Fast Kernel Spectral Clustering

Rocco Langone[1a], Johan A. K. Suykens[a]

[a]*KU Leuven, ESAT-STADIUS, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium.*

Abstract

Spectral clustering suffers from a scalability problem in both memory usage and computational time when the number of data instances N is large. To solve this issue, we present a fast spectral clustering algorithm able to effectively handle millions of datapoints at a desktop PC scale. The proposed technique relies on a kernel-based formulation of the spectral clustering problem, also known as kernel spectral clustering. In this framework, the Nyström approximation of the feature map of size m, with $m \ll N$, is used to solve the primal optimization problem. This leads to a reduction of time complexity from $O(N^3)$ to $O(mN)$ and space complexity from $O(N^2)$ to $O(mN)$. The effectiveness of the proposed algorithm in terms of computational efficiency and clustering quality is illustrated on several datasets.

Keywords: spectral clustering, kernel methods, big data, Nyström approximation.

1. Introduction

Data clustering represents a valuable data analysis tool in modern applications of machine learning and data mining. In many domains clustering is used to gain first insights in the data under investigation and to provide solutions to several real-life problems, such as customer segmentation in marketing campaigns, fault detection in industrial process monitoring, topic modeling in text mining, image segmentation in computer vision.

Spectral clustering (SC) [12, 13, 14, 15] has been shown to be among the most effective clustering algorithms, due to its ability of dealing with nonlinear separable problems. This effectiveness can be explained by considering that the data in the original space are mapped into a new embedding where patterns of similar points (if any) emerge more easily. This embedding is the space spanned by the eigenvectors of the Laplacian matrix, which is derived from the graph similarity matrix. A major drawback of SC is its cubic computational complexity and quadratic memory bottleneck. Furthermore, it lacks of a systematic out-of-sample property, which is only approximate. In order to face these problems some improved spectral clustering algorithms have been proposed. Examples include [9] where the data similarity matrix and clustering structure are learned simultaneously, [8] which proposes to adjust the initial data

[1]Corresponding author: rocco.langone@esat.kuleuven.be

graph itself as part of the clustering procedure, [7] where a linearity regularization is explicitly added into the objective function of SC methods to better deal with high-dimensional data, [6] which exploits a semi-supervised learning framework to perform clustering, [5] where the addition of nonnegative constraints into the minmax cut graph clustering lead to solutions that are very close to the ideal class indicator matrix, kernel spectral clustering [16, 17] (KSC). KSC allows to tackle the issues of selecting an appropriate number of clusters and predicting the memberships of new points using a kernel-based modeling approach. More precisely, KSC represents a spectral clustering model within a least-squares support vector machine [18] setting. The primal problem is formulated as a weighted kernel PCA objective and, just as in the classifier case, the binary clustering model is expressed by a hyperplane in a high dimensional space induced by a kernel. Furthermore, after training a KSC model, it is possible to assign a membership to unseen points by calculating their projections onto the eigenvectors computed in the training phase. The KSC model parameters are obtained by solving an eigenvalue problem in the dual weight space of size $N_{\text{tr}} \times N_{\text{tr}}$, being N_{tr} the number of training instances. As for SC, when N_{tr} is large the problem can become intractable. In this article we discuss a strategy to handle this issue by means of a fixed-size procedure, which has been introduced in [18] and [19] for classification and regression problems, respectively. The approach has been originally proposed in our previous work [1] and relies on the Nyström approximation [20] of the nonlinear mapping induced by the kernel matrix to solve the primal optimization problem. In particular, after computing an approximate explicit feature map through a random subset of size $m \ll N_{\text{tr}}$, the primal problem is reformulated in an unconstrained form. Its solution can be obtained by computing an eigenvalue decomposition of an $m \times m$ matrix, and the cost of calculating the cluster memberships is decreased to $O(mN)$. Compared to [1], in this paper more comparisons with other fast spectral clustering algorithms are performed, additional insights concerning the clustering performance as function of the Nyström subset size are gained and a better understanding of the produced results is attained.

The rest of this paper is organized as follows. In Section 2 some related works proposed to speed-up spectral clustering are summarized. Section 3 briefly reviews the standard KSC algorithm. Section 4 introduces the proposed approach, where a primal KSC model instead of a dual model is derived using an approximated explicit feature map. Section 5 reports the experimental results and finally some conclusions are draw in Section 6.

2. Related work

Several algorithms have been devised to speed-up spectral clustering. Power iteration clustering [21] finds a very low-dimensional embedding of a dataset using truncated power iteration on a normalized pair-wise similarity matrix, which turns out to be an effective cluster indicator. In spectral grouping using the Nyström method [22] approximate eigenvectors of the Laplacian are computed instead of the exact ones at a much lower computational cost. A closely related method has been devised in [4], where performance guarantees have also been proved theoretically. Fast approximate spectral clustering [3] use local k-means clustering and random projection tree to scale-up spectral clustering. A number of incremental methods have also been proposed,

where some initial clusters computed on a small subset of the data are modified by updating the eigenvalue system using linear algebra techniques [23], through hyperbolic smoothing [24], by means of a model-based projection [25]. In parallel spectral clustering [26] the Laplacian matrix is sparsified via retaining nearest neighbors and both memory use and computation are parallelized on distributed computers. A number of methods make use of the incomplete Cholesky decomposition to construct an approximation of the graph Laplacian and reduce the size of the related eigenvalue problem [27, 28, 29, 30]. Landmark-based spectral clustering [31] works by selecting representative data points as the landmarks and represents the original data points as the linear combinations of these landmarks. The spectral embedding of the data can then be efficiently computed with the landmark-based representation. Other approaches include consensus spectral clustering [32], vector quantization based approximate spectral clustering [33], approximate pairwise clustering [34], low-rank KPCA proposed in [11], large-scale multi-view spectral clustering [10].

2.1. The Nyström method

The Nyström method has been used to scale-up kernel-based algorithms including support vector machine, Gaussian processes, kernel principal component analysis [35]. In this realm, approximate explicit feature maps corresponding to certain kernels can serve as a basis for reducing the cost of learning nonlinear models with large datasets [36, 18]. Given a positive definite kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and a data density distribution $p(\mathbf{x})$, the Nyström approximation of order N is the feature map $\hat{\varphi} : \mathbb{R}^d \to \mathbb{R}^N$ that best approximate the kernel at points $\mathbf{x}_i$ and $\mathbf{x}_j$ sampled from $p(\mathbf{x})$. In particular, the components $\hat{\varphi}_i(\mathbf{x})$ are eigenfunctions of the kernel, which in practice are computed by replacing the integral with an empirical average [37]:

$$\hat{\varphi}_i(\mathbf{x}) = \frac{\sqrt{N}}{\lambda_i} \sum_{j=1}^{N} \mathbf{u}_{ji} k(\mathbf{x}_j, \mathbf{x}) \tag{1}$$

where λ_i and $\mathbf{u}_i$ are the eigenvalues and eigenvectors of the kernel matrix $\boldsymbol{K}$. Finally, instead of computing the eigendecomposition of the original $N \times N$ kernel matrix, their Nyström approximations $\hat{\lambda}_i$ and $\hat{\mathbf{u}}_i$ can be calculated based on the eigensystem involving the kernel matrix $\boldsymbol{K}_{m \times m}$ computed between $m \ll N$ random points selected from the original data set:

$$\hat{\varphi}_i(\mathbf{x}) \approx \sqrt{\frac{m}{\hat{\lambda}_i}} \sum_{j=1}^{m} \hat{\mathbf{u}}_{ji} k(\mathbf{x}_j, \mathbf{x}) \tag{2}$$

with $\hat{\lambda}_i = \frac{N}{m} \lambda_i^{(m)}$, $\hat{\mathbf{u}}_i = \sqrt{\frac{m}{N}} \frac{1}{\lambda_i^{(m)}} \boldsymbol{K}_{N \times m} \mathbf{u}_i^{(m)}$, $\hat{\boldsymbol{K}} = \boldsymbol{K}_{N \times m} \boldsymbol{K}_{m \times m}^{-1} \boldsymbol{K}_{N \times m}^T$. The symbol $\boldsymbol{K}_{N \times m}$ denotes the similarity between the m selected points and the N input data, and $\hat{\boldsymbol{K}}$ represents the Nyström approximation of the kernel matrix $\boldsymbol{K}$, where $\boldsymbol{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

3. Kernel Spectral Clustering

Kernel spectral clustering [17] is a formulation of the spectral clustering problem in the least squares support vector machines [18] learning framework. This setting brings two main advantages, namely a rigorous tuning procedure for the selection of a proper number of clusters and the prediction of the cluster memberships for unseen points using out-of-sample extension.

3.1. Primal space

Given a set of N datapoints $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ to be clustered in k clusters, with $\mathbf{x}_i \in \mathbb{R}^d$, the primal KSC optimization problem related to N_{tr} training data is given by the following weighted kernel PCA formulation:

$$\begin{aligned} \min_{\mathbf{w}^{(l)},\mathbf{e}^{(l)},b_l} \quad & \frac{1}{2}\sum_{l=1}^{k-1} \mathbf{w}^{(l)^T}\mathbf{w}^{(l)} - \frac{1}{2}\sum_{l=1}^{k-1} \gamma_l \mathbf{e}^{(l)^T}\mathbf{D}^{-1}\mathbf{e}^{(l)} \\ \text{subject to} \quad & \mathbf{e}^{(l)} = \boldsymbol{\Phi}\mathbf{w}^{(l)} + b_l \mathbf{1}_{N_{\text{tr}}}. \end{aligned} \tag{3}$$

Equation (3) means that one wants to find some directions $\mathbf{w}^{(l)}$ with minimal norm such that the weighted variances of the projections along these directions, i.e. $\mathbf{e}^{(l)^T}\mathbf{D}^{-1}\mathbf{e}^{(l)}$ are maximized. In (3) $\mathbf{D}^{-1} \in \mathbb{R}^{N_{\text{tr}}\times N_{\text{tr}}}$ denotes the inverse of the degree matrix $\mathbf{D}$, which is diagonal with diagonal $\mathbf{d} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{1}_{N_{\text{tr}}}$, $\boldsymbol{\Phi}$ is the $N_{\text{tr}} \times d_h$ feature matrix $\boldsymbol{\Phi} = [\varphi(\mathbf{x}_1)^T; \ldots; \varphi(\mathbf{x}_{N_{\text{tr}}})^T]$, $\varphi : \mathbb{R}^d \to \mathbb{R}^{d_h}$ indicates the mapping to a high-dimensional feature space, b_l are bias terms, γ_l are regularization constants trading-off the maximization of the weighted variances of the projections with the minimization of the L_2 norm of the vectors $\mathbf{w}^{(l)}$. The vectors $\mathbf{e}^{(l)} = [e_1^{(l)}, \ldots, e_i^{(l)}, \ldots, e_{N_{\text{tr}}}^{(l)}]^T$ indicate the clustering scores, that is the projections of the training data mapped in the feature space along the directions $\mathbf{w}^{(l)}$. For a given point $\mathbf{x}_i$ its projection $\mathbf{e}_i = [e_i^{(1)}, \ldots, e_i^{(k-1)}]$ can be computed as

$$e_i^{(l)} = \mathbf{w}^{(l)^T}\varphi(\mathbf{x}_i) + b_l, \;\; l = 1, \ldots, k-1. \tag{4}$$

In case of a new datapoint $\mathbf{x}_i^{\text{test}}$, the related clustering score can be obtained by projecting the point in the training eigenspace:

$$e_i^{(l),\text{test}} = \mathbf{w}^{(l)^T}\varphi(\mathbf{x}_i^{\text{test}}) + b_l, \;\; l = 1, \ldots, k-1. \tag{5}$$

3.2. Dual space

Since in general the feature map $\boldsymbol{\Phi}$ is unknown and can be even infinite-dimensional, from the Karush-Kuhn-Tucker (KKT) conditions for optimality of the Lagrangian associated with (3) the following dual problem can be derived:

$$\mathbf{M_D}\mathbf{D}^{-1}\boldsymbol{K}\boldsymbol{\alpha}^{(l)} = \lambda_l \boldsymbol{\alpha}^{(l)} \tag{6}$$

where $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$ indicates the kernel matrix, $\boldsymbol{M_D}$ is a centering matrix and $\lambda_l = \frac{1}{\gamma_l}$. The solutions $\boldsymbol{\alpha}^{(l)}$ allow to compute the clustering score for any data point without

explicitly knowing the expression of the feature map. In particular, the dual clustering score for a generic data point $\mathbf{x}$ can be calculated as:

$$e^{(l)}(\mathbf{x}) = \sum_{j=1}^{N_{\text{tr}}} k(\mathbf{x}_j, \mathbf{x})\alpha_j^{(l)} + b_l. \tag{7}$$

The clustering memberships can be obtained by taking the sign of (7) and using an Error Correcting Output Codes (ECOC) coding scheme.

Remark: Notice that since in general (3) is a nonconvex problem, the KKT conditions for optimality of the Lagrangian are necessary but not sufficient. However, it can be shown [17] that if we write (3) in terms of the dual variables, it equals zero when $(\boldsymbol{\alpha}^{(l)}, \lambda_l = \frac{1}{\gamma_l})$ is an eigenpair of $\mathbf{M_D}\mathbf{D}^{-1}\boldsymbol{K}$. Therefore, (3) can be interpreted as selecting from a pool of candidate solutions the $\boldsymbol{\alpha}^{(l)}$ eigenvectors corresponding to the highest eigenvalues, which maximize the second term in (3), i.e. $\frac{1}{2}\sum_{l=1}^{k-1}\gamma_l {\mathbf{e}^{(l)}}^T \mathbf{D}^{-1}\mathbf{e}^{(l)} = \frac{1}{2}\sum_{l=1}^{k-1} {\boldsymbol{\alpha}^{(l)}}^T \boldsymbol{K}\boldsymbol{\alpha}^{(l)}$.

3.3. Cluster assignment rule: ECOC procedure

After binarizing the clustering scores of all the training points a code-book $\mathcal{C}_B$ with the most frequent binary indicators is formed. For example in case of three clusters ($k = 3, l = 1, 2$) it may happen that the most occurring code-words are given by the set $\mathcal{C}_B = \{[1\ 1], [-1\ 1], [-1\ -1]\}$. In this case the ECOC scheme for a given point $\mathbf{x}$ works as follows:

- compute its projection vector as in eq.(7)
- compute $q(\mathbf{x})$ =sign($[e^{(1)}(\mathbf{x}), e^{(2)}(\mathbf{x})]$)
- suppose that $q(\mathbf{x}) = [1\ 1]$, then assign $\mathbf{x}$ to cluster 1, i. e. the closest prototype in the codebook $\mathcal{C}_B$ in terms of Hamming distance.

In the cases where sign($\mathbf{e}(\mathbf{x})$) has the same Hamming distance to more than one prototype, then $\mathbf{x}$ is assigned to the cluster whose mean value is closer to $\mathbf{e}(\mathbf{x})$ in terms of Euclidean distance. In our example, this would occur when $q(\mathbf{x})) = [1\ -1]$, whose Hamming distance from $[1\ 1]$ and $[-1\ -1]$ is the same.

3.4. Model selection

One advantage of the KSC method compared to standard spectral clustering is represented by the possibility of selecting the tuning parameters such as the number of clusters k and the kernel parameters in a systematic manner. More precisely, a model selection procedure based on grid search can be adopted: a KSC model is trained for each combination of parameters and a given cluster quality criterion is evaluated on the partitioning predicted for the validation data; finally, the parameters associated to the optimum value of the criterion are selected. The Balanced Line Fit (BLF) was first proposed in [17] to perform the tuning. It indicates the amount of collinearity between validation points belonging to the same cluster, in the space of the projections. It reaches its maximum value 1 in case of well separated clusters, represented as lines in the space of the clustering scores. An example of model selection using the BLF criterion in case of the Iris dataset is shown in Figure 1.

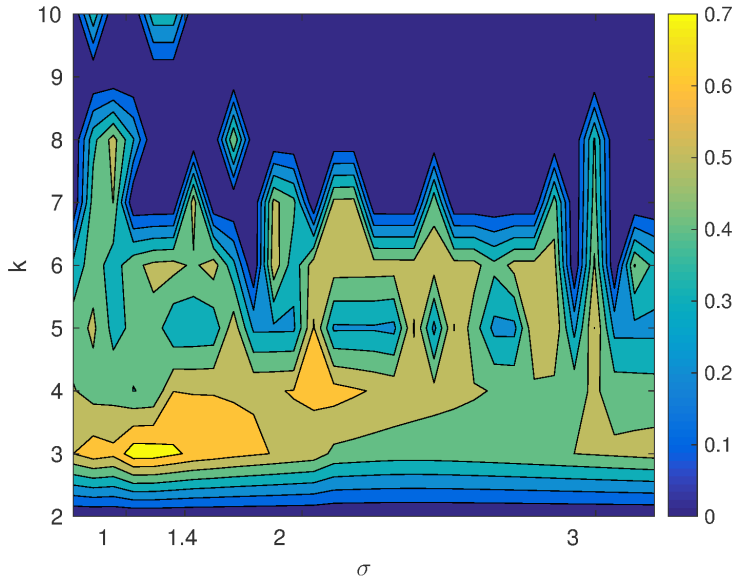


Figure 1: **KSC model selection example**. Tuning of the number of clusters k and the bandwidth σ of the Gaussian kernel in case of the Iris dataset. The colorbar indicates the BLF values range (the higher the better).

4. Proposed method

When the number of datapoints N is large, there are two possible solutions to handle the clustering problem by means of the KSC algorithm: (i) select a small number of training data $N_{\text{tr}} \ll N$, train a KSC model by solving the dual of (3), compute the cluster memberships for the remaining points by means of the out-of-sample extension property [38]; (ii) utilize a fixed-size approach by solving the primal problem, as proposed in [18] in case of classification and regression. In this paper we follow the second direction. In particular, we extend our previous work [1] on fixed-size kernel spectral clustering (FSKSC) by performing an extensive benchmarking of the method, which allows to gain more insights about the Nyström subset size to choose and the quality of the produced results.

4.1. Unconstrained reformulation

The FSKSC algorithm is based on the following unconstrained formulation of the KSC primal objective [1]:

$$\min_{\hat{\mathbf{w}}^{(l)},\hat{b}_l} \quad J(\hat{\mathbf{w}}^{(l)},\hat{b}_l) = \frac{1}{2}\sum_{l=1}^{k-1}\hat{\mathbf{w}}^{(l)^T}\hat{\mathbf{w}}^{(l)} - \frac{1}{2}\sum_{l=1}^{k-1}\gamma_l(\hat{\boldsymbol{\Phi}}\hat{\mathbf{w}}^{(l)} + \hat{b}_l\mathbf{1}_{N_{\text{tr}}})^T\hat{\boldsymbol{D}}^{-1}(\hat{\boldsymbol{\Phi}}\hat{\mathbf{w}}^{(l)} + \hat{b}_l\mathbf{1}_{N_{\text{tr}}}) \tag{8}$$

where $\hat{\boldsymbol{\Phi}} = [\hat{\varphi}(\mathbf{x}_1)^T;\ldots;\hat{\varphi}(\mathbf{x}_{N_{\text{tr}}})^T] \in \mathbb{R}^{N_{\text{tr}}\times m}$ is the approximated feature matrix, $\hat{\boldsymbol{D}} \in \mathbb{R}^{N_{\text{tr}}\times N_{\text{tr}}}$ denotes the corresponding degree matrix, and $\hat{\varphi}: \mathbb{R}^d \to \mathbb{R}^m$ indicates

a finite dimensional approximation of the feature map $\varphi(\cdot)$, γ_l are regularization parameters. The m points needed to estimate the components of $\hat{\varphi}$ can be selected at random or by means of active sampling techniques such as the Renyi entropy criterion [2]. In order to minimize (8), we can compute the necessary and sufficient condition for optimality $\nabla J(\hat{\mathbf{w}}^{(l)}, \hat{b}_l) = 0$, i.e.:

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \hat{\mathbf{w}}^{(l)}} = 0 \quad &\rightarrow \quad \hat{\mathbf{w}}^{(l)} = \gamma_l (\hat{\boldsymbol{\Phi}}^T \hat{\boldsymbol{D}}^{-1} \hat{\boldsymbol{\Phi}} \hat{\mathbf{w}}^{(l)} + \hat{\boldsymbol{\Phi}}^T \hat{\boldsymbol{D}}^{-1} \mathbf{1}_{N_{\text{tr}}} \hat{b}_l) \\ \frac{\partial \mathcal{J}}{\partial \hat{b}_l} = 0 \quad &\rightarrow \quad \mathbf{1}_{N_{\text{tr}}}^T \hat{\boldsymbol{D}}^{-1} \hat{\boldsymbol{\Phi}} \hat{\mathbf{w}}^{(l)} = -\mathbf{1}_{N_{\text{tr}}}^T \hat{\boldsymbol{D}}^{-1} \mathbf{1}_{N_{\text{tr}}} \hat{b}_l. \end{aligned}$$

After obtaining $\hat{b}_l = -\frac{\mathbf{1}_{N_{\text{tr}}}^T \hat{\mathbf{D}}^{-1} \hat{\boldsymbol{\Phi}}}{\mathbf{1}_{N_{\text{tr}}}^T \hat{\mathbf{D}}^{-1} \mathbf{1}_{N_{\text{tr}}}} \hat{\mathbf{w}}^{(l)}$ from the second equation and replacing this expression in the first equation, the following eigenvalue problem can be used to computes the optimal parameters:

$$\mathbf{R} \hat{\mathbf{w}}^{(l)} = \hat{\lambda}_l \hat{\mathbf{w}}^{(l)} \tag{9}$$

with $\hat{\lambda}_l = \frac{1}{\gamma_l}$ and $\mathbf{R} = \hat{\boldsymbol{\Phi}}^T \hat{\mathbf{D}}^{-1} \hat{\boldsymbol{\Phi}} - \frac{(\mathbf{1}_{N_{\text{tr}}}^T \hat{\mathbf{D}}^{-1} \hat{\boldsymbol{\Phi}})^T (\mathbf{1}_{N_{\text{tr}}}^T \hat{\mathbf{D}}^{-1} \hat{\boldsymbol{\Phi}})}{\mathbf{1}_{N_{\text{tr}}}^T \hat{\mathbf{D}}^{-1} \mathbf{1}_{N_{\text{tr}}}}$. It is worth noticing that we now have to solve an eigenvalue problem of size $m \times m$, which can be done very efficiently by choosing m such that $m \ll N_{\text{tr}}$. Furthermore, we can compute the diagonal of matrix $\hat{\mathbf{D}}$ as $\hat{\mathbf{d}} = \hat{\boldsymbol{\Phi}}(\hat{\boldsymbol{\Phi}}^T \mathbf{1}_m)$, without constructing the full matrix $\hat{\boldsymbol{\Phi}} \hat{\boldsymbol{\Phi}}^T$.

4.2. Cluster assignments

One of the KKT optimality conditions of the Lagrangian leading to problem (6) is $\mathbf{1}_N^T \boldsymbol{\alpha}^{(l)} = 0, l = 1, \ldots, k-1$. This means that the vectors $\boldsymbol{\alpha}^{(l)}$ and $\mathbf{e}^{(l)}$ are centered around zero and the ECOC scheme can be readily used to determine the cluster memberships. However, in case of the proposed algorithm this reasoning does not hold anymore and the optimal threshold for binarization is not available. Therefore, once the optimal model parameters $\hat{\mathbf{w}}^{(l)}, \hat{b}_l$ have been computed, the cluster memberships can be obtained by applying the k-means algorithm on the projections $\hat{e}_i^{(l)} = \hat{\mathbf{w}}^{(l)^T} \hat{\varphi}(\mathbf{x}_i) + \hat{b}_l$ for training data and $\hat{e}_i^{(l),\text{test}} = \hat{\mathbf{w}}^{(l)^T} \hat{\varphi}(\mathbf{x}_i^{\text{test}}) + \hat{b}_l$ in case of test points, as for the classical spectral clustering technique. The entire algorithm is depicted in Figure 1 and the corresponding Matlab implementation is available for download at: *http://www.esat.kuleuven.be/stadius/ADB/langone/softwareKSCFSlab.php*. Finally, Figure 2 illustrates an example of clustering obtained in case of the *Iris* dataset.

4.3. Model selection

Although the BLF criterion that comes with the KSC method can still be used for model selection, the balanced angular fit (BAF) quality metric [44] is more suitable in case of the FSKSC algorithm. In the BAF criterion for every cluster the sum of the cosine similarity between the validation points and the cluster prototype, divided by the cardinality of that cluster, is computed. These similarity values are then summed up and divided by the total number of clusters to obtain an overall measure of cluster quality. Unlike BLF, the BAF criterion does not assume that the clusters in the space of the clustering scores have a line structure (in the ideal case). Therefore, it can better deal with noisy projections as the ones resulting from the Nyström approximation.

Algorithm 1: FSKSC [1]

Input : training set $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N_{\text{tr}}}$, Test set $\mathcal{D}_{\text{test}} = \{\mathbf{x}_i\}_{i=1}^{N_{\text{test}}}$.
Settings : size Nyström subset m, kernel parameter σ, number of clusters k
Output : $\mathbf{q}$ and $\mathbf{q}_{\text{test}}$ vectors of predicted cluster memberships.

/* ***Approximate feature map:*** */
Compute $\boldsymbol{K}_{m\times m}$
Compute [$\mathbf{U}$,$\mathbf{\Lambda}$] = SVD($\boldsymbol{K}_{m\times m}$)
Compute $\hat{\mathbf{\Phi}}$ by means of (2)

/* ***Training:*** */
Compute $\mathbf{R} = \hat{\mathbf{\Phi}}^T \hat{\mathbf{D}}^{-1} \hat{\mathbf{\Phi}} - \frac{(\mathbf{1}_{N_{\text{tr}}}^T \hat{\mathbf{D}}^{-1} \hat{\mathbf{\Phi}})^T (\mathbf{1}_{N_{\text{tr}}}^T \hat{\mathbf{D}}^{-1} \hat{\mathbf{\Phi}})}{\mathbf{1}_{N_{\text{tr}}}^T \hat{\mathbf{D}}^{-1} \mathbf{1}_{N_{\text{tr}}}}$
Solve $\mathbf{R}\hat{\mathbf{w}}^{(l)} = \hat{\lambda}_l \hat{\mathbf{w}}^{(l)}$
Compute $\hat{\mathbf{e}}^{(l)} = \hat{\mathbf{\Phi}} \hat{\mathbf{w}}^{(l)} + \hat{b}_l \mathbf{1}_N$ and $\mathbf{E} = [\hat{\mathbf{e}}^{(1)}, \ldots, \hat{\mathbf{e}}^{(k-1)}]$
[$\mathbf{q}$,$\mathbf{C}_{\text{tr}}$] = kmeans($\mathbf{E}$,k)

/* ***Test:*** */
Compute $\hat{\mathbf{e}}^{(l)} = \hat{\mathbf{\Phi}}_{\text{test}} \hat{\mathbf{w}}^{(l)} + \hat{b}_l \mathbf{1}_{N_{\text{test}}}$ and $\mathbf{E}_{\text{test}} = [\hat{\mathbf{e}}_{\text{test}}^{(1)}, \ldots, \hat{\mathbf{e}}_{\text{test}}^{(k-1)}]$
$\mathbf{q}_{\text{test}}$ = kmeans($\mathbf{E}_{\text{test}}$,k,'start',$\mathbf{C}_{\text{tr}}$)

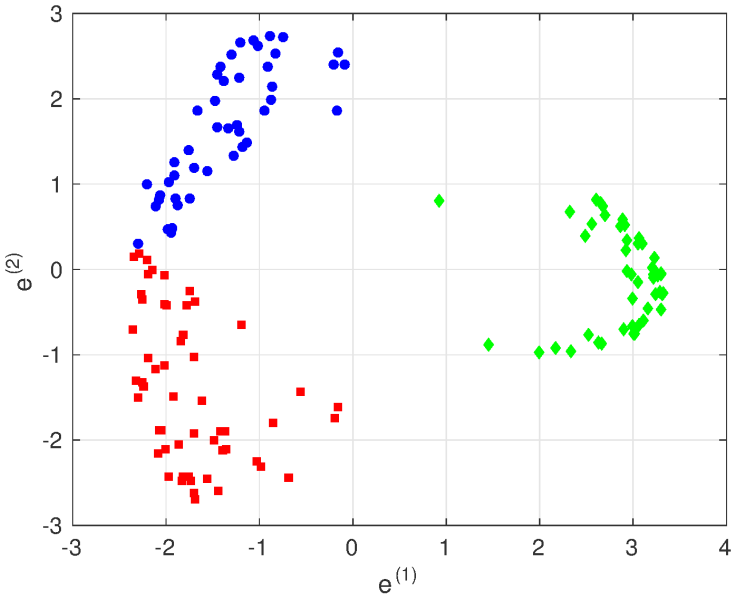


Figure 2: **FSKSC embedding**. Iris data points represented in the space of the projections. The different colors relate to the detected clusters.

4.4. Computational complexity

The computational complexity of the proposed algorithm depends mainly on the size m of the Nyström subset used to construct the approximate feature map $\hat{\mathbf{\Phi}}$. More precisely, the total time complexity is approximately $O(m^3) + O(mN_{\text{tr}}) + O(mN_{\text{test}})$, which is the time needed to solve (9) and to compute the training and test clustering scores. Moreover, the space complexity is $O(m^2) + O(mN_{\text{tr}}) + O(mN_{\text{test}})$, which refers to the construction of matrix $\mathbf{R}$ and of the training and test feature matrices $\hat{\mathbf{\Phi}}$ and $\hat{\mathbf{\Phi}}_{\text{test}}$.

5. Experiments

In this Section the proposed algorithm is tested on several datasets and compared with k-means (KM) [39], landmark-based spectral clustering (LSC) [31] and self-tune parallel spectral clustering (PSC) [26]. In LSC a small number of representative data points are selected as the landmarks and the original data points are represented as the linear combinations of these landmarks. As a consequence, thanks to this landmark-based representation, the spectral embedding of the data can be efficiently computed and the algorithm is able to scale linearly with the problem size. The PSC approach works by sparsifying the dense similarity matrix through a t-nearest neighbor technique and by then using a sparse parallelized eigensolver.

5.1. Performance evaluation

The clustering performance is assessed by means of two quality metrics, namely the Davies-Bouldin (DB) [40] criterion and the adjusted rand index (ARI [41]). The first measures the separation between each pair of clusters (in terms of between cluster scatter) and, within each cluster, how tightly grouped all the data are (in terms of within cluster scatter). The ARI index measures the agreement between two partitions and is usually utilized to assess the correlation between the outcome of a clustering algorithm and the available ground-truth. The datasets used in the experiments are listed in Table 1 and mainly comprise databases available at the UCI repository [42]. Although strictly speaking they relate to classification problems, they can also be used to evaluate the performance of clustering algorithms (where the labels play the role of the ground-truth), in view of the cluster assumption. The latter states that if points are in the same cluster they are likely to be of the same class [43].

5.2. Settings

In all the experiments we have used the following settings. For all the methods the number of clusters k has been set equal to the number of classes. In case of the proposed algorithm, the Gaussian kernel defined as $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma'^2})$ is used to induce the nonlinear mapping. In addition, a grid search procedure using the balanced angular fit (BAF) [44] as quality criterion is used to select an optimal bandwidth σ. An example of tuning using BAF is shown in Figure 3 in case of the S1 dataset. Also, 80% of data forms the training set and 20% the test set. The Nyström subset size has been set to $m = 100$ for the small databases and $m = 150$ for the

Table 1: Datasets

Size	Dataset	N	d
Small	Iris	150	4
	Ecoli	336	8
	Libras	360	91
	Dermatology	366	33
	Vowel	528	10
	Coil20	1440	1024
	Spambase	4601	57
	S1	5000	2
	S4	5000	2
Medium	Opt Digits	5620	64
	USPS	9298	256
	Pen Digits	10992	16
	Magic	19020	11
	RCV1	20242	1960
	Shuttle	58000	9
	MNIST	70000	784
Large	Skin	245057	3
	Covertype	581012	54
	GalaxyZoo	667944	9
	Susy	5000000	18
	Higgs	11000000	28

medium and large datasets. These choice has been empirically observed to represent a good trade-off between cluster quality and computation time. In this realm, Figure 4 shows the variation of the clustering performance with respect to m in case of the PenDigits dataset. Finally, for a fair comparison, the number of landmark points in the LSC algorithm and the number of nearest neighbors in the PSC method have been set equal to m.

5.3. Discussion

Table 2 reports the performance of the proposed algorithm, k-means (KM), LSC and PSC in terms of ARI and DB average values over 20 runs. In general it can be observed that FSKSC and PSC tend to be superior in terms of ARI and k-means reaches the highest performance according to the DB index. This results can be explained by considering that k-means works by minimizing the distortion function defined by the the sum of the squared distances between each observation and its closed centroid. In practice, it seeks clusters that are as much compact as possible, which is also measured by the within cluster scatter term in the definition of the DB index. However, when the clustering structure lies in a lower dimensional manifold, the DB criterion can fail to detect it. In these cases, such as in most of the datasets shown in Table 2, spectral clustering methods tend to be superior to k-means, hence the higher ARI values. Figure 5 illustrates the execution time of the various algorithms for the medium and large datasets. It is easy to notice that the proposed method is very efficient and outperforms the other approaches in case of the largest databases. Therefore, it represents a valuable tool for clustering large-scale datasets.

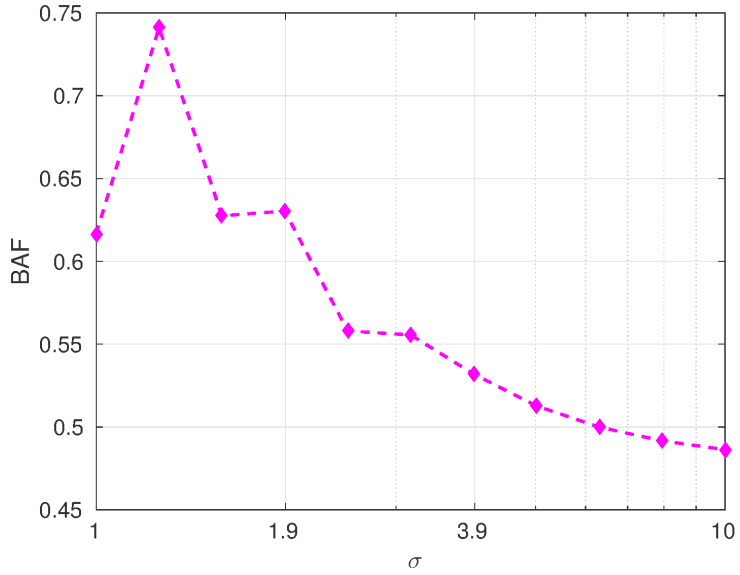


Figure 3: **Selection of the Gaussian kernel bandwidth**. Tuning of the σ parameter using the BAF criterion for the S1 dataset (the higher the better) in case of the FSKSC algorithm.

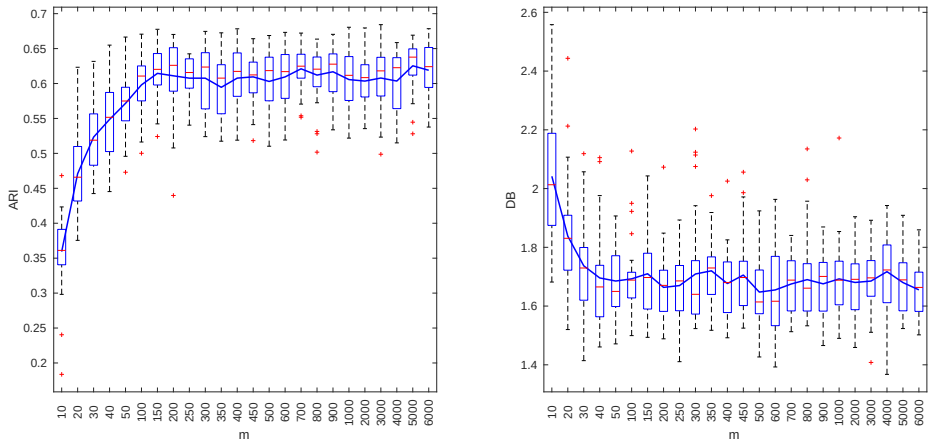


Figure 4: **Choice of the Nyström subset size**. Clustering performance VS the Nyström subset size m in case of the Pendigits dataset.

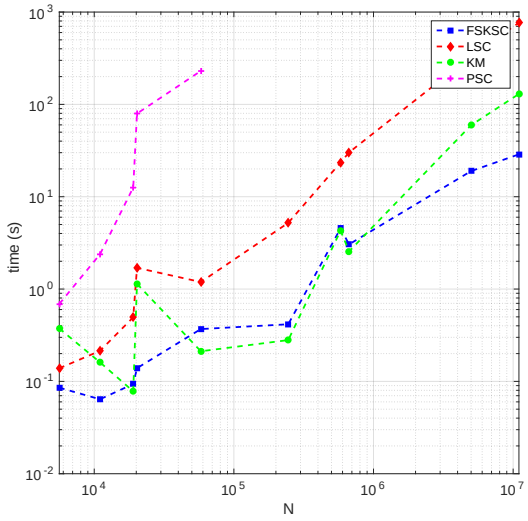


Figure 5: **Efficiency evaluation**. Runtime of FSKSC (train + test), LSC, PSC and and k-means algorithms in the analysis of medium and large datasets.

Dataset	**FSKSC**		**KM**		**LSC**		**PSC**	
	ARI	**DB**	**ARI**	**DB**	**ARI**	**DB**	**ARI**	**DB**
Iris	**0.64**	0.85	0.57	**0.83**	**0.64**	1.03	**0.64**	0.84
Ecoli	**0.50**	1.57	**0.50**	**1.17**	0.35	1.80	0.39	1.29
Libras	0.32	1.46	0.29	**1.32**	0.32	1.53	**0.34**	**1.32**
Dermatology	0.83	1.87	0.69	1.91	0.82	1.91	**0.89**	**0.59**
Vowel	**0.12**	1.67	0.09	**1.60**	0.09	1.72	0.09	1.63
Coil20	0.61	1.87	0.53	**1.82**	0.56	2.01	**0.64**	1.86
Spambase	0.38	3.87	0.22	**1.83**	**0.44**	3.97	0.01	1.05
S1	0.96	0.40	0.89	0.49	0.90	0.78	**0.98**	**0.36**
S4	**0.66**	**0.67**	0.64	0.68	0.61	0.84	0.64	0.68
Opt Digits	0.52	3.12	0.52	**1.93**	**0.69**	2.28	0.65	2.07
USPS	0.48	2.23	0.43	2.09	0.48	2.16	**0.58**	**2.06**
Pen Digits	0.61	1.63	0.57	**1.43**	**0.66**	1.58	**0.66**	1.45
Magic	**0.04**	3.28	0.006	1.43	0.02	**1.42**	0.00	1.43
RCV1	**0.08**	2.03	0.008	**0.67**	0.06	2.03	0.07	1.92
Shuttle	0.29	2.00	**0.35**	**0.75**	0.11	1.22	0.21	1.49
MNIST	0.31	3.52	0.28	**3.08**	0.36	3.58	**0.42**	3.49
Skin	0.03	**0.67**	-0.03	0.69	**0.09**	1.20	NA	NA
Covertype	**0.07**	3.85	0.05	**1.89**	0.05	2.78	NA	NA
GalaxyZoo	0.25	1.69	**0.27**	**1.12**	0.23	1.46	NA	NA
Susy	**0.12**	2.17	0.11	**2.08**	0.03	2.27	NA	NA
Higgs	**0.008**	3.34	0.006	**2.68**	0.004	3.87	NA	NA

Table 2: **Clustering performance**. Comparison of the proposed FSKSC approach against k-means, LSC and PSC algorithms in terms of mean ARI (the higher the better) and mean DB (the lower the better) over 20 runs. Best performance in bold. NA means either that the execution time is above 48 hours or that the available memory is not sufficient for the algorithm to run.

6. Conclusions

In this paper we have presented an efficient and accurate spectral clustering algorithm. The proposed approach, which is devised in a kernel-based setting, uses the Nyström approximation of the feature map to solve the primal optimization problem characterizing a kernel spectral clustering model. This leads to a reduction of time and space complexity compared to standard spectral clustering because the construction of the Laplacian matrix and its eigendecomposition are avoided. A number of experiments performed on well-known benchmarks show the usefulness of the proposed algorithm.

Acknowledgments

EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC AdG A-DATADRIVE-B (290923). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. Research Council KUL: CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity), PhD/Postdoc grant iMinds Medical Information Technologies SBO 2015 IWT: POM II SBO 100031 Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017.

References

[1] R. Langone, R. Mall, V. Jumutc, J. A. K. Suykens, Fast in-memory spectral clustering using a fixed-size approach, in: Proceedings of the 24 European Symposium on Artificial Neural Networks, Computational Intelligence and machine Learning (ESANN), 2016, pp. 557-562.

[2] A. Rényi, On measures of information and entropy, in: Proceedings of the fourth Berkeley Symposium on Mathematics, Statistics and Probability, 1960. pp. 547-561.

[3] D. Yan, L. Huang, M. I. Jordan, Fast approximate spectral clustering, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), 2009, pp. 907-916.

[4] A. Choromanska, T. Jebara, Y. Kim, M. Mahesh, C. Monteleoni, Fast Spectral Clustering via the Nyström Method, in. Proceedings of the 24th International Conference on Algorithmic Learning Theory (ALT), 2013, pp. 367-381.

[5] F. Nie, C. H. Q. Ding, D. Luo, H. Huang, Improved MinMax Cut Graph Clustering with Nonnegative Relaxation, ECML/PKDD (2) 2010, pp. 451-466.

[6] F. Nie, D. Xu, X. Li, Initialization Independent Clustering With Actively Self-Training Method, IEEE Transactions on Systems, Man, and Cybernetics, Part B 42(1) (2012), pp. 17-27.

[7] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, C. Zhang, Spectral Embedded Clustering: A Framework for In-Sample and Out-of-Sample Spectral Clustering, IEEE Transactions on Neural Networks 22(11) (2011), pp. 1796-1808.

[8] F. Nie, X. Wang, M. I. Jordan, H. Huang, The Constrained Laplacian Rank Algorithm for Graph-Based Clustering, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 1969-1976.

[9] F. Nie, X. Wang, H. Huang, Clustering and projected clustering with adaptive neighbors, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), 2014, pp. 977-986.

[10] Y. Li, F. Nie, H. Huang, J. Huang, Large-scale multi-view spectral clustering via bipartite graph, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2750-2756.

[11] C. Zhang, F. Nie and S. Xiang, A general kernelization framework for learning algorithms based on kernel {PCA}, Neurocomputing, 73 (4-6) (2010), pp. 959–967.

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

[13] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in Neural Information Processing Systems, 2002, pp. 849–856.

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

[15] H. Jia, S. Ding, X. Xu, R. Nie, The latest research progress on spectral clustering, Neural Computing and Applications 24 (7-8) (2014) 1477–1486.

[16] R. Langone, R. Mall, C. Alzate, J. A. K. Suykens, Unsupervised Learning Algorithms, Springer International Publishing, 2016, Ch. Kernel Spectral Clustering and Applications, pp. 135–161.

[17] 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) (2010) 335–347.

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

[19] K. De Brabanter, J. De Brabanter, J. A. K. Suykens, B. De Moor, Optimized fixed-size kernel models for large data sets, Computational Statistics & Data Analysis 54 (6) (2010) 1484–1504.

[20] C. Baker, The numerical treatment of integral equations, Clarendon Press, Oxford, 1977.

[21] F. Lin, W. W. Cohen, Power iteration clustering, in: International Conference on Machine Learning, 2010, pp. 655–662.

[22] C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral grouping using the Nyström method, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2) (2004) 214–225.

[23] H. Ning, W. Xu, Y. Chi, Y. Gong, T. S. Huang, Incremental spectral clustering with application to monitoring of evolving blog communities., in: SIAM International Conference on Data Mining, 2007, pp. 261–272.

[24] A. M. Bagirov, B. Ordin, G. Ozturk, A. E. Xavier, An incremental clustering algorithm based on hyperbolic smoothing, Computational Optimization and Applications 61 (1) (2014) 219–241.

[25] R. Langone, O. M. Agudelo, B. De Moor, J. A. K. Suykens, Incremental kernel spectral clustering for online learning of non-stationary data, Neurocomputing 139 (0) (2014) 246–260.

[26] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, E. Chang, Parallel spectral clustering in distributed systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (3) (2011) 568–586.

[27] C. Alzate, J. A. K. Suykens, Sparse kernel models for spectral clustering using the incomplete Cholesky decomposition, in: Proc. of the 2008 International Joint Conference on Neural Networks (IJCNN 2008), 2008, pp. 3555–3562.

[28] K. Frederix, M. Van Barel, Sparse spectral clustering method based on the incomplete Cholesky decomposition, Journal of Computational and Applied Mathematics 237 (1) (2013) 145–161.

[29] M. Novak, C. Alzate, R. Langone, J. A. K. Suykens, Fast kernel spectral clustering based on incomplete Cholesky factorization for large scale data analysis, Internal Report 14-119, ESAT-SISTA, KU Leuven (Leuven, Belgium) (2014) 1–44.

[30] R. Langone, M. Van Barel, J. A. K. Suykens, Entropy-based incomplete cholesky decomposition for a scalable spectral clustering algorithm: Computational studies and sensitivity analysis, Entropy 18 (5) (2016) 182.

[31] X. Chen, D. Cai, Large scale spectral clustering with landmark-based representation, in: AAAI Conference on Artificial Intelligence, 2011.

[32] D. Luo, C. Ding, H. Huang, F. Nie, Consensus spectral clustering in near-linear time, in: International Conference on Data Engineering, 2011, pp. 1079–1090.

[33] K. Taşdemir, Vector quantization based approximate spectral clustering of large datasets, Pattern Recognition 45 (8) (2012) 3034–3044.

[34] L. Wang, C. Leckie, R. Kotagiri, J. Bezdek, Approximate pairwise clustering for large data sets via sampling plus extension, Pattern Recognition 44 (2) (2011) 222–235.

[35] S. Sun, J. Zhao, J. Zhu, A review of Nyström methods for large-scale machine learning, Information Fusion 26 (C) (2015) 36–48.

[36] A. Vedaldi, A. Zisserman, Efficient additive kernels via explicit feature maps, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (3) (2012) 480–492.

[37] C. K. I. Williams, M. Seeger, Using the Nyström method to speed up kernel machines, in: Advances in Neural Information Processing Systems, 2001.

[38] R. Mall, R. Langone, J. A. K. Suykens, Multilevel hierarchical kernel spectral clustering for real-life large scale complex networks, PLoS ONE 9 (6) (2014) e99966.

[39] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, 1967, pp. 281–297.

[40] D. L. Davies, D. W. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (2) (1979) 224–227.

[41] L. Hubert, P. Arabie, Comparing partitions, Journal of Classification 1 (2) (1985) 193–218.

[42] A. Frank, A. Asuncion, UCI machine learning repository, http://archive.ics.uci.edu/ml, (2010).

[43] O. Chapelle, B. Schölkopf, A. Zien (Eds.), Semi-Supervised Learning, MIT Press, Cambridge, MA, 2006.

[44] R. Mall, R. Langone, J. A. K. Suykens, Kernel spectral clustering for big data networks, Entropy (Special Issue on Big Data) 15 (5) (2013) 1567–1586.