Proceedings of International Joint Conference on Neural Networks, San Jose, California, USA, July 31 – August 5, 2011

Modularity-based model selection for kernel spectral clustering

Rocco Langone, Carlos Alzate and Johan A. K. Suykens

***Abstract*—A proper way of choosing the tuning parameters in a kernel model has a fundamental importance in determining the success of the model for a particular task. This paper is related to model selection in the framework of community detection on weighted and unweighted networks by means of a kernel spectral clustering model. Here we propose a new method based on Modularity (a popular measure of community structure in a network) which can deal with quite general situations (i.e. overlapping communities with different sizes). Thus we use Modularity criterion for model selection and not at the training level, which is the case of all the clustering algorithms proposed so far in the literature.**

I. Introduction

In recent years, the study of networks has become a major issue in the scientific community. Many complex systems can be depicted as networks, where the nodes (or vertices) represent some entities between which some relationships exist. Examples include social networks, web graphs, telecommunication networks, biological networks, trade networks. An analysis very useful in starting to shed light on such network data sets is the community detection problem or clustering, that is identifying groups of nodes within which the connections (or edges) are numerous and between which they are scarce. Spectral clustering methods are a standard technique used for clustering, based on the eigendecomposition of a Laplacian matrix derived from the data. Recently a spectral clustering formulation as a weighted kernel PCA problem with primal and dual representations has been proposed in [1].

The main advantage of this interpretation is the extension of the clustering model to out-of-sample points. The clustering model can be trained on a subset of the whole graph (by solving an affordable eigenvalue problem) and then applied to the rest of the network in a learning framework. This issue is particularly important when we have to deal with huge complex networks. In this picture, it becomes crucial to have a good criterion to properly select the parameters to feed into the model (like the kernel parameters and the number of clusters). In fact this allows to obtain a relevant grouping among the data.

In the past a method to achieve this goal was proposed by some of the authors and is called Balanced line fit (BLF, see [1]). The main drawback of this method is that it gives optimal results when the clusters to find are well separated. Moreover it is characterized by a free parameter accounting of the balance of the clusters which has to be selected. On the other hand, in the real cases typically the communities are of unequal size and/or density and they can shares a little or a big amount of nodes (i.e. they overlap). Therefore in this paper we propose a new criterion, that can be used in a complementary or alternative way to the BLF method, in order to deal with more general cases. This criterion is based on Modularity, a quality function introduced in [2] and well known among the experts. The Modularity statistic has been proved to be a meaningful quality function accounting for the presence of a significant community structure in networks. It quantifies the quality of a division of a network into modules. Good divisions, which have high values of the Modularity, are those in which there are dense internal connections between the nodes within modules but only few connections between different modules. Due to its versatility, it gave rise to many applications. In particular in this paper we use it to extend our previous work, in order to find a model selection criterion more suitable than the previous (the BLF), in relation to the analysis of network data. The most common use of the Modularity is as a basis for optimization methods for detecting community structure in networks, but only at the training level (like in [3]). In our case, however, we use it as a cluster validity criterion for our model selection purposes. In particular, we consider Modularity to judge the partitions found by the kernel spectral clustering algorithm, which is based on its own optimization problem (briefly described later). Comparison of the kernel spectral clustering model with other algorithms, in terms of its ability in discovering interesting communities in graphs, is beyond the scope of this paper.

The rest of this paper is organized as follows: Section II summarizes the kernel spectral clustering model. Section III describes the Modularity-based criterion for model selection. In Section IV we describe how we select the training and validation set to use in the learning process. Some simulation results are presented in Section V: weighted and unweighted networks are taken into account, considering both real and artificial datasets. Finally, Section VI concludes the paper and suggests some future works.

II. Kernel spectral clustering model

A. General picture

A graph (or network) is a mathematical structure used to model pairwise relations between objects from a certain collection. It refers to a set of vertices or nodes and a collection of edges that connect pairs of vertices. A way to represent a graph is the use of a similarity matrix, which is an $N \times N$ matrix with N equal to the number of vertices in the network. If the graph is unweighted S is called adjacency matrix (in general indicated with the symbol A) and $S_{ij} = 1$ if there is an edge connecting the vertices i and j, otherwise $S_{ij} = 0$. In the case of a weighted graph, S is called affinity matrix and S_{ij} indicates the strength of the link between the vertices i and j. Associated to the similarity matrix there is the degree matrix D, a diagonal matrix with diagonal entries $d_i = \sum_j S_{ij}$ indicating the sum of all the edges (or weights) connecting node i with the other vertices.

The authors are with the Department of Electrical Engineering ESAT-SCD-SISTA, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium (email: {rocco.langone, carlos.alzate, johan.suykens}@esat.kuleuven.be).

978-1-4244-9636-5/11/$26.00 ©2011 IEEE

Given a graph (weighted or unweighted), several properties of it can be explained through spectral graph theory, which is the study of the eigenspectrum of graph Laplacian matrices ([4], [5]). Typical graph Laplacians are: the unnormalized Laplacian defined as $L = D - S$, the symmetric normalized Laplacian $L_{\text{SYM}} = D^{-1/2}LD^{-1/2} = I_N - D^{-1/2}SD^{-1/2}$ and the non-symmetric normalized Laplacian $L_{\text{RW}} = D^{-1}L = I_N - D^{-1}S$ denoted L_{RW} because it is related to a random walk on the graph. In the latter case, the clustering problem can be interpreted as finding a partition of the graph in such a way that the random walker remains most of the time in the same cluster with few jumps to other clusters, minimizing the probability of transitions between clusters. The stochastic transition matrix P of a random walker on a graph can be obtained by normalizing the similarity matrix S associated to the graph such that its rows sum to 1. The ij-th entry of P represents the probability of moving from node i to node j in one step of the process. This transition matrix can be defined as $P = D^{-1}S$. The corresponding eigenvalue problem becomes $Pr = \xi r$, and as we show afterward it can be viewed as the dual problem of a constrained optimization problem typical of least squares support vector machines (LS-SVM)[6].

B. Primal-dual formulation

Given training data $\mathcal{D} = \{x_i\}_{i=1}^N, x_i \in \mathbb{R}^d$ and the number of clusters k, the primal problem of spectral clustering via weighted kernel PCA is formulated as follows [1]:

$$\min_{w^{(l)},e^{(l)},b_l} \frac{1}{2}\sum_{l=1}^{k-1} w^{(l)^T} w^{(l)} - \frac{1}{2N}\sum_{l=1}^{k-1}\gamma_l e^{(l)^T} D^{-1} e^{(l)} \quad (1)$$

$$\text{such that } e^{(l)} = \Phi w^{(l)} + b_l 1_N \quad (2)$$

where $e^{(l)} = [e_1^{(l)}, \ldots, e_N^{(l)}]^T$ are the projections, $l = 1, \ldots, k-1$ indicates the score variables needed to encode the k clusters to find, $D^{-1} \in \mathbb{R}^{N\times N}$ is the inverse of the degree matrix D introduced in the previous section, Φ is the $N \times d_h$ feature matrix $\Phi = [\varphi(x_1)^T; \ldots; \varphi(x_N)^T]$ and $\gamma_l \in \mathbb{R}^+$ are regularization constants. The clustering model is expressed by:

$$e_i^{(l)} = w^{(l)^T}\varphi(x_i) + b_l, i = 1, \ldots, N \quad (3)$$

where $\varphi : \mathbb{R}^d \to \mathbb{R}^{d_h}$ is the mapping to a high-dimensional feature space, b_l are bias terms, $l = 1, \ldots, k-1$. The projections $e_i^{(l)}$ represent the latent variables of a set of $k-1$ binary clustering indicators given by $\text{sign}(e_i^{(l)})$ which can be combined to form the final groups in an encoding/decoding scheme. The dual problem related to this primal formulation is:

$$D^{-1}M_D\Omega\alpha^{(l)} = \lambda_l\alpha^{(l)} \quad (4)$$

where Ω is the kernel matrix with ij-th entry $\Omega_{ij} = K(x_i, x_j)$, M_D is a centering matrix defined as $M_D = I_N - \frac{1}{1_N^T D^{-1} 1_N} 1_N 1_N^T D^{-1}$, the $\alpha^{(l)}$ are dual variables. The kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ plays the role of the similarity function of the graph. Now, the dual problem is equivalent to the random walk model and represents the weighted kernel PCA formulation of it used in our simulations (for a complete derivation see [1]).

C. Encoding/decoding scheme

In the ideal case of k well separated clusters and properly chosen kernel parameters, the matrix $D^{-1}M_D\Omega$ has $k-1$ piecewise constant eigenvectors with eigenvalue 1 (see for example [7]). In the eigenvector space every cluster $\mathcal{A}_p$, $p = 1, \ldots, k$ is a point and is represented with a unique codeword $c_p \in \{-1, 1\}^{k-1}$. The codebook $\mathcal{CB} = \{c_p\}_{p=1}^k$ can then be obtained in the training process from the rows of the binarized eigenvector matrix $[\text{sign}(\alpha^{(1)}), \ldots, \text{sign}(\alpha^{(k)})]$. An effect of the centering matrix M_D defined in the last section is the fact that the eigenvectors have zero mean. This is important for encoding since the optimal threshold for binarizing the eigenvectors is automatically determined. Taking into account that the first eigenvector $\alpha^{(1)}$ already provides a binary clustering then number of score variables needed to encode k clusters is $k-1$. The decoding scheme consists of comparing the cluster indicators obtained in the validation/test stage with the codebook and selecting the nearest codeword in terms of Hamming distance. This scheme corresponds to the ECOC decoding procedure and it is used for out-of-sample extension. In particular, the proposed extension is based on the score variables which correspond to the projections of the mapped out-of-sample points onto the eigenvectors found in the training stage. The cluster indicators can be obtained by binarizing the score variables for out-of-sample points as follows:

$$\text{sign}(e_{\text{test}}^{(l)}) = \text{sign}(\Omega_{\text{test}}\alpha^{(l)} + b_l 1_{\text{Ntest}}) \quad (5)$$

with $l = 1, \ldots, k-1$. Ω_{test} is the $N_{\text{test}} \times N$ kernel matrix evaluated using the test points with entries $\Omega_{\text{test,ri}} = K(x_r^{\text{test}}, x_i)$, $r = 1, \ldots, N_{\text{test}}$, $i = 1, \ldots, N$. This natural extension to out-of-sample points corresponds to the main advantage of the kernel spectral clustering framework. In this way, the clustering model can be trained, validated and tested in an unsupervised learning scheme.

D. BLF model selection criterion

The BLF criterion exploits the shape of the points in the projections space: it reaches its maximum value 1 when the clusters do not overlap, and in this ideal situation the communities are represented as lines in this space. In particular the BLF is defined in the following way[1]:

$$\text{BLF}(\mathcal{D}^V, k) = \eta\text{linefit}(\mathcal{D}^V, k) + (1-\eta)\text{balance}(\mathcal{D}^V, k) \quad (6)$$

where $\mathcal{D}^V$ represents the validation set and k as usual indicates the number of clusters. The linefit index equals 0 when the score variables are distributed spherically (i.e. the eigenvalues are identical) and equals 1 when the score variables are collinear (representing points in the same cluster). The balance index equals 1 when the clusters have the same number of elements and tends to 0 in extremely unbalanced cases. The parameter η controls the importance given to the linefit with respect to the balance index and takes values in the range $[0, 1]$. Then (6) can be used to select the number of clusters and the kernel tuning parameters.

III. Model selection criterion by means of Modularity evaluation

A. Introduction

Often people use heuristics to select the tuning parameters present in their models. Since model selection is a crucial point, here we propose a systematic way to do it properly. Our method is based on a validation procedure[1]. We train the kernel spectral clustering model described in the previous section with different number of clusters and (where needed) several values of the kernel parameters. In the validation step the obtained groupings are judged depending on Modularity: the one (or more) partition with the highest value of Modularity is selected.

Modularity is a quality function of a graph introduced in [2]. It is based on the idea that a random graph is not expected to have a cluster structure, so the possible existence of clusters can be revealed by the comparison between the actual density of edges and the density one would expect to have in the graph if the vertices were attached randomly, regardless of community structure (this characterizes a particular null[2] model). Modularity can be either positive or negative, with positive high values indicating the possible presence of a strong community structure. It can be written as follows:

$$Q = \frac{1}{2m} \sum_{ij} (S_{ij} - F_{ij}) \delta_{ij} \tag{7}$$

with $i, j \in \mathcal{A}_p$. The sum runs over all pairs of vertices, S as before is the similarity[3] matrix, m indicates the sum of all the weights, and F_{ij} represent the expected number of edges between vertices i and j in the null model. The δ_{ij} function yields 1 if vertices i and j are in the same community and 0 otherwise. Since the standard null model of Modularity imposes that the expected degree sequence matches the actual degree sequence of the graph, the Modularity can be written as: $Q = \frac{1}{2m} \sum_{ij} (S_{ij} - \frac{d_i d_j}{2m}) \delta_{ij}$, where we indicate with $d_i = \sum_j S_{ij}$ the degree of the vertex i. Then, after some linear algebra calculations [9], it can be shown that the problem of maximizing the Q-measure in order to find the optimal partition is given by:

$$\max_X [\text{tr}(X^T M X)] \text{ such that } X^T X = D^M \tag{8}$$

Here $M = S - \frac{1}{2m} dd^T$ is the Modularity matrix or Q-Laplacian, $d = [d_1, \ldots, d_N]$ indicates the vector of the degrees of each node, $D^M \in \mathbb{R}^{k \times k}$ is a diagonal matrix with diagonal entry $D^M_{ii} = |C_i|$ where $|C_i|$ is the number of nodes in cluster C_i, and X represents the cluster indicator matrix.

[1] In principle cross-validation can be used as well.

[2] A null model of a graph, called also random graph, is defined as a graph with the same properties of the original network like the degree sequence, degree distribution etc., but with the edges placed at random.

[3] The first version of the Modularity is given for unweighted networks where the similarity matrix is called adjacency and m represent the total number of edges in the graph. However the generalization of Q-measure to weighted graphs also exists [8].

B. Proposed algorithm

Our model selection algorithm can briefly be expressed in the following way:

Algorithm MS Modularity-based model selection algorithm

Input: training set, validation set stage I, validation set stage II, positive (semi-) definite kernel function $K(x_i, x_j)$
Output: selected number of clusters k and (if any) kernel parameters

1) compute cluster indicator matrix X from the cluster results of the different models, obtained using the training set and the validation set I stage in the learning process,
2) compute the Modularity matrix $B = S - \frac{dd^T}{2m}$, where S refer to the validation set used in the II stage of the validation process
3) compute the Modularity $M = \frac{1}{2m} \text{tr}(X^T B X)$,
4) select the model (i.e. k and the kernel parameters) corresponding to the partition(s) which gives the highest Modularity value.

The training set, validation set and the two stages of the validation process have the following meaning. The training set is the matrix given as input to the kernel spectral clustering model during the training phase and it is different depending on the kind of network or the model selection criterion considered (see for example Figures 1 and 2). The validation process can be divided in two stages:

1) stage I: the cluster memberships for the validation set (data not belonging to the training set) are predicted by the model based on eq. (5)
2) stage II: the quality of the predicted memberships are judged by means of a criterion (BLF or Modularity).

For BLF we do not change representation. On the other hand for the Modularity criterion, in these two stages the validation sets involve the same data (the nodes of the graphs under study) but represented in different ways. See the next section, Table I and Figures 1 and 2 for further clarification.

It is worthwhile to notice that the definition of the Modularity function is general[4] because it does not make any assumption on the kind of the community structure of the network to detect. This feature is one of the main differences with the BLF criterion, which is optimized to detect clusters that are well separated.

IV. Data handling

Here the way of choosing the training and validation set for the BLF and Modularity criterion is discussed, both for weighted and unweighted graphs.

[4] Some modifications of the Q-measure to better detect overlapping communities or to fit in the fuzzy logic framework have also been proposed (see [10]).

A. Weighted graphs

In the analysis of the weighted networks the well known RBF kernel, characterized by the bandwidth parameter σ, is used to capture the similarity within the nodes. This point is very important and it is worthwhile to discuss, mostly for its implication in treating large graphs. For the BLF, we do not consider the affinity matrix $S \in \mathbb{R}^{N \times N}$ of the whole network as a data matrix over which to apply the RBF kernel. In fact, if N is very large, like for huge networks, the points of this data matrix would have a very high dimension leading to prohibitive computational burden for the calculation of the kernel matrix). So we take the following steps:

- we select a square submatrix from the affinity matrix of the whole network, $S \in \mathbb{R}^{C \times C}$, with $C < N$, which represents a subgraph of the whole network,
- changing our perspective, we consider now this matrix not as the affinity matrix of the subgraph, but as a data matrix of points in Euclidean space representing the training set, over which an RBF kernel is build,
- the validation set is also a data matrix with the same dimensionality of the training set.

For the Modularity, we follow the same steps as before but after having obtained the cluster results we change again our perspective. In the second stage of the validation process we jump back from the Euclidean representation to the graph representation. So now the training set is considered again as an adjacency matrix related to the training subgraph, and we take as validation set the adjacency matrix representing the remaining subgraph. This is done because the Modularity matrix B includes in its definition an affinity (square) matrix. Figure 1 summarizes this discussion.

B. Unweighted graphs

In dealing with unweighted networks a recently proposed kernel function particularly suited for the study of unweighted networks, the community kernel [11] is used to build up the similarity matrix of the graph. This kernel function does not have any parameter to tune and the similarity K_{ij} between two nodes i and j is defined as the number of edges connecting the common neighbors of these two nodes: $K_{ij} = \sum_{k,l \in \mathcal{N}_{ij}} A_{kl}$. Here $\mathcal{N}_{ij}$ is the set of the common neighbors of nodes i and j, A indicates the adjacency matrix of the graph, K is the kernel function. As a consequence, even if two nodes are not directly connected to each other, if they share many common neighbors their similarity k_{ij} will be set to a large value.

For the BLF, if we consider, in order to obtain the cluster results during the learning process, the same training and validation sets used for weighted graphs, we obtain very poor results. So now for the learning process we consider another representation for the initial graph. We think of the graph as an adjacency list regarding each row of the adjacency matrix as a source node (see section V-B). In this case the computational burden due to constructing the (training and validation) kernel matrices depends on the sparsity of the initial graph. For the Modularity, we consider in the second stage of the validation process the same validation set used for the weighted networks, representing now the nodes of the training set used in the first stage not as an adjacency list but in terms of a square adjacency matrix. Figure 2 clarifies this point.

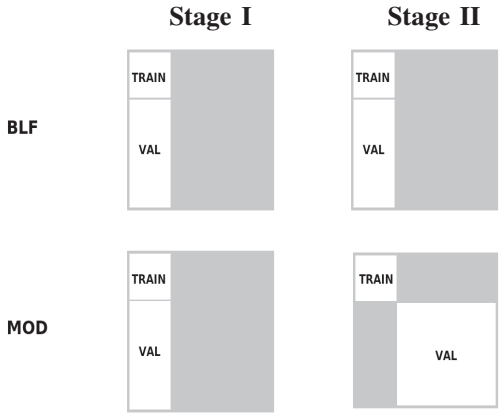


Fig. 1. Example showing the way the datasets are built up for the weighted networks: in this specific case the first 25% of the total nodes form the training set and the remaining 75% the validation set. The first row refers to BLF, the second to Modularity. The first column represents the first stage of the validation process (prediction of memberships), the second column depicts the second stage. For Modularity, in the second stage of the validation process we change again representation.

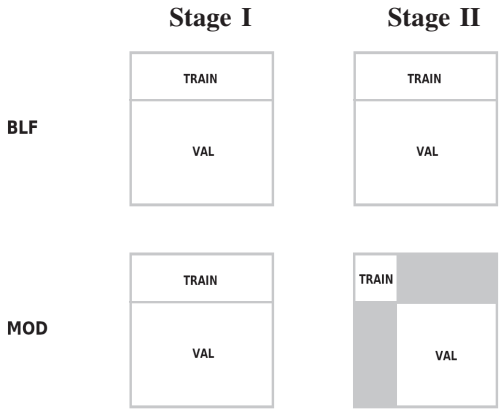


Fig. 2. Example of training and validation set used for unweighted graphs. Also in this case the first 25% of the total nodes form the training set and the remaining 75% the validation set. Like for weighted graphs, in the second stage of the validation process consisting of evaluating the quality of the predicted partitions by means of Modularity criterion, there is a change in the representation of the initial training and validation sets.

	Validation set WN	**Validation set UNWN**
Stage I	Points in an Euclidean space	Adjacency list
Stage II	Square adjacency matrix	Square adjacency matrix

TABLE I
REPRESENTATION OF THE VALIDATION SET FOR WEIGHTED (WN) AND UNWEIGHTED (UNWN) GRAPHS IN THE TWO STAGES OF THE VALIDATION PROCESS CONCERNING THE MODULARITY CRITERION.

V. Simulation Results

In this section we discuss the results of the model selection task achieved by using the method described in the last section and we also compare it with respect to the BLF criterion. Since an intrinsic characteristic of the BLF is that the values it takes for 2 clusters are often very high, we do not consider it as an useful information to take into account in order to tune the number of clusters. In all the experiments we use the 25% of the whole network as training set and the remaining 75% as validation set. Real and artificial[5] datasets are investigated. Moreover, random subsampling is used considering 10 randomization of the training (and validation) set. Thus in all the following figures the variability due to the randomization process is represented by means of boxplots, while the average values are connected by a continuous line. Finally, the significance of the detected clusters is expressed by satisfactory values of the ARI[6] index between the true memberships and the predicted memberships for two benchmark networks taken as examples (the unweighted network formed by 9 communities indicated with Net_unw9 and the weighted graph characterized by the presence of 6 clusters referred as Net_w6 in Table II).

A. Weighted networks

A weighted network is a network where the edges among nodes have weights assigned to them. In a number of real-world networks, not all ties in a network have the same capacity. In fact, ties are often associated with weights that differentiate them in terms of their strength, intensity or capacity. For example the strength of social relationships in social networks can be a function of their duration, emotional intensity, intimacy, and exchange of services. For non-social networks, weights often refer to the function performed by ties, e.g. the carbon flow between species in food webs, the number of synapses and gap junctions in neural networks or the amount of traffic flowing along connections in transportation networks.

Here two networks are analyzed:

- a benchmark network with 3 000 nodes and 148 928 edges formed by 6 communities without overlapping nodes
- an artificial graph formed by 3 000 nodes and 149 033 edges with 4 overlapping communities.

B. Unweighted networks

In an unweighted network there is no strength associated to the edges linking the nodes. Several representations can be used:

- adjacency list: it is implemented as an array of lists, with one list of destination nodes for each source node,

[5]The software provided by Fortunato related to the paper [12] is used.

[6]ARI stands for Adjusted Rand Index and it is a measure of agreement between clustering results of a model and a known grouping which acts like a ground-truth. The ARI ranges between -1 and 1 (perfect fit). For more information see [13].

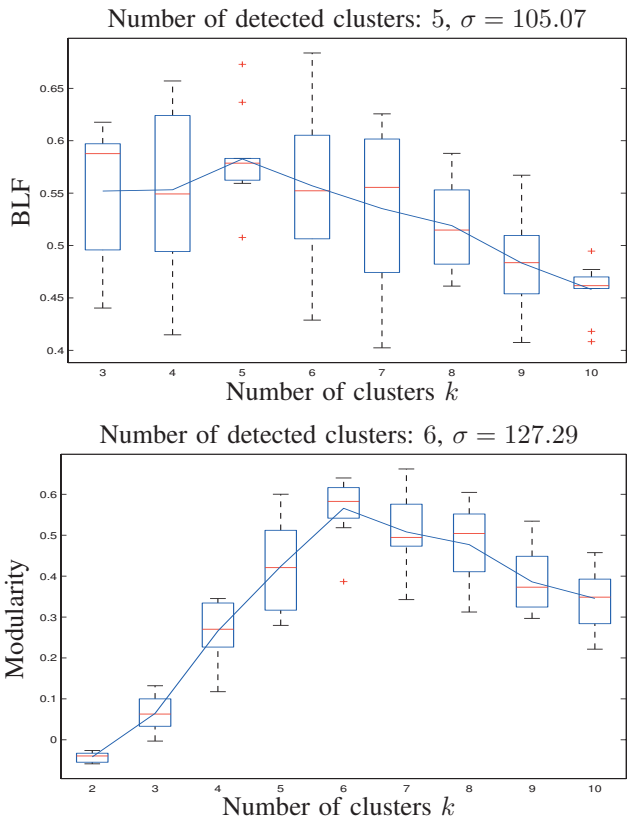


Fig. 3. Artificial network with 6 not overlapping communities

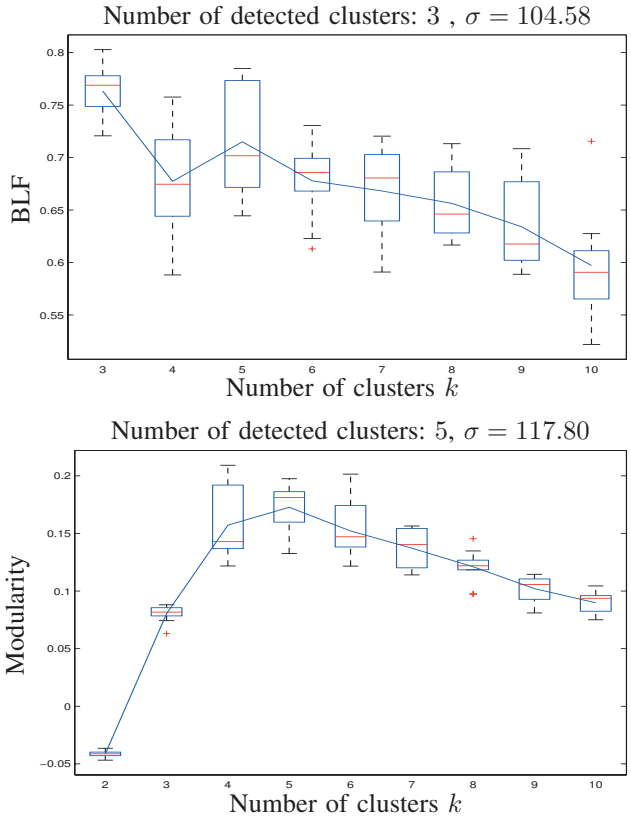


Fig. 4. Artificial network with 4 overlapping communities: both criteria seem not able to detect the 4 overlapping clusters. This is likely due to the not strong community structure testified by quite low values of Modularity corresponding to the various partitions. So, in the future it could be fruitfull to try out to use in cases like this the definition of Modularity suited for graphs with overlapping communities.

- incidence list: a variant of the adjacency list that allows for the description of the edges at the cost of additional edges,
- adjacency matrix: a two-dimensional Boolean matrix, in which the rows and columns represent source and destination vertices and entries in the matrix indicate whether an edge exists between the vertices associated with that row and column,
- incidence matrix: a two-dimensional Boolean matrix, in which the rows represent the vertices and columns represent the edges. The array entries indicate if both are related, i.e. incident.

In this section three networks are analyzed:

- the network of Western USA power grid [14] formed by 4 941 nodes and 6 594 edges,
- a benchmark network with 3 000 nodes and 22 904 edges formed by 9 communities without overlapping nodes
- an artificial graph formed by 3 000 nodes and 149 535 edges with 1 000 overlapping nodes.

In order to judge the results obtained on the real dataset, they are compared with those obtained by applying the Louvain method [3], considered as one of the fastest and accurate algorithms for community detection.

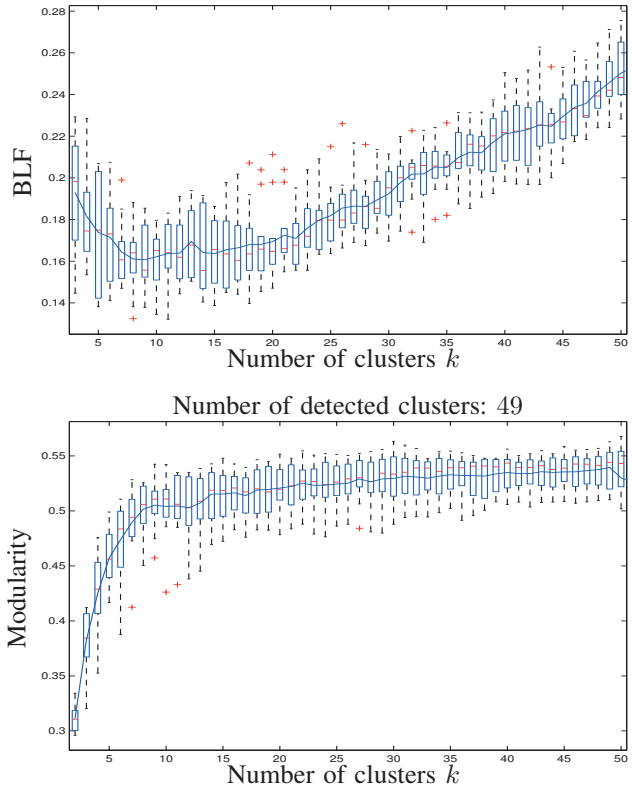


Fig. 5. Western USA power grid network: the number of communities detected by the Louvain method at the finest resolution is 41. The BLF criterion seems not able to recognize any community structure in the network, since it takes quite low values.

C. General Discussion

Our technique is tested on 2 weighted and 3 unweighted networks (real and artificial) and it is compared to another

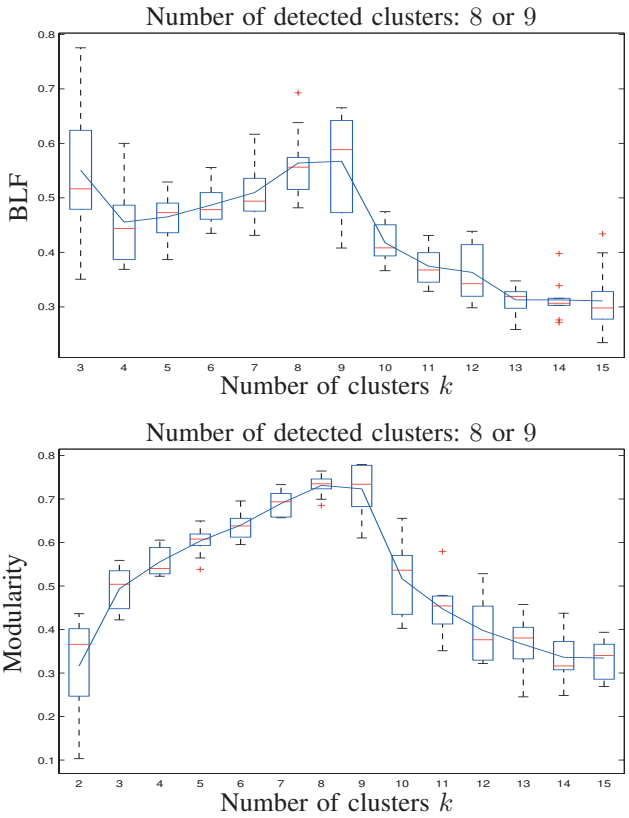


Fig. 6. Artificial network with 9 non-overlapping communities.

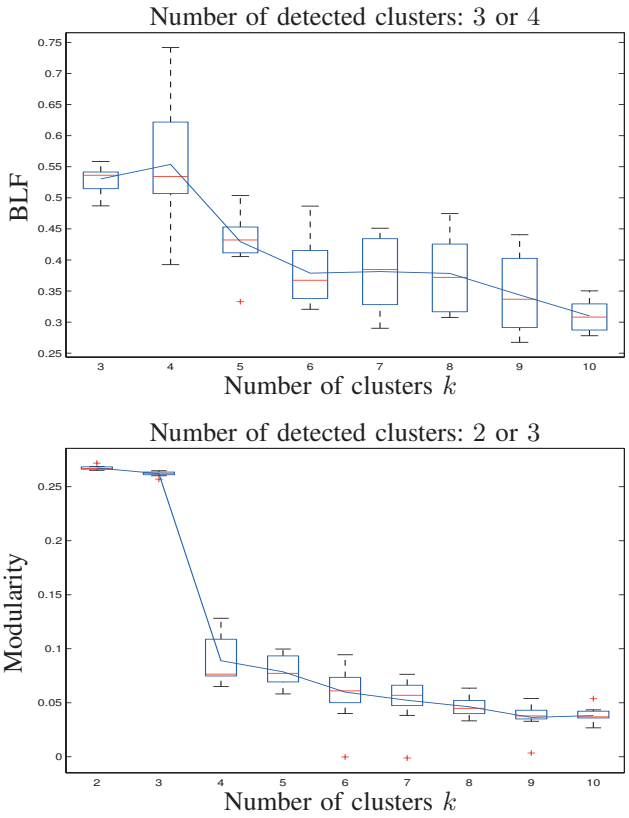
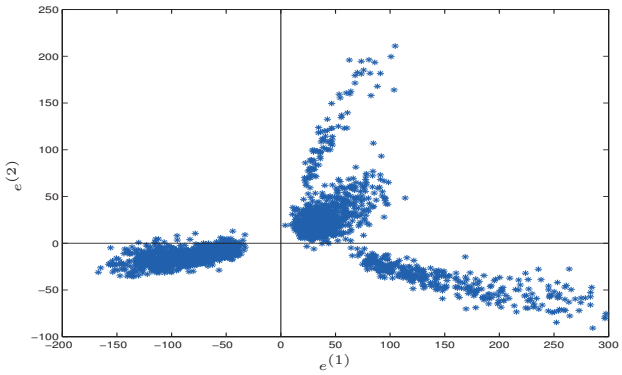
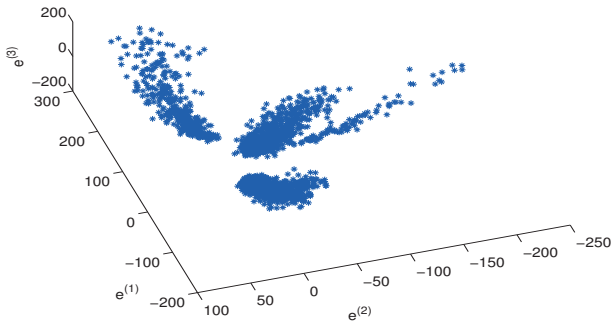


Fig. 7. Artificial network with 3 overlapping communities. It is interesting to notice that in this case BLF also suggest to select 4 clusters. On the other hand Modularity criterion correctly identify the possible presence of 3 clusters without falling in this confusion.



(a) 3 clusters



(b) 4 clusters

Fig. 8. Artificial network with 3 overlapping communities. Nodes represented in the space of the latent variables for 3 and 4 clusters. We can notice that in the case of 3 communities there are not 3 clear lines, while in the case of 4 communities the line structure is more evident. Probably the BLF criterion is considering the overlapping nodes as belonging to a separate community.

Network	ARI_BLF	ARI_mod
Net_w6	0.4384 ± 0.0715	0.5674 ± 0.1247
Net_unw9	0.8288 ± 0.1164	0.8285 ± 0.1166

TABLE II

ARI INDEX BETWEEN TRUE AND PREDICTED MEMBERSHIPS FOR TWO ARTIFICIAL NETWORKS: THE UNWEIGHTED NETWORK FORMED BY 9 COMMUNITIES (NET_UNW9) AND THE WEIGHTED GRAPH CHARACTERIZED BY THE PRESENCE OF 6 CLUSTERS (INDICATED WITH NET_W6). THE SUBSCRIPTS BLF AND MOD INDICATE THAT THE PARAMETERS k AND σ GIVEN AS INPUT TO THE KERNEL SPECTRAL CLUSTERING MODEL ARE ESTIMATED RESPECTIVELY BY BLF AND MODULARITY CRITERION. THE VALUES ARE AVERAGED ON 10 RANDOMIZATION OF THE TRAINING SET AND THE VARIABILITY IS REPRESENTED BY THE STANDARD DEVIATION.

methodology previously developed by some of the authors, that is the BLF criterion. It achieves comparable or better performances than BLF criterion on the computer generated graphs, even when the BLF criterion seems to fails, i.e. in the analysis of the synthetic network with 3 overlapping communities. Indeed in this case (see Fig. 7) from the boxplot it could seem that the BLF is slightly more likely to detect 3 communities rather than 4 (less variability), but viewing the results from another perspective (in the projections space, namely the space of the score variables $e^{(1)}, e^{(2)}, \ldots, e^{(k-1)}$), we can be convinced that this is not true. In fact in the space of the latent variables, as we already mentioned in subsection II-D, in the ideal case of perfectly separated clusters every line represents a different community (refer to [1] for more details). In Fig. 8 we compare the line structure in this space for 3 and 4 clusters (related to the average value of the BLF on the 10 runs we considered in Fig. 7), showing as in the latter case the line structure is more clear. This is an indication that the BLF criterion is probably detecting 4 clusters rather than 3. In the real network represented by the topology of the Western US Power grid, the results given by the method here proposed have a good agreement with respect to the Louvain method, taken for comparison issues. On the other hand the BLF criterion seems not able to carry out any useful information about the presence of well formed communities.

VI. CONCLUSIONS

A model selection method based on Modularity evaluation is presented. In particular, this quality function is used to judge the partitions obtained by means of out-of-sample extension on a validation set by a kernel-based spectral clustering model previously trained. In this way it is possible to find the tuning parameters (like the number of clusters in which to divide the graph under investigation and the kernel parameters) to feed into the model, namely those related to the partition(s) giving the highest value of Modularity.

The obtained results suggest that the Modularity-based criterion can be a very useful tool for tuning the parameters to feed into the kernel machines for the clustering of networks (like the number of communities to look for and, when it is the case, the kernel parameters), basically because of its general definition. Moreover, it is important to notice that the kernel spectral clustering model developed by some of the authors, conveniently filled with optimal parameters, can be used in principle to cluster very huge complex networks in a reasonable time (thanks to the out-of-sample extension), working with sparse representations. Finally, in future work instead of using random subsampling, an active subsampling technique like the Renyi Entropy criterion (see [6]) could be considered in order to select a unique small representative subgraph to use as training set, which is essential for analyzing large graphs with a low computational burden.

ACKNOWLEDGEMENTS

This work was supported by Research Council KUL: GOA/11/05 Ambiorics, GOA/10/09 MaNet , CoE EF/05/006 Optimization in Engineering(OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants;Flemish Government:FWO: PhD/postdoc grants, projects: G0226.06 (cooperative systems and optimization), G0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) research communities (WOG: ICCoS, ANMMM, MLDM); G.0377.09 (Mechatronics MPC) IWT: PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare; Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011); EU: ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940); Contract Research: AMINAL; Other:Helmholtz: viCERP, ACCM,

Bauknecht, Hoerbiger. Rocco Langone is a phd student at the K.U. Leuven, Belgium. Carlos Alzate is a postdoctoral fellow of the Research Foundation - Flanders (FWO). Johan Suykens is a professor at the K.U.Leuven, Belgium. The scientific responsibility is assumed by its authors.

REFERENCES

[1] C. Alzate and J. A. K. Suykens, "Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 335–347, February 2010.

[2] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, 2006.

[3] 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*, vol. 2008, no. 10, p. P10008, 2008.

[4] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.

[5] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[6] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[7] M. Meila and J. Shi, "A random walks view of spectral segmentation," in *Artificial Intelligence and Statistics AISTATS*, 2001.

[8] M. E. J. Newman, "Analysis of weighted networks," *Phys. Rev. E*, vol. 70, no. 5, p. 056131, Nov 2004.

[9] J. Q. Jiang, A. W. Dress, and G. Yang, "A spectral clustering-based framework for detecting community structures in complex networks," *Applied Mathematics Letters*, vol. 22, no. 9, pp. 1479 – 1482, 2009.

[10] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75 – 174, 2010.

[11] Y. Kang and S. Choi, "Kernel PCA for community detection," in *Business Intelligence Conference*, 2009.

[12] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Phys. Rev. E*, vol. 80, no. 1, p. 016118, Jul 2009.

[13] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 1, no. 2, pp. 193–218, 1985.

[14] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, no. 393, pp. 440–442, 1998.