Sparse Kernel Spectral Clustering Models for Large-Scale Data Analysis

Carlos Alzate*, Johan A. K. Suykens

Katholieke Universiteit Leuven, Department of Electrical Engineering ESAT-SCD-SISTA, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

Abstract

Kernel spectral clustering has been formulated as a primal - dual optimization setting allowing natural extensions to out-of-sample data together with model selection in a learning framework. This becomes important for predictive purposes and for good generalization capabilities. The clustering model is formulated in the primal in terms of mappings to high dimensional feature spaces typical of support vector machines and kernel-based methodologies. The dual problem corresponds to an eigenvalue decomposition of a centered Laplacian matrix derived from pairwise similarities within the data. The out-of-sample extension can also be used to introduce sparsity and to reduce the computational complexity of the resulting eigenvalue problem. In this paper, we propose several methods to obtain sparse and highly sparse kernel spectral clustering models. The proposed approaches are based on structural properties of the solutions when the clusters are well formed. Experimental results with difficult toy examples and images show the applicability of the proposed sparse models with predictive capabilities.

Keywords: kernel spectral clustering, sparseness, large-scale data, kernel methods

1. Introduction

Classical spectral clustering [1, 2, 3, 4] has proven successful in difficult clustering tasks where classical methods such as k-means fail. These techniques are relaxations of NP-hard problems arising from graph partitioning problems [1]. The relaxations take the form of eigenvalue problems of a Laplacian matrix derived from the graph which contains pairwise similarity values. The clustering information is contained in certain eigenvectors of the Laplacian matrix. However, converting this information into cluster indicators is not straightforward and several methods have been proposed to tackle this issue [2, 3, 5, 6]. Another issue with classical spectral clustering relies on selecting good values for the similarity function. This is a critical task since the performance of the method depends greatly on the choice of the similarity and its parameters, even though several heuristics and rules of thumb exist. Computing the cluster membership of new points is also problematic in classical spectral clustering due to the lack of an underlying model. In the case of large-scale data, approximations such as the Nyström method [7] are needed to reduce the size of the eigenvalue problem.

A new spectral clustering algorithm based on a weighted version of kernel PCA has been recently introduced [8]. This approach is formulated in a primal and dual optimization framework allowing the extension of the clustering model to out-of-sample points via projections onto the eigenvectors. The formulation also allows model selection in a learning framework with training, validation and test stages which is important for predictive purposes and good generalization capabilities. The out-of-sample extension can also be used for reducing the computational complexity by training a model on a subset of the full data set and inferring the cluster membership of the remaining unseen data points. The clustering model is expressed in terms of non-sparse kernel expansions where every training data point contributes. In this work, we propose different methods in order to obtain sparse and highly sparse model representations. The objective is to approximate the clustering model such that it depends only on a very reduced number of data points while still providing highly performant models. The proposed sparse methods exploit structural properties of the projections when the clusters are well formed. These methodologies are based on a reduced set approach to approximate the projections. This work is an extension of [9]. This paper is organized as follows. Section 2 contains preliminaries of classical spectral clustering methods. Section 3 contains a summary of the kernel spectral clustering method introduced in [8]. In Section 4, we propose different methods in order to achieve sparse and highly sparse models. Section 5 contains the experimental results with difficult toy problems and an image segmentation problem, and in Section 6, we give conclusions.

2. Preliminaries

In graph theory, a set of training data points $\mathcal{D} = \{x_i\}_{i=1}^N$, $x_i \in \mathbb{R}^d$ and a similarity (also called affinity) measure $a_{ij} \in \mathbb{R}^+$ between each pair of points x_i and x_j define an undirected graph if the similarity measure is symmetric $a_{ij} = a_{ji}$. The vertices of the graph are represented by the data points and the weight of

*Corresponding author.

Email addresses: carlos.alzate@esat.kuleuven.be (Carlos Alzate), johan.suykens@esat.kuleuven.be (Johan A. K. Suykens)

the edges is determined by a_{ij}. The degree of the i-th vertex is the sum of the edge weights connected to it: $\deg_i = \sum_{j=1}^{N} a_{ij}$ and the degree matrix D can then be defined as the diagonal matrix $D = \text{diag}(\deg_1; \ldots; \deg_N)$.

2.1. The k-way Normalized Cut Relaxation

A classical problem in graph theory is the so called k-way graph partitioning problem. The objective is to split the graph into k disjoint sets $\mathcal{A}_1, \ldots, \mathcal{A}_k$ with $\mathcal{A}_1 \cap \mathcal{A}_2 \cap \ldots \cap \mathcal{A}_k = \emptyset$ and $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \ldots \cup \mathcal{A}_k = \mathcal{D}$ such that data points (vertices of the graph) in the same set are more similar than data points in different sets. This problem is NP-hard due to its combinatorial nature. However, useful relaxations have been proposed in the literature such as the k-way normalized cut relaxation [10]:

$$\begin{aligned} &\max_{\tilde{G}} \text{Tr}(\tilde{G}^T \hat{L} \tilde{G}) \\ &\text{s.t. } \tilde{G}^T \tilde{G} = I_k, \end{aligned} \tag{1}$$

where $\hat{L} = D^{-1/2} A D^{-1/2}$ is a normalized Laplacian, $\tilde{G} \in \mathbb{R}^{N \times k}$ is a relaxed cluster indicator matrix and I_k is the $k \times k$ identity matrix. This maximization problem is a special form of Fan's theorem [11, 12, 13]. The optimal solution of the relaxed problem is given by:

$$\tilde{G}^\star = BR_2$$

where $B = [b^{(1)}, \ldots, b^{(k)}] \in \mathbb{R}^{N \times k}$ is any orthonormal basis of the k-th principal subspace of $\hat{L}$ and $R_2 \in \mathbb{R}^{k \times k}$ is an arbitrary orthogonal matrix [11, 6]. In other words, the relaxed solution is contained in the eigenvectors corresponding to the k largest eigenvalues of the normalized Laplacian $\hat{L} = D^{-1/2} A D^{-1/2}$ but the eigenvectors can have arbitrary rotations. To obtain cluster indicators from the eigenvectors of the normalized Laplacian, several methods have been proposed. Typical approaches include recursive binary cuts [2], reclustering [2] and rounding [6].

2.2. Markov Random Walks

A random walk on a graph consists of random jumps from vertex to vertex. Many properties of spectral clustering methods can be expressed in terms of a transition matrix P obtained by normalizing the affinity matrix 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. This transition matrix can be defined as $P = D^{-1}A$. The corresponding eigenvalue problem becomes $Pr = \xi r$. The eigenvalues of P are identical to the eigenvalues of the k-way normalized cut Laplacian $\hat{L}$ and the eigenvectors are related by $r^{(i)} = D^{-1/2} b^{(i)}, i = 1, \ldots, N$ [14]. Maximizing the random walks algorithm can be interpreted as finding a partition of the graph in such a way that the random walk remains most of the time in the same cluster with few jumps to the other cluster. An important concept introduced in [15] is the piecewise constant property: A piecewise constant vector $\alpha = [\alpha_1, \ldots, \alpha_N] \in \mathbb{R}^N$ relative to a partition of $\Delta = \{\mathcal{A}_p\}_{p=1}^k$ and a dataset $\mathcal{D} = \{x_i\}_{i=1}^N$ is defined as

$$\alpha_j = c_p, \text{ for all } x_j \in \mathcal{A}_p, p = 1, \ldots, k,$$

where c_p is a constant value corresponding to the p-th cluster. In the case of a graph with k disconnected subgraphs, the stochastic matrix P will have k eigenvalues equal to 1 and the corresponding eigenvectors will be piecewise constant on the underlying partitioning[1]. This means that the clusters in the original input space are represented as single points in the k-dimensional space spanned by the eigenvectors with eigenvalue 1. This is the main motivation for using k-means as a reclustering approach since the clusters in the new representation are very localized.

3. Kernel Spectral Clustering

In this Section, we summarize the kernel spectral clustering method introduced in [8]. This methodology is cast in a primal dual optimization setting and can be interpreted as a weighted form of kernel PCA allowing model selection and out-of-sample extensions with predictive capabilities.

3.1. Primal - Dual Formulation

Given training data $\mathcal{D} = \{x_i\}_{i=1}^N, x_i \in \mathbb{R}^d$ and the number of desired groups k, the following clustering model can be created:

$$e_i^{(l)} = w^{(l)^T} \varphi(x_i) + b_l, i = 1, \ldots, N, l = 1, \ldots, n_e \tag{2}$$

where $\varphi : \mathbb{R}^d \to \mathbb{R}^{d_h}$ is the mapping to a high-dimensional feature space and b_l are bias terms. The projected variables $e^{(l)} = [e_1^{(l)}; \ldots; e_N^{(l)}]$ represent the latent variables of a set of n_e binary cluster indicators obtained through $\text{sign}(e^{(l)}), l = 1, \ldots, n_e$ which can be encoded to obtain the final k groups. The primal problem is then formulated as [8]:

$$\begin{aligned} &\min_{w^{(l)}, e^{(l)}, b_l} \frac{1}{2N} \sum_{l=1}^{n_e} \gamma_l e^{(l)^T} V e^{(l)} - \frac{1}{2} \sum_{l=1}^{n_e} w^{(l)^T} w^{(l)} \\ &\text{s.t. } e^{(l)} = \Phi w^{(l)} + b_l 1_N, l = 1, \ldots, n_e \end{aligned} \tag{3}$$

where $\gamma_l \in \mathbb{R}^+$ are regularization parameters, $V > 0$ is a $N \times N$ user-defined weighting matrix, 1_N is a vector of N ones and $\Phi \in \mathbb{R}^{N \times d_h}$ contains the mapped training points in the rows:

$$\Phi = \begin{bmatrix} \varphi(x_1)^T \\ \vdots \\ \varphi(x_N)^T \end{bmatrix}.$$

The Lagrangian of the constrained optimization problem (3) becomes

$$\begin{aligned} \mathcal{L}(w^{(l)}, e^{(l)}, b_l) = &\frac{1}{2N} \sum_{l=1}^{n_e} \gamma_l e^{(l)^T} V e^{(l)} - \frac{1}{2} \sum_{l=1}^{n_e} w^{(l)^T} w^{(l)} \\ &- \sum_{l=1}^{n_e} \alpha^{(l)} (e^{(l)} - \Phi w^{(l)} - b_l 1_N) \end{aligned}$$

[1] If the k subgraphs have few edges with low weight between them, the eigenvectors with eigenvalue 1 of P will be approximately piecewise constant.

where $\alpha^{(l)} \in \mathbb{R}^N, l = 1, \ldots, n_e$ are Lagrange multiplier vectors. The Karush-Kuhn-Tucker (KKT) optimality conditions are given by:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w^{(l)}} = 0 \rightarrow w^{(l)} = \Phi^T \alpha^{(l)} \\ \frac{\partial \mathcal{L}}{\partial e^{(l)}} = 0 \rightarrow \alpha^{(l)} = \frac{\gamma_l}{N} V e^{(l)} \\ \frac{\partial \mathcal{L}}{\partial b_l} = 0 \rightarrow 1_N^T \alpha^{(l)} = 0 \\ \frac{\partial \mathcal{L}}{\partial \alpha^{(l)}} = 0 \rightarrow e^{(l)} = \Phi w^{(l)} + b_l 1_N \end{cases} \tag{4}$$

After eliminating the primal variables $w^{(l)}, e^{(l)}, b_l$ and applying Mercer's theorem (also called the kernel trick) the following dual eigenvalue problem can be formulated:

$$V M_V \Omega \alpha^{(l)} = \lambda_l \alpha^{(l)}, l = 1, \ldots, n_e \tag{5}$$

where $\lambda_l = N/\gamma_l$, $M_V = I_N - 1/(1_N^T V 1_N) 1_N 1_N^T V$ is a weighted centering matrix, $\Omega = \Phi\Phi^T$ is the kernel matrix with ij-th entry $\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$ and $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a positive definite kernel. As mentioned in [8], each eigenvalue/eigenvector of (5) leads to a value zero in the primal objective function in (3). Thus, all pairs constitute a pool of candidates. The solutions to (5) are then the n_e eigenvectors that maximize the term $\frac{1}{2N} \sum_{l=1}^{n_e} \gamma_l e^{(l)^T} V e^{(l)} = \frac{1}{2} \sum_{l=1}^{n_e} \lambda_l \alpha^{(l)^T} V^{-1} \alpha^{(l)}$ in the objective function (3).

3.2. Choice of the Weighting Matrix

When V is set to the identity matrix I_N, then M_V becomes the centering matrix and (5) corresponds to kernel PCA. On the other hand, if $V = D^{-1}$ then (5) is related to the random walks Laplacian used in spectral clustering methods [15]. The relationship can be interpreted as first removing a weighted mean from the columns of the kernel matrix (which acts here as an affinity matrix) and then applying the random walks algorithm. The resulting eigenvalue problem is

$$D^{-1} M_D \Omega \alpha^{(l)} = \lambda_l \alpha^{(l)}, l = 1, \ldots, n_e, \tag{6}$$

with $M_D = I_N - 1_N 1_N^T D^{-1} / (1_N^T D^{-1} 1_N)$. The matrix $D^{-1} M_D \Omega$ is positive semi-definite with eigenvalues $\lambda_i, i = 1, \ldots, N$ such that $1 \geq \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N \geq 0$. The clustering model (2) can now be expressed in terms of the dual variables as:

$$e_i^{(l)} = w^{(l)^T} \varphi(x_i) + b_l = \sum_{j=1}^{N} \alpha_j^{(l)} K(x_i, x_j) + b_l \tag{7}$$

and the bias terms $b_l = -1/(1_N^T D^{-1} 1_N) 1_N^T D^{-1} \Omega \alpha^{(l)}, l = 1, \ldots, n_e$.

3.3. Choice of n_e

In the case of Ω representing the affinity matrix of a graph consisting of k subgraphs with no connections between them, the eigenvectors $\alpha^{(l)}$ corresponding to the largest $k-1$ eigenvalues of (6) are piecewise constant on the underlying partitioning of the graph. Moreover, these particular eigenvectors have eigenvalue equal to 1. This means that all clustering information is contained in the subspace spanned by the $k-1$ dominant eigenvectors of $D^{-1} M_D \Omega$. Thus, the problem of selecting the number of additional projected variables n_e is solved by setting $n_e = k - 1$. An important difference with popular spectral clustering algorithms such as the normalized cut [2] and the random walks model [15] relies on the fact that the constant vector is not an eigenvector of (6). This can be seen from the KKT condition $1_N^T \alpha^{(l)} = 0, l = 1, \ldots, k-1$ imposed by the bias terms which results in having eigenvectors with zero mean. Thus, the eigenvectors are piecewise constant and have zero mean which corresponds to representing each cluster as a single point in a different orthant in the $(k-1)-$ dimensional subspace spanned by the eigenvectors with eigenvalue 1.

3.4. Encoding the Eigenvectors

In classical spectral clustering, k-means is performed over the eigenvectors to obtain the final clustering. As discussed before, this approach works since in ideal cases the clusters have a very localized representation in the eigenvectors. However in our approach, we do not work on the eigenvectors but on the projections $e^{(l)}$ which do not have a localized structure. Thus, the encoding stage aims at combining the set of $k-1$ binary cluster indicators given by $\text{sign}(e^{(l)}), l = 1, \ldots, k-1$ into the final k groups as it is common in multiclass formulations (e.g. LS-SVM classifiers [16]). Since $\alpha^{(l)}$ and $e^{(l)}$ have the same sign (from the KKT conditions), it is possible to obtain a codebook from either the binarized projections or the eigenvectors. In ideal situations, when there is no connection between the k subgraphs, there will be exactly k orthants in which the clusters lie. However in real applications, there might be weak connections between the underlying groups leading to more than k codewords. In this case, the k codewords with most occurrences form the codebook and the remaining spurious codewords are assigned to those of the codebook with minimal Hamming distance.

3.5. Making Predictions - Decoding

One of the main advantages of the proposed formulation is the possibility to evaluate the model on out-of-sample points. This is important for predictive capabilities, model selection and even large-scale data analysis. Given an arbitrary input data point x, the projections are calculated as:

$$\hat{e}^{(l)}(x) = w^{(l)^T} \varphi(x) + b_l = \sum_{j=1}^{N} \alpha_j^{(l)} K(x_j, x) + b_l, l = 1, \ldots, k-1 \tag{8}$$

which correspond to the latent variables of the binary clustering indicators given by $\text{sign}(\hat{e}^{(l)}(x))$. The decoding step consists of assigning $\text{sign}(\hat{e}^{(l)}(x))$ to the cluster codeword in the codebook with minimal Hamming distance. In classical spectral clustering, the extensions to unseen data points are not clearly defined and should rely on approximations such as the Nyström method [17, 7]. The projected variables (8) provide a natural way of extending the model to out-of-sample points.

3.6. Model Selection

Model selection is a critical issue in clustering due mainly to the lack of a clear model. In classical spectral clustering, the grouping is obtained from the eigenvectors of a Laplacian matrix containing pairwise similarity information. One of the most popular similarity functions used in spectral clustering is the RBF kernel defined as $K(x_i, x_j) = \exp(-\|x_i - x_j\|_2^2/\sigma^2)$. The clustering results depend greatly on the choice of the bandwidth σ and typically this parameter is considered known which is unrealistic for real-life problems. However, since in our setting we have a model and a mechanism to extend the clustering to out-of-sample points, we can perform model selection in a learning framework with training, validation and test stages. A model selection criterion called the Balanced Line Fit (BLF) was introduced in [8] and is summarized as follows. Consider the assumptions: (i) the training data $\mathcal{D}$ has been sampled i.i.d. from an unknown distribution $\mathcal{P}$, (ii) $\mathcal{P}$ contains k clusters denoted by $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$, (iii) the eigenvectors $\alpha^{(l)}, l = 1, \ldots, k-1$ are piecewise constant, (iv) $x \in \mathcal{A}_p$ is an unseen data point sampled i.i.d. from $\mathcal{P}$ and (v) $K(x, z) \to 0$ when x and z are in different clusters. In this case, the projections (8) of x can be rewritten as:

$$\hat{e}^{(l)}(x) = c_p^{(l)} \sum_{j \in \mathcal{A}_p} K(x, x_j) + \sum_{u \notin \mathcal{A}_p} \alpha_u^{(l)} K(x, x_u) + b_l, \qquad (9)$$

$l = 1, \ldots, k-1$. Due to assumption (v), the term $\sum_{u \notin \mathcal{A}_p} \alpha_u^{(l)} K(x, x_u)$ vanishes because x and x_u are in different clusters. Thus, (9) becomes

$$\hat{e}^{(l)}(x) = c_p^{(l)} \sum_{j \in \mathcal{A}_p} K(x, x_j) + b_l, l = 1, \ldots, k-1 \qquad (10)$$

which means that the projection of x is collinear with the projections of the points in the same cluster $\mathcal{A}_p$. This structural property allows us to look for clustering parameters (i.e. kernel parameters and number of clusters k) using validation data such that the projections are as collinear as possible. Consider the following measure of collinearity of a validation dataset $\mathcal{D}^v$, with respect to a partitioning $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ and $k > 2$:

$$\text{linefit}(\mathcal{D}^v, k) = \frac{1}{k} \sum_{p=1}^{k} \frac{k-1}{k-2} \Big(\frac{\zeta_1^{(p)}}{\sum_{l=1}^{k-1} \zeta_l^{(p)}} - \frac{1}{k-1} \Big), \qquad (11)$$

where $\zeta_1^{(p)} \geq \ldots \geq \zeta_{k-1}^{(p)}$ are the ordered eigenvalues of the covariance matrix $C_Z^{(p)} = (1/|\mathcal{A}_p|) Z^{(p)^T} Z^{(p)}$, $p = 1, \ldots, k$, $Z^{(p)} \in \mathbb{R}^{|\mathcal{A}_p| \times (k-1)}$ is the matrix containing the projections (with the mean removed) of validation data assigned to the p-th cluster on the rows. The term $\zeta_1^{(p)} / \sum_l \zeta_l^{(p)}$ equals 1 when the rows of $Z^{(p)}$ are collinear. The additional terms in (11) scale the linefit between 0 and 1. The balance of the obtained clustering is also important and is measured with the balance index: $\text{balance}(\mathcal{D}^v, k) = \min\{|\mathcal{A}_p|\} / \max\{|\mathcal{A}_p|\}$, $p = 1, \ldots, k$. The BLF model selection criterion is then defined as [8]:

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

where $0 \leq \eta \leq 1$ is a user-defined parameter determining the weight given to the linefit with respect to the balance index.

In general, isotropic kernels (kernels that depend only on $\|x - z\|$) such as the RBF kernel and the χ^2 kernel fulfill condition (v) making them suitable for the BLF because when $\|x - z\|$ is large, $K(x, z)$ tends to zero.

4. Sparse Clustering Models

Classical spectral clustering methods need to store a $N \times N$ matrix and to compute an eigendecomposition. This can be problematic for large-scale data (large N). Approximations such as the Nyström method can be used to solve a reduced eigenvalue problem and to approximate the full-size eigenvectors solution. These techniques typically use random subsampling which may lead to choosing a subset of data points that do not represent the underlying structure of the data[2]. In the case of kernel spectral clustering, we can use the out-of-sample extension to train the clustering model on a subset of the data and to predict the cluster indicators of the remaining data points. An issue with kernel spectral clustering concerns the fact that the clustering dual model is expressed in terms of non-sparse kernel expansions where every data point in the training contributes. The non-sparsity of the clustering model is due to the KKT condition $w^{(l)} = \sum_{j=1}^{N} \alpha_j^{(l)} \varphi(x_j) = \Phi^T \alpha^{(l)}, l = 1, \ldots, k-1$ which gives an optimal representation of the primal vectors $w^{(l)}$ as linear combinations of the mapped training data points in the feature space. Since the coefficients $\alpha^{(l)}$ in the expansion are eigenvectors, the sum goes through all training set data points due to the fact that the $\alpha_i^{(l)}$ are typically non-zero. The objective is then to find a reduced set of training points $\mathcal{R} = \{\tilde{x}_m\}_{m=1}^{R} \subset \mathcal{D}$ and reduced set coefficients $\beta^{(l)} = [\beta_1^{(l)}; \ldots; \beta_R^{(l)}]$ to approximate $w^{(l)}$ by:

$$\tilde{w}^{(l)} = \sum_{m=1}^{R} \beta_m^{(l)} \varphi(\tilde{x}_m) = \Psi^T \beta^{(l)}, \qquad (12)$$

where $\Psi = [\varphi(\tilde{x}_1)^T; \ldots; \varphi(\tilde{x}_R)^T] \in \mathbb{R}^{R \times d_h}, l = 1, \ldots, k-1$ and $R < N$. The approximated clustering model for an arbitrary data point x can be computed as:

$$\hat{e}^{(l)}(x) \approx \sum_{m=1}^{R} \beta_m^{(l)} K(x, \tilde{x}_m) + \hat{b}_l, l = 1, \ldots, k-1. \qquad (13)$$

where $\hat{b}_l$ are the re-estimated bias terms.

Thus, we will incorporate sparsity using reduced set methods by working in three stages: first, the size of the eigenvalue problem will be reduced from the total size of the dataset N_{full} to N, where $N \ll N_{\text{full}}$ by selecting a representative subset of training points. The second stage consists of finding the reduced set $\mathcal{R}$. The last stage aims at obtaining reduced set coefficients $\beta^{(l)}$ in order to compute the approximated clustering model (13).

4.1. Stage 1 - Selecting representative points

4.1.1. Quadratic Renyi entropy

A nonparametric estimation of the quadratic Renyi entropy using RBF kernels has been discussed in [19]. Given a full data

[2] An improved Nyström approximation method has been reported in [18] which uses k-means to select landmark points instead of random selection.

set $\mathcal{D}_{\text{full}} = \{x_h\}_{h=1}^{N_{\text{full}}}$, this scheme finds a representative subset of N data points $\mathcal{D} \subset \mathcal{D}_{\text{full}}$ such that the quadratic Renyi entropy

$$H_R(\mathcal{D}, \sigma) = -\log \int p(x)^2 dx \approx -\log\left(\frac{1}{N^2} 1_N^T \Omega 1_N\right) \quad (14)$$

is maximized. Here Ω is the $N \times N$ RBF kernel matrix with parameter σ. This criterion can be maximized iteratively in a greedy manner in order to select points that preserve the underlying structure of the data [16]. The pseudocode is shown in Algorithm 1.

Algorithm 1 Greedy subset selection using the quadratic Renyi entropy

Input: Full data set $\mathcal{D}_{\text{full}} = \{x_h\}_{h=1}^{N_{\text{full}}}$, RBF kernel parameter $\sigma \in \mathbb{R}$, subset size N.
Output: Subset that maximizes the quadratic Renyi entropy $\mathcal{D} = \{x_i\}_{i=1}^{N}$

1: Randomly select a subset $\mathcal{D} = \{x_i\}_{i=1}^{N}$ from the full data set $\mathcal{D}_{\text{full}} = \{x_h\}_{h=1}^{N_{\text{full}}}, N \ll N_{\text{full}}$.
2: Compute the quadratic Renyi entropy of $\mathcal{D}$ using (14).
3: Select a data point $x^\star$ from $\mathcal{D}$ and select a data point $x^{\star\star}$ from the remaining pool of data $\mathcal{D}_{\text{full}} \setminus \mathcal{D}$.
4: Compute the quadratic Renyi entropy of $\tilde{\mathcal{D}} = \mathcal{D} \setminus \{x^\star\} \cup \{x^{\star\star}\}$
5: If the entropy of $\tilde{\mathcal{D}}$ increases with respect to the entropy of $\mathcal{D}$, then put $x^\star$ in $\mathcal{D}_{\text{full}} \setminus \mathcal{D}$ and $x^{\star\star}$ in $\mathcal{D}$. If there is no increase in entropy, then $x^\star$ and $x^{\star\star}$ are not swapped.
6: Iterate from step 3 until convergence.

4.1.2. Incomplete Cholesky decomposition

The incomplete Cholesky decomposition (ICD) [20] aims at computing a reduced-rank approximation of a symmetric positive definite matrix $A \in \mathbb{R}^{N_{\text{full}} \times N_{\text{full}}}$ as $A \approx GG^T$ where $G \in \mathbb{R}^{N_{\text{full}} \times N}$ is the Cholesky factor with $\|A - GG^T\|_2^2 \leq \theta$ and $N \leq N_{\text{full}}$. The user-defined parameter θ is an error threshold that controls the rank of the approximation. Large values of θ lead to large errors in the approximation but a small N. The ICD has been used to decrease the storage requirements and computational complexity of interior point methods in SVMs [21], to approximate contrast functions for ICA [22, 23, 24] and to compute approximated eigenvectors in spectral clustering [25]. This decomposition also gives information about the subset of N points (also called pivots) that define the reduced rank approximation. The pivots can be used as a representative subset of training data points and the approximation of (6) becomes [25]:

$$U^T M U \Lambda^2 \zeta^{(l)} = \lambda_l \zeta^{(l)}$$

where $\Omega \approx GG^T$ is the ICD of Ω, $G = U\Lambda V^T$ is the singular value decomposition of the Cholesky factor G, $\zeta^{(l)} = U^T \alpha^{(l)}$ and $l = 1, \ldots, k-1$. Since $U \in \mathbb{R}^{N_{\text{full}} \times N}$, the size of the eigenvalue problem is reduced to $N \times N$ instead of $N_{\text{full}} \times N_{\text{full}}$. It should be noted that this reduction in size applies only at the training level (eigenvalue problem). The clustering model still depends on the full dataset and several ways to sparsify the projections will be discussed in the sequel.

4.2. Stage 2 - Finding a reduced subset

4.2.1. Highly sparse representations

A highly sparse method for kernel spectral clustering was introduced in [9]. This method exploits the collinearity property of the projections when the clusters are well-separated. Since the cluster membership depends on the orthant in which the projections lie, the farthest point from the origin in each line can serve as a cluster prototype. These points are the most certain to belong to the given cluster. The method proposed in [9] use both tips of the lines (farthest and closest points relative to the origin) and the median point leading to highly sparse models that depend only on $3k$ data points where k is the number of clusters. This method has been proven successful in applications such as image segmentation achieving a large degree of sparsity. However, in cases of extremely nonlinear datasets, the approximation of the clustering model to depend only on $3k$ data points has to be treated with caution. This issue is due to the fact that if an arbitrary point x is not in the local neighborhood of any of the $3k$ data points that define the model expansion then the projections of x tend to be very small making the cluster membership decision unreliable. This problem has been observed for nonlinear toy datasets where the tuned RBF kernel parameter has a very small value which restricts the zone of influence of every data point to a small neighborhood.

4.2.2. $L_2 + L_1$ penalization

A method to find a reduced subset is through L_1 penalization [26, 27]. Given $\mathcal{D}$ and $\alpha^{(l)}$, the following optimization can be formulated:

$$\min_{\beta^{(l)}} \|\Phi^T \alpha^{(l)} - \Phi^T \beta^{(l)}\|_2^2 + \rho \|\beta^{(l)}\|_1, l = 1, \ldots, k-1 \quad (15)$$

where $\rho \in \mathbb{R}^+$ is a user-defined parameter which determines the tradeoff between sparseness and approximation quality. Since (15) consists of $k-1$ decoupled minimization problems, the resulting $\beta^{(l)}$ will introduce sparsity *per component*. This might not be desirable if the indices of the non-zero coefficients differ too much across components. In order to tackle this issue, the following modification to (15) was proposed in [27]:

$$\min_{\beta^{(l)}} \sum_{l=1}^{k-1} \left\| \sum_{i=1}^{N} \alpha_i^{(l)} \varphi(x_i) - \sum_{i=1}^{N} \beta_i^{(l)} \varphi(x_i) \right\|_2^2 + \rho \sum_{i=1}^{N} \xi_i \quad (16)$$
$$\text{s.t.} \begin{cases} \left|\beta_i^{(l)}\right| \leq \xi_i, i = 1, \ldots, N, l = 1, \ldots, k-1 \\ \xi_i \geq 0, \end{cases}$$

which aims at obtaining sparse reduced set coefficients with the same set of support vectors per component. The indices of the non-zero coefficients of $\beta^{(l)}$ are used to build a reduced set[3].

[3] Both the reduced set and the reduced set coefficients can be obtained by solving (16). However, it has been reported in [27] that using only the reduced set and recomputing the coefficients with other methods leads to better approximations.

4.3. *Stage* 3 - *Finding the reduced set coefficients*

Having found the reduced set $\mathcal{R}$ by any of the methods described previously, the last stage consists of finding the reduced set coefficients $\beta^{(l)}$ in the expansion $\tilde{w} = \Psi^T \beta^{(l)}$. This can done by minimizing the squared two-norm of the residuals: $\min_{\beta^{(l)}} \|w^{(l)} - \tilde{w}^{(l)}\|_2^2$. This methodology has been described in [27] for approximating similar expansions in kernel PCA problems and in [25] for sparse spectral clustering combined with the incomplete Cholesky factorization. Using $w^{(l)} = \Phi^T \alpha^{(l)}$, $\tilde{w}^{(l)} = \Psi^T \beta^{(l)}$ and applying the kernel trick leads to:

$$\min_{\beta^{(l)}} \alpha^{(l)^T} \Omega \alpha^{(l)} - 2\beta^{(l)^T} \Omega^{\Psi\Phi} \alpha^{(l)} + \beta^{(l)^T} \Omega^{\Psi\Psi} \beta^{(l)}, l = 1, \ldots, k-1, \tag{17}$$

where $\Omega^{\Psi\Phi} \in \mathbb{R}^{R \times N}$, with $\Omega^{\Psi\Phi}_{mj} = K(\tilde{x}_m, x_i), m = 1, \ldots, R, j = 1, \ldots, N, \Omega^{\Psi\Psi} \in \mathbb{R}^{R \times R}$, with $\Omega^{\Psi\Psi}_{mn} = K(\tilde{x}_m, \tilde{x}_n), n = 1, \ldots, R$. The first-order optimality condition $\partial \|w^{(l)} - \tilde{w}^{(l)}\|_2^2 / \partial \beta^{(l)} = 0$ gives:

$$\Omega^{\Psi\Psi} \beta^{(l)} = \Omega^{\Psi\Phi} \alpha^{(l)}, l = 1, \ldots, k-1, \tag{18}$$

which is a linear system to be solved in $\beta^{(l)}$. The general algorithm for sparse kernel spectral clustering models is shown in Algorithm 2.

Algorithm 2 Sparse kernel spectral clustering model

Input: Full data set $\mathcal{D}_{\text{full}} = \{x_h\}_{h=1}^{N_{\text{full}}}$, subset size N, positive definite kernel function $K(x_i, x_j)$, number of clusters k.
Output: $\Delta = \{\mathcal{A}_1 \ldots \mathcal{A}_k\}$.

1: Obtain a representative subset of size N for training $\mathcal{D} = \{x_i\}_{i=1}^N$ (e.g. using the quadratic Renyi entropy criterion or the incomplete Cholesky decomposition of the kernel matrix).
2: Compute the eigenvectors $\alpha^{(l)}, l = 1, \ldots, k-1$ corresponding to the largest $k-1$ eigenvalues of $D^{-1} M_D \Omega$ and Ω is the training kernel matrix $\Omega_{ij} = K(x_i, x_j)$.
3: Find a reduced set of training data points $\mathcal{R} = \{\tilde{x}_m\}_{m=1}^R \subset \mathcal{D}$ (e.g. using the highly sparse method or the $L_2 + L_1$ penalization method (16)).
4: Using $\mathcal{R}$, compute the reduced set coefficients $\beta^{(l)}$ by means of (18).
5: Compute the approximated training score variables: $\hat{e}_i^{(l)} \approx \sum_{m=1}^R \beta_m^{(l)} K(x_i, \tilde{x}_m) + \hat{b}_l, l = 1, \ldots, k-1$.
6: Binarize the training score variables: $\text{sign}(\hat{e}_i^{(l)}), i = 1, \ldots, N, l = 1, \ldots, k-1$, and let $\text{sign}(\hat{e}_i) \in \{-1, 1\}^{k-1}$ be the encoding vector for the training data point x_i, $i = 1, \ldots, N$.
7: Count the occurrences of the different encodings and find the k encodings with most occurrences. Let the codeset be formed by these k encodings: $C = \{c_p\}_{p=1}^k, c_p \in \{-1, 1\}^{k-1}$
8: $\forall i$, assign x_i to $\mathcal{A}_{p^\star}$ where $p^\star = \text{argmin}_p d_H(\text{sign}(\alpha_i), c_p)$ and $d_H(\cdot, \cdot)$ is the Hamming distance.
9: For an out-of-sample point x, compute the approximated projections $\hat{e}_i^{(l)}(x)$ using (13).
10: Binarize the projections $\hat{e}_i^{(l)}(x)$ and let $\text{sign}(\hat{e}_i^{(l)}(x)) \in \{-1, 1\}^{k-1}$ be the encoding vector of x.
11: Assign x to $\mathcal{A}_{p^\star}$ where $p^\star = \text{argmin}_p d_H(\text{sign}(\hat{e}_i^{(l)}(x)), c_p)$.

5. Experimental Results

In this Section, some experimental results are presented. All reported experiments were carried out in MATLAB 7.10 on an Intel Core 2 Duo, 3.06 GHz, 4 GB RAM, running Mac OSX 10.6. All datasets were split into training, validation and test sets to perform model selection using the BLF criterion. This criterion was used to find the number of clusters k and the kernel parameters. The η parameter of the BlF was fixed to $\eta = 0.75$, hence giving more emphasis to the linefit with respect to the balance. The adjusted Rand index (ARI) [28] is used as a criterion to assess the performance with respect to known external cluster indicators. This clustering performance index has a maximal value of 1 indicating perfect match between the obtained cluster indicators and the external reference. All performances are given for the whole datasets, including the training, validation and test sets. The `CVX` toolbox [29, 30] is used to solve the $L_2 + L_1$ penalization method (16).

5.1. *Toy problems*

The first toy problem consists of clustering Gaussian clouds in $2D$. The number of Gaussian clouds is increased iteratively to assess the performance of several methods with respect to an increasing number of clusters. Figure 1 shows performance and computation times of several methods for selecting a subset of points for training. A random selection procedure, the incomplete Cholesky decomposition (ICD) and the quadratic Renyi entropy are used to select the training set $\mathcal{D}$ for kernel spectral clustering. The size of the training set is $N = N_{\text{full}}/5$. A validation set is also selected to obtain the RBF kernel parameter σ via the BLF. Performance results with the Nyström method for spectral clustering [7] are also reported. In terms of ARI, the quadratic Renyi entropy and the ICD give the best results. However, the ICD takes a considerable amount of time compared to the other methods.

The second toy problem consists of detecting three concentric rings in $2D$. This type of dataset is challenging for spectral clustering due mainly to its nonlinearity. The full dataset consists of $1,200$ data points. Figure 2 shows the selection of a representative subset of training data using the incomplete Cholesky decomposition and the entropy criterion. The latter needs less data points to obtain a perfect separation of the rings (ARI = 1) and requires less computation time. Figure 3 shows the influence of the L_1 penalization parameter ρ on performance and sparsity. The training set ($N = 342$) is obtained through the entropy criterion and (16) is solved using `CVX`. The average computation time of solving (16) for a given ρ is 21 seconds. Figure 3 shows the support vectors of the highly sparse method. Note that $R = 3k = 9$ support vectors are not enough to solve this highly nonlinear clustering problem with an ARI of 1. In this particular case, $R = 13k = 39$ is the minimum number of support vectors required.

5.2. *Image segmentation*

We used a color image from the Berkeley image data set [31]. The image is 321×481 pixels for a total number of $154,401$ pixels. In these experiments, we computed a local color histogram with a 5×5 pixels window around each pixel using minimum variance color quantization of 8 levels. We used the χ^2 test to compute the distance between two local color histograms $h^{(i)}$ and $h^{(j)}$ [32] $\chi^2_{ij} = 0.5 \sum_{b=1}^B (h_b^{(i)} - h_b^{(j)})^2 / (h_b^{(i)} + h_b^{(j)})$, where B is total number of quantization levels. The histograms are assumed to be normalized $\sum_{b=1}^B h_b^{(i)} = 1, i = 1, \ldots, N$. The χ^2

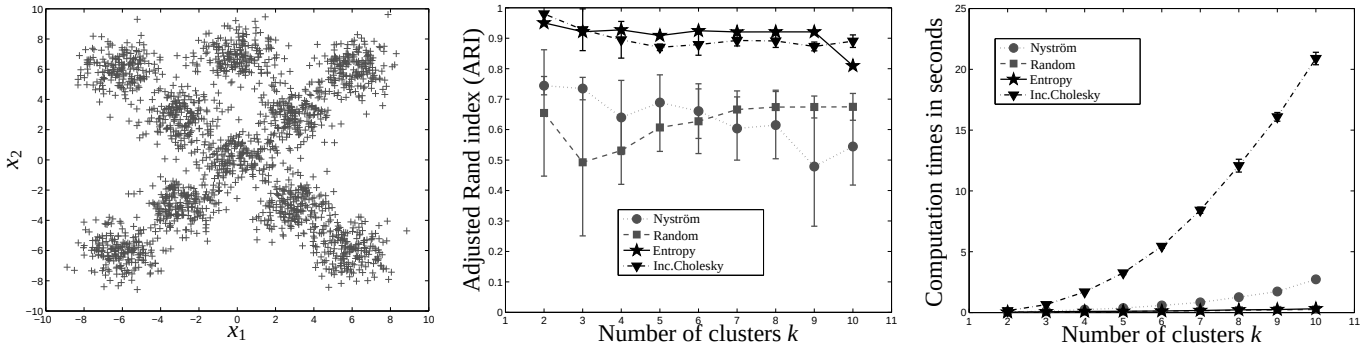

Figure 1: Toy example 1 - Performance and computation times for selecting a representative subset of training points. **Left:** Complete dataset of 10 Gaussian clouds, $N_{\text{full}} = 2,000$. **Center:** Performance in terms of ARI. **Right:** Computation times. The entropy criterion and the incomplete Cholesky decomposition (ICD) outperform the other algorithms in terms of ARI. However, the computation time of the ICD depends greatly on an increasing number of clusters.

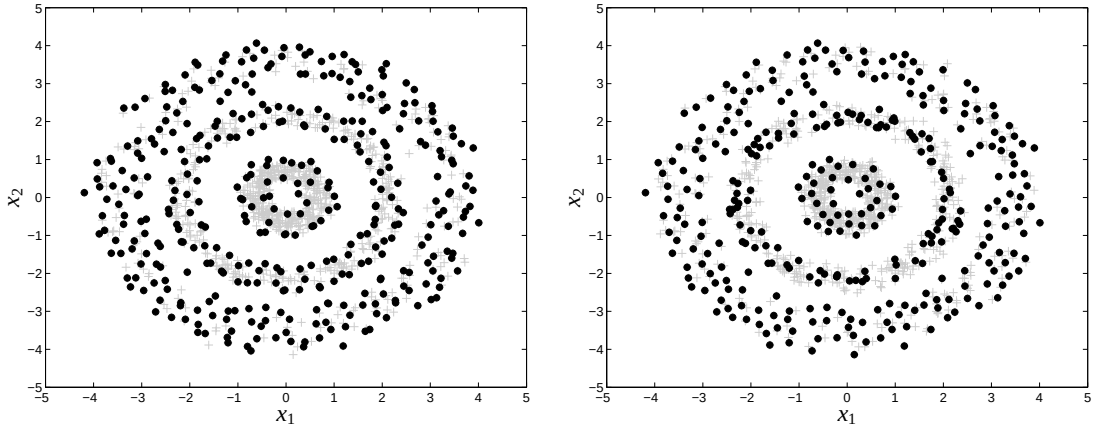

Figure 2: Toy example 2 - Representative subset of training points. The full dataset $N_{\text{full}} = 2,000$ is depicted in grey. Selected subset in black. The training set size $N < N_{\text{full}}$ is determined as the minimum number of data points needed to have an ARI of 1 on a validation set (perfect separation of the rings). Tuned RBF kernel parameter $\sigma = 0.1$. **Left:** Incomplete Cholesky decomposition, $N = 378$, computation time: 4.6 seconds. **Right:** Quadratic Renyi entropy, $N = 342$, computation time: 2.4 seconds. The entropy criterion requires less data points and runs faster.

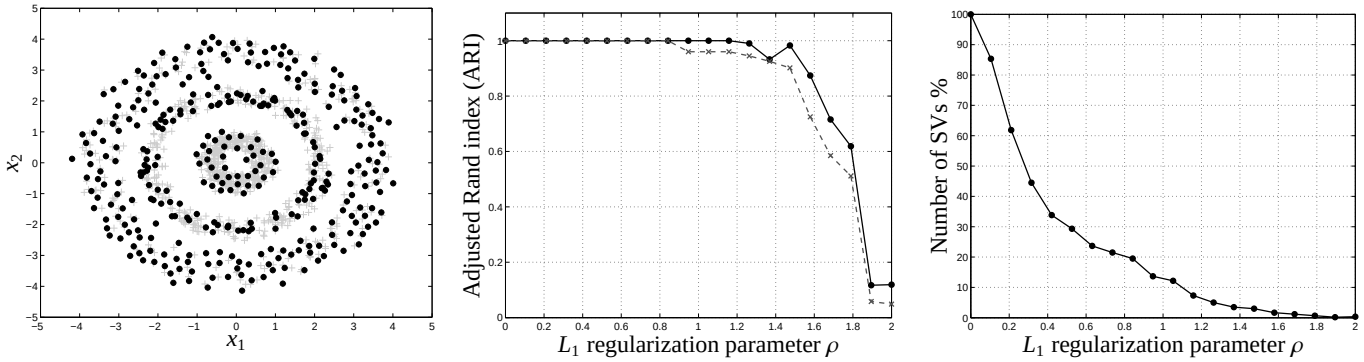

Figure 3: Toy example 2 - $L_2 + L_1$ penalization. **Left:** Full dataset (grey) and training data points (black) $N = 342$, obtained through the entropy criterion. **Center:** ARI vs. ρ. The dashed line corresponds to solving (16). The solid line corresponds to obtaining support vectors from (16) and recomputing the reduced set coefficients by (18). **Right:** Sparseness vs. ρ. Note that only 10% ($R = 34$) of the training set is required to perfectly detect the three rings. Computation time is 21 seconds using `CVX`.

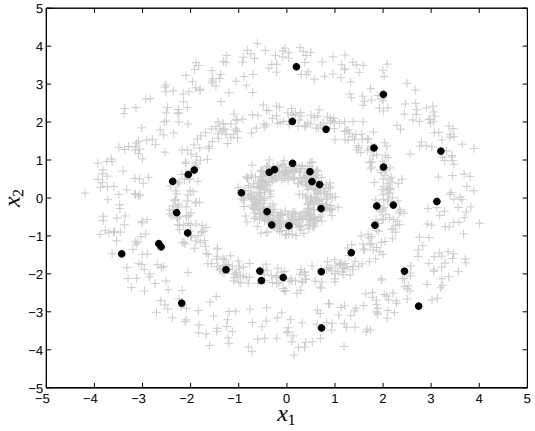

Figure 4: Toy example 2 - Highly sparse representations. For this nonlinear toy dataset, using only $R = 3k = 9$ support vectors does not lead to a perfect separation of the rings. The training set $N = 600$ is obtained through the entropy criterion. The tuned RBF kernel parameter is $\sigma = 0.1$. In order to get an ARI of 1 of validation data, $R = 13k = 39$ support vectors are needed. These support vectors correspond to uniformly sampled points on the lines.

kernel $K(h^{(i)}, h^{(j)}) = \exp(-\chi^2_{ij}/\sigma_\chi)$ with parameter $\sigma_\chi \in \mathbb{R}^+$ is positive definite and has shown to be a robust and efficient way of comparing the similarity between two histograms for color discrimination and image segmentation [7]. Figure 5 shows the training set obtained via incomplete Cholesky decomposition and the entropy criterion. The training set size is fixed to $N = 600$. Both methods give similar results for selecting a representative subset, but the incomplete Cholesky decomposition is more costly in terms of computation times. Figure 6 shows the results for the highly sparse method. The clustering model is approximated such that it depends only on $R = 3k = 12$ data points. These points correspond to both tips of each line and the median point. The agreement between the approximated cluster membership and the indicators of the model without approximation is ARI = 0.84 indicating a good agreement with a highly sparse representation. The $L_2 + L_1$ penalization method gives similar results in terms of ARI compared to the highly sparse approach. However, the latter approach requires much less reduced set points. Figure 7 shows the results of applying a Canny edge detector [33] over the image segments. This filter is a standard algorithm in image processing to find edges in images. The results are visually appealing with a higher degree of sparsity.

6. Conclusions

Several methods for obtaining sparse and highly sparse model representations for kernel spectral clustering are presented. The formulation is based on weighted kernel PCA for a specific choice of weights. The methodology is cast in a constrained optimization setting with primal and dual model representations. This allows the extension of the clustering model to out-of-sample points which can be used for predictive purposes in a learning framework. The methodology is aimed to tackle large-scale datasets in an efficient way. The highly sparse formulation exploits the structure of the projected variables when the clusters are well-formed. Comparisons with other methods are given and experimental results with difficult toy problems and image segmentation problems are reported.

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

[2] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Machine Intell. 22 (8) (2000) 888–905.

[3] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: T. G. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems 14, MIT Press, Cambridge, MA, 2002, pp. 849–856.

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

[5] C. Ding, X. He, Linearized cluster assignment via spectral ordering, in: ICML '04: Proceedings of the twenty-first International Conference on Machine Learning, ACM Press, New York, NY, USA, 2004, p. 30.

[6] F. R. Bach, M. I. Jordan, Learning spectral clustering, with application to speech separation, Journal of Machine Learning Research 7 (2006) 1963–2001.

[7] 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.

[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) (2010) 335–347.

[9] C. Alzate, J. A. K. Suykens, Highly sparse kernel spectral clustering with predictive out-of-sample extensions, in: Proc. of the 18th European Symposium on Artificial Neural Networks (ESANN 2010), 2010, pp. 235–240.

[10] M. Gu, H. Zha, C. Ding, X. He, H. Simon, Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering, Technical report, Penn. State Univ, Computer Science and Engineering. (2001).

[11] K. Fan, On a theorem of Weyl concerning eigenvalues of linear transformations., Proc. Natl. Acad. Sci. USA 35 (11) (1949) 652–655.

[12] P. K. Chan, M. D. F. Schlag, J. Y. Zien, Spectral k-way ratio-cut partitioning and clustering, IEEE Trans. Computer-Aided Design 13 (9) (1994) 1088–1096.

[13] C. Ding, X. He, K-means clustering via principal component analysis, in: ICML '04: Proceedings of the 21st international conference on Machine learning, 2004, p. 29.



Figure 5: Image segmentation - Representative subsets for training. The number of training points is fixed to $N = 600$. **Left:** Original 321×481 image, $N_{\text{full}} = 154,401$. **Center:** Representative subset found by the quadratic Renyi entropy criterion. **Right:** Representative subset found by the incomplete Cholesky decomposition. Both methods give similar results.

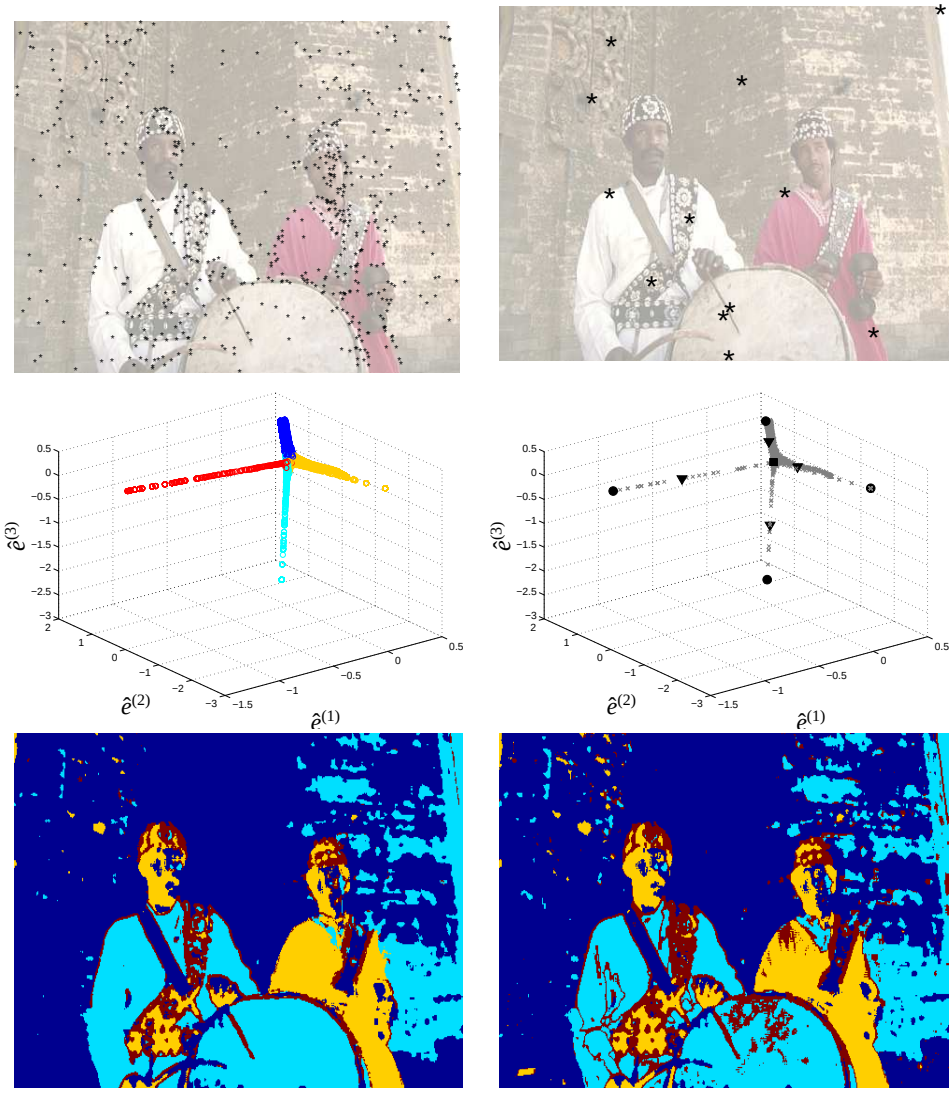

Figure 6: Image segmentation - Highly sparse method. **Top left:** Representative subset found by the entropy criterion, $N = 600$. **Top right:** Support vectors found by the highly sparse method. **Center left:** Projected variables of validation data used for model selection using the BLF. Note the strong line structure of each cluster. The obtained maximal BLF is 0.89 for a χ^2 kernel parameter of $\sigma_\chi = 0.035$. **Center right:** Projections and support vectors (both ends of the lines and median point). **Bottom:** Image segments without approximations and with the highly sparse method respectively.

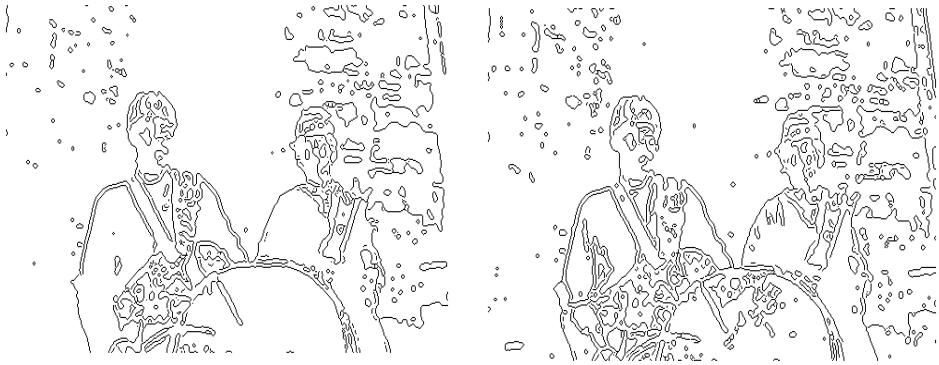

Figure 7: Image segmentation - Highly sparse method. Edges found by applying a Canny edge detector on the image segments. **Left:** Results without approximations. **Right:** Results with the highly sparse method.

[14] C. Alzate, J. A. K. Suykens, A weighted kernel PCA formulation with out-of-sample extensions for spectral clustering methods, in: Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN'06), 2006, pp. 138–144.

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

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

[17] C. K. I. Williams, M. Seeger, Using the Nyström method to speed up kernel machines, in: Advances in Neural Information Processing Systems 13, MIT Press, 2001.

[18] K. Zhang, I. Tsang, J. Kwok, Improved nystrom low-rank approximation and error analysis, in: ICML '08: Proceedings of the 25th Annual International Conference on Machine Learning, 2008, pp. 1232–1239.

[19] M. Girolami, Orthogonal series density estimation and the kernel eigenvalue problem, Neural Computation 14 (3) (2002) 669–688.

[20] G. H. Golub, C. F. Van Loan, Matrix Computations, The Johns Hopkins University Press, 1996.

[21] S. Fine, K. Scheinberg, Efficient SVM training using low-rank kernel representations, Journal of Machine Learning Research 2 (2001) 243–264.

[22] F. R. Bach, M. I. Jordan, Kernel independent component analysis, Journal of Machine Learning Research 3 (2002) 1–48.

[23] A. Gretton, R. Herbrich, A. J. Smola, O. Bousquet, B. Schölkopf, Kernel methods for measuring independence, Journal of Machine Learning Research 6 (2005) 2075–2129.

[24] C. Alzate, J. A. K. Suykens, A regularized kernel CCA contrast function for ICA, Neural Networks 21 (2-3) (2008) 170–181.

[25] 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.

[26] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[27] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K. R. Müller, M. Scholz, G. Rätsch, A. J. Smola, Input space versus feature space in kernel-based methods, IEEE Transactions on Neural Networks 10 (5) (1999) 1000–1017.

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

[29] M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 1.21, `http://cvxr.com/cvx` (May 2010).

[30] M. Grant, S. Boyd, Graph implementations for nonsmooth convex programs, in: V. Blondel, S. Boyd, H. Kimura (Eds.), Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 2008, pp. 95–110.

[31] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proc. 8th Int'l Conf. Computer Vision, Vol. 2, 2001, pp. 416–423.

[32] J. Puzicha, T. Hofmann, J. Buhmann, Non-parametric similarity measures for unsupervised texture segmentation and image retrieval, in: Computer Vision and Pattern Recognition, 1997, pp. 267–272.

[33] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6) (1986) 679–698.