communities at the previous levels as nodes. We then calculate the average cosine distance between these nodes using the information present in these nodes. We create a new affinity matrix at each level (h) as:

$$A^{(h)}_{valid}(i,j) = \frac{\sum_{k \in C^{(h-1)}_i} \sum_{l \in C^{(h-1)}_j} A^{(h-1)}_{valid}(k,l)}{|C^{(h-1)}_i| \times |C^{(h-1)}_j|}, \quad (4)$$

where $|\cdot|$ represents the cardinality of the set. We estimate the minimum cosine distance between each individual cluster and the other clusters (not considering itself). We then select the mean of these values as the new threshold for that level to combine clusters. This is because if we consider the minimum of all the distance values then there is a risk of only combining 2 clusters at that level. However, it is desirable to combine multiple sets of different clusters. Thus, the new threshold $t^{(h)}$ at level h is set as: $t^{(h)} = \text{mean}(\min_j(A^{(h)}_{valid}(i,j)))$, $i \neq j$. We use this process iteratively till we have 1 big cluster containing all the nodes. As a consequence we obtain the hierarchical clustering $C = \{C^{(0)}, \ldots, C^{(maxh)}\}$ where $maxh$ is the maximum level in the hierarchy and is obtained automatically. We also obtain a set of distance thresholds $\mathcal{T} = \{t^{(0)}, \ldots, t^{(maxh)}\}$.

3.2 Hierarchical Clustering of Test nodes

We use the entire network as test set. The latent variable matrix for the test set obtained by out-of-sample extensions is defined as $E_{test} = [e_1, \ldots, e_{N_{test}}]^\intercal$

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.

with time complexity $O(N_{tr} \times N_{test})$. We can store this E_{test} matrix in memory but cannot create an affinity matrix of size $N_{test} \times N_{test}$ ($N_{test} \gg N_{tr}$) due to memory constraints. As the validation set is a representative subset of the whole network [5], the threshold set $\mathcal{T}$ can be used to obtain a hierarchical clustering for the entire network. To make the proposed approach self-tuned we use $t^{(i)} > t^{(0)} > 0.15$, $i > 0$, in the test phase.

In order to avoid creating the affinity matrix for the large network we take the projection of the first test node and calculate its similarity with the projections of all the test nodes. We then locate the indices (j) of those projections s.t. $CosDist(e_1, e_j) < t^{(1)}$ and put them in cluster $C_1^{(1)}$. Since $t^{(1)} > 0.15$, typically the number of computations required to construct $C_1^{(1)}$ is $O(p \times N_{test})$ where $p \ll N_{test}$. We then remove entries corresponding to those projections in E_{test} to obtain a reduced matrix. We perform the procedure recursively until E_{test} is empty to obtain $C^{(1)} = \{C_1^{(1)}, \ldots, C_p^{(1)}\}$ where p is the maximum number of clusters at hierarchical level 1. After 1^{st} level we use the same procedure as for validation set i.e. creating an affinity matrix at each level using the cluster information along with the threshold set $\mathcal{T}$ to obtain the hierarchical structure in an agglomerative way. The cluster memberships are propagated iteratively from 1^{st} level to highest level of hierarchy. We illustrate this test phase on a synthetic network with $10,000$ nodes in Figure 1.

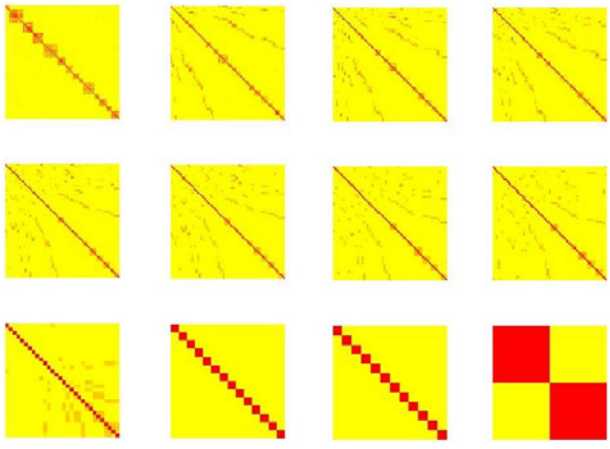

Fig. 1: Affinity matrices for levels of hierarchy on test set

3.3 Experiments

We generated synthetic benchmark networks Net_1, Net_2, Net_3 and Net_4 with $2,000$, $10,000$, $50,000$ and $250,000$ nodes respectively from the toolkit proposed in [7]. Figure 2 plots the original synthetic network and the network estimated by proposed approach for $10,000$ nodes. This toolkit generates benchmark networks with only 2 levels of hierarchy using different mixing parameter μ_1 and μ_2 for macro and micro communities respectively. We fixed $\mu_1 = 0.1$ and $\mu_2 = 0.2$ 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.

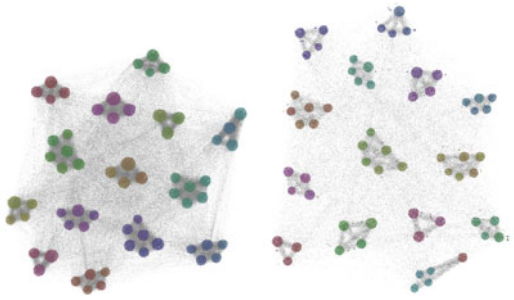

Fig. 2: Original hierarchical network (left) and estimated hierarchical network (right) for synthetic network with $10,000$ nodes. The orientation and position of the communities might vary in the two plots. Both plots have 3 clusters with 6 micro communities, 3 clusters with 5 micro communities, 6 clusters with 4 micro communities and 2 clusters with 3 micro communities.

our experiments. We evaluated the quality of the communities using external quality metrics like adjusted rand index (ARI) and variation of information (VI) and internal quality metrics like modularity (Q) and cut-conductance (CC).

Table 1 provides the top 10 levels of hierarchy for these 4 synthetic networks by our proposed approach referred as AH-KSC. The actual number of clusters at the 2 levels of hierarchy for Net_1, Net_2, Net_3 and Net_4 are $(37, 9)$, $(63, 14)$, $(141, 13)$ and $(64, 15)$ respectively. Table 1 shows that AH-KSC not only detects these 2 groundtruth clusters in most cases but also extracts several other meaningful levels of hierarchy. Due to space limitations we restrict the results to the 2 best levels of hierarchy w.r.t. various quality metrics. Table 2 also shows a comparison with the Louvain (LOU) and Infomap (IMAP) methods. We report the mean results for LOU and IMAP over 10 iterations. In Table 2, k_1 and k_2 are the best match for various methods w.r.t. the 2 level of groundtruth clusters.

Hierarchy	Net_1 (k)	Net_2 (k)	Net_3 (k)	Net_4 (k)
10	-	84	**134**	-
9	-	80	112	250,000
8	2,000	76	106	982
7	63	70	103	541
6	40	**63**	97	400
5	39	30	87	187
4	**37**	**14**	44	**66**
3	**15**	12	**13**	**9**
2	9	2	5	2
1	1	1	1	1

Table 1: Number of clusters (k) for top 10 levels of hierarchy by AH-KSC method. The number of clusters close to the true number of clusters are highlighted. The AH-KSC method provides more insight by identifying several meaningful levels of hierarchy with good quality clusters w.r.t. quality metrics like ARI, VI, Q and CC.

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.

From Table 2 we observe that the Louvain method works better when the number of clusters is small but fails to effectively find clusters of finer granularity. The Infomap method is the weakest among the 3 methods. However, AH-KSC can identify granular clusters of high quality as can be observed for Net_1, Net_2 and Net_3 along with obtaining good quality clusters at coarser levels. From Table 2 we also observe that the Q metric is biased towards small clusters and CC is biased towards a large number of communities.

Dataset	**Methods**	k_2	ARI_2	k_2	VI_2	k_2	Q_2	k_2	CC_2	k_1	ARI_1	k_1	VI_1	k_1	Q_1	k_1	CC_1
Net_1	AH-KSC	37	**1.00**	37	**0.00**	15	**0.77**	63	**4.7e-4**	9	**1.00**	9	**0.00**	9	**0.79**	39	**4.8e-4**
	LOU	32	0.84	32	0.22	32	0.69	32	**4.7e-4**	9	**1.00**	9	**0.00**	9	**0.79**	9	4.9e-4
	IMAP	8	0.32	8	1.52	8	**0.77**	8	0.5e-4	8	0.92	8	0.13	6	0.49	6	0.5e-4
Net_2	AH-KSC	76	**0.99**	76	**0.02**	14	**0.83**	146	**9.5e-5**	14	**1.00**	14	**0.00**	12	0.81	95	**9.7e-05**
	LOU	52	0.75	52	0.35	15	0.82	52	9.8e-5	14	**1.00**	14	**0.00**	14	**0.83**	14	9.9e-5
	IMAP	13	0.32	13	1.58	13	0.82	249	9.8e-5	13	0.95	13	0.08	6	0.52	13	9.8e-5
Net_3	AH-KSC	134	0.68	134	0.61	44	0.77	134	1.98e-5	13	**1.00**	13	**0.00**	13	**0.82**	103	**1.99e-5**
	LOU	135	**0.85**	135	**0.19**	20	**0.81**	135	1.98e-5	13	**1.00**	13	**0.00**	13	**0.82**	13	2.0e-5
	IMAP	13	0.16	13	2.38	14	0.62	590	**1.97e-5**	13	**1.00**	13	**0.00**	13	**0.82**	13	2.0e-5
Net_4	AH-KSC	187	**0.43**	187	**1.22**	197	0.73	982	**3.93e-6**	66	0.86	66	0.49	66	0.77	541	**3.96e-6**
	LOU	19	0.39	19	1.3	19	0.81	19	3.99e-6	15	**1.00**	15	**0.00**	15	**0.83**	15	3.99e-6
	IMAP	11	0.21	11	1.87	6,869	0.2	6,869	4.0e-5	11	0.68	11	0.42	11	0.78	11	4.0e-15

Table 2: Evaluation of clusters by different hierarchical methods

4 Conclusion

We proposed AH-KSC for large scale networks exploiting the structure of the projections in the eigenspace using a set of distance thresholds ($\mathcal{T}$). The proposed method overcomes the resolution limit problem and can locate high quality communities at finer levels of granularity.

Acknowledgements: The work is supported by Research Council KUL, ERC AdG A-DATADRIVE-B, GOA/10/09MaNet, CoE EF/05/006, FWO G.0588.09, G.0377.12, SBO POM, IUAP P6/04 DYSCO.

References

[1] U. von Luxburg A tutorial on Spectral clustering. *Statistics and Computing*, 17(4):395-416, 2007.

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

[3] M. Rosvall and C. Bergstrom, Maps of random walks on complex networks reveal community structure. *PNAS*, 105:1118-1123, 2008.

[4] R. Mall, R. Langone and J.A.K. Suykens, Kernel Spectral Clustering for Big Data Networks, *Entropy (Special Issue: Big Data)*, 15(5):1567-1586, 2013.

[5] R. Mall, R. Langone and J.A.K. Suykens, FURS: Fast and Unique Representative Subset selection retaining large scale community structure, *Social Network Analysis and Mining*, 3(4):1075-1095, 2013.

[6] R. Mall, R. Langone and J.A.K. Suykens, Self-Tuned Kernel Spectral Clustering for Large Scale Networks, *Proceedings of the IEEE International Conference on Big Data*, (IEEE BigData 2013), October 6-9, Santa Clara (U.S.A), 2013.

[7] S. Fortunato, Community detection in graphs. *Physics Reports*, 486:75-174, 2009.

[8] C. Alzate, J.A.K. Suykens, Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):335-347, 2010.