A Weighted Kernel PCA Formulation with Out-of-Sample Extensions for Spectral Clustering Methods

Carlos Alzate, Johan A. K. Suykens

***Abstract*— A new formulation to spectral clustering methods based on the weighted kernel principal component analysis is presented. This formulation fits in the Least Squares Support Vector Machines (LS-SVM) framework as a primal-dual interpretation in the context of constrained optimization problems. Starting from the LS-SVM formulation to kernel PCA, a weighted approach is derived. An advantage of this method is the possibility to apply the trained clustering model to out-of-sample (test) data points without using approximation techniques such as the Nyström method. Links with some existing spectral clustering techniques are given, showing that these techniques are particular cases of weighted kernel PCA. Simulation results with toy and real-life data show improvements in terms of generalization to new samples.**

I. Introduction

Spectral clustering techniques have emerged as promising unsupervised learning methods to group data points that are similar. These methods have been successfully applied to machine learning, data analysis, image processing, pattern recognition and VLSI design.

Although several algorithms have been developed, they all use the eigenvectors of a similarity matrix derived from the data, see e.g. [12], [14], [3]. Starting from graph theory, spectral clustering methods are formulated as optimization problems in which the objective is to minimize the cost of a bipartitioning (also called cut). In general, these optimization problems are NP-hard [5] due to a discrete constraint on the clustering indicator vector q. If this constraint is relaxed by letting q take real values while controlling its norm, then an approximate solution can be obtained in an efficient way. This approximate solution is given by an eigenvalue problem of the affinity matrix.

Kernel Principal Component Analysis (Kernel PCA) [13] is an unsupervised learning technique used for nonlinear feature extraction, dimensionality reduction and denoising. The objective is to find projected variables with maximal variance in a kernel induced feature space. The solutions are given by the eigenvectors of the centered kernel matrix of a dataset and the corresponding eigenvalues indicate the amount of variance captured by the projection. A weighted formulation to kernel PCA based on the LS-SVM framework was introduced in [2]. Using this formulation, it is possible to achieve desirable properties such as robustness and sparseness.

In this paper, we consider the weighted formulation to kernel PCA. Within this formulation, links to existing spectral clustering methods can be made, with weighted kernel PCA as a unifying view. In this way, the extension to out-of-sample data points can be calculated in an exact way by means of the projections over the eigenvectors. A different approach to extend the formulation to new test points was introduced in [4]. This formulation uses the Nyström method in order to approximate the underlying eigenfunction. The proposed out-of-sample extensions do not rely on approximations of the feature map, but are based on the primal-dual formulation insights as demonstrated in [1], [2], [16], [15].

This paper is organized as follows: Section II contains a review of existing spectral clustering techniques. In Section III, we describe the weighted kernel PCA formulation and the links to spectral clustering, together with the extension to out-of-sample data points. In Section IV, we report the empirical results and in Section V we give conclusions.

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

II. Spectral Clustering

In this section, a review of some spectral clustering methods is presented. In graph theory, a set of data points $\mathcal{D} = \{x_i\}_{i=1}^N$ can be represented as an undirected graph. The nodes of the graph are the data points, and an edge is created between pairs of nodes. For each edge, there is an associated weight representing a function of the similarity between each pair of nodes.

Given an undirected graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of N nodes and $\mathcal{E}$ is the set of edges, the problem of graph bipartitioning consists of separating the graph into two sets $\mathcal{A}$, $\mathcal{B}$ by eliminating edges connecting the two sets. The sets should be disjoint such that: $\mathcal{A} \cup \mathcal{B} = \mathcal{V}$ and $\mathcal{A} \cap \mathcal{B} = \emptyset$. The total weight of the edges that have to be eliminated is called the *cut*:

$$\mathrm{cut}(\mathcal{A}, \mathcal{B}) = \sum_{a \in \mathcal{A}, b \in \mathcal{B}} w(a, b)$$

where $w(a, b)$ is the weight between node a and b.

A. The Mincut

The *mincut* criterion [8] is formulated as follows:

$$\min_q J_{mincut} = \mathrm{cut}(\mathcal{A}, \mathcal{B}) = \frac{1}{2} \sum_{i,j} w(i, j)(q_i - q_j)^2 \quad (1)$$

where q_i is a cluster membership indicator:

$$q_i = \begin{cases} 1, & \text{if } i \in \mathcal{A} \\ -1, & \text{if } i \in \mathcal{B}. \end{cases}$$

Expanding J_{mincut} leads to:

$$\begin{aligned} J_{mincut} &= \frac{1}{2}\sum_{i,j} w(i,j)(q_i - q_j)^2 \\ &= \frac{1}{2}\Big(\sum_i q_i^2 \sum_j w(i,j) + \sum_j q_j^2 \sum_i w(i,j) \\ &\quad -2\sum_{i,j} w(i,j) q_i q_j\Big) \\ &= \frac{1}{2}\Big(\sum_i q_i^2 d_i + \sum_j q_j^2 d_j - 2\sum_{i,j} w(i,j) q_i q_j\Big) \end{aligned}$$

where $d_i = \sum_j w(i,j)$.

Defining $D = \mathrm{diag}(d_1; \ldots; d_N)$ and W as the symmetric *affinity* matrix with ij-th entry equal to $w(i,j)$ leads to following primal problem:

$$\begin{aligned} \min_q J_{mincut} &= q^T(D-W)q \\ \text{such that} &\quad q \in \{-1,1\}^N \end{aligned} \tag{2}$$

which is NP-hard [5] due to the constraint on q. A sub-optimal solution can be found by relaxing the constraint and letting q take real values.

The solution to the relaxed problem with constraint $\tilde{q}^T\tilde{q} = 1$ is given by the following eigenvalue problem:

$$L\tilde{q} = \lambda\tilde{q} \tag{3}$$

The matrix L is the *Laplacian* of the graph and it is defined as $L = D - W$. The sub-optimal solution $\tilde{q}$ is the eigenvector corresponding to the second smallest eigenvalue (also called the Fiedler vector). Then the cluster membership indicator is obtained by binarizing $\tilde{q}$ using a suitable threshold θ:

$$q_i = \mathrm{sign}(\tilde{q}_i - \theta),\ i = 1, \ldots, N.$$

The *mincut* criterion has a bias for separating small sets of points. This is due to the fact that there are no restrictions in (1) related to the size or the balance of the clusters.

B. The Normalized Cut

The normalized cut criterion introduced in [14] is defined as:

$$\min_q J_{ncut} = \frac{\mathrm{cut}(\mathcal{A},\mathcal{B})}{d_{\mathcal{A}}} + \frac{\mathrm{cut}(\mathcal{A},\mathcal{B})}{d_{\mathcal{B}}} \tag{4}$$

where $d_{\mathcal{A}} = \sum_{i\in\mathcal{A}} d_i$. This criterion penalizes small sets or isolated points by taking into account the total weight of each cluster.

Equation (4) can be rewritten as (see, [14]):

$$\min_q J_{ncut} = \frac{q^T L q}{q^T D q} \tag{5}$$

such that $q \in \{-b, 1\}^N$, $q^T D 1_N = 0$, where $b = \sum_{i\in\mathcal{A}} d_i / \sum_{j\in\mathcal{B}} d_j$ and 1_N is $N \times 1$ vector of ones. Minimizing the *ncut* criterion is NP-complete [14]. However, an approximate solution can be found efficiently by relaxing the discrete constraint on q.

If q can take real values then J_{ncut} corresponds to the Rayleigh quotient of the following generalized eigenvalue problem:

$$L\tilde{q} = \lambda D\tilde{q}. \tag{6}$$

The constraint $\tilde{q}^T D 1_N = 0$ is automatically satisfied in the generalized eigenproblem. The relaxed solution to the normalized cut is the Fiedler vector.

C. Markov Random Walks

A probabilistic interpretation of spectral clustering was introduced in [11]. The *affinity* matrix W of a graph can be used to define a random walk with stochastic matrix P. The ij-th entry of P is the probability of moving from node i to node j in one step. This stochastic matrix can be obtained as: $P = D^{-1}W$ such that each of its rows sum to 1. The corresponding eigenvalue problem becomes:

$$Pr = \xi r. \tag{7}$$

with ordered eigenvalues $\xi_1 \geq \xi_2 \geq \ldots \geq \xi_N$ and corresponding eigenvectors $r^{(1)}, r^{(2)}, \ldots, r^{(N)}$. The relaxed solution becomes the second eigenvector $r^{(2)}$. This is due to the fact that the first eigenvector is a vector of ones.

Consider the generalized eigenvalue problem of the normalized cut (6):

$$L\tilde{q} = \lambda D\tilde{q}$$

premultiplying by D^{-1} leads to:

$$\begin{aligned} D^{-1}(D-W)\tilde{q} &= \lambda\tilde{q} \\ (I_N - D^{-1}W)\tilde{q} &= \lambda\tilde{q} \\ (I_N - P)\tilde{q} &= \lambda\tilde{q} \end{aligned} \tag{8}$$

where I_N is the $N \times N$ identity matrix. This shows that $r = \tilde{q}$ and $\lambda = 1 - \xi$.

D. The NJW Algorithm

A different spectral algorithm was proposed in [12], where the *affinity* matrix is first normalized using symmetric divisive normalization. The resulting eigenvalue problem is:

$$D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tilde{q} = \lambda\tilde{q}. \tag{9}$$

Premultiplying by $D^{-\frac{1}{2}}$:

$$D^{-1} W D^{-\frac{1}{2}} \tilde{q} = \lambda D^{-\frac{1}{2}} \tilde{q}$$

and defining $u = D^{-\frac{1}{2}}\tilde{q}$, one obtains:

$$\begin{aligned} D^{-1}Wu &= \lambda u \\ Pu &= \lambda u \end{aligned} \tag{10}$$

which shows the relation between the random walks algorithm and the NJW algorithm.

E. Kernel Alignment

Kernel alignment was introduced in [6] as a measure of similarity between a kernel and an objective function or between two kernel functions. If one considers the cluster membership as the objective function, then the alignment can be used for clustering as a measure of the partition cut cost [7].

The empirical alignment of a kernel function $K(x,y)$ with the cluster membership indicator vector q with respect to the set of data points $\mathcal{D}$ is defined as follows:

$$\hat{A}(K,q,\mathcal{D}) = \frac{q^T\Omega q}{N||\Omega||_F} \tag{11}$$

where Ω denotes the kernel matrix with ij-th entry equal to $K(x_i,x_j)$.

The problem of maximizing the alignment can be formulated as follows:

$$\begin{aligned} \max_q \hat{A}(K,q,\mathcal{D}) &= q^T\Omega q \\ \text{such that} &\quad q\in\{-1,1\}^N. \end{aligned} \tag{12}$$

Letting q take real values such that $\tilde{q}^T\tilde{q}=1$ leads to the eigendecomposition of the kernel matrix:

$$\Omega\tilde{q} = \lambda\tilde{q} \tag{13}$$

which corresponds to kernel PCA [13], [16]. The approximate alignment problem is then solved by the first eigenvector $\tilde{q}^{(1)}$ and it is upper bounded by a multiple of the first eigenvalue λ_1 (see, [7]). The clusters can be obtained by choosing a threshold θ that gives maximal alignment of:

$$q_i = \text{sign}(\tilde{q}_i^{(1)} - \theta), i = 1,\dots N.$$

III. Kernel PCA and Spectral Clustering

A. LS-SVM approach to kernel PCA

It was shown in [16] that kernel PCA is the dual solution to a primal optimization problem formulated in a kernel induced feature space. In this formulation, the underlying loss function associated to kernel PCA is explicit.

Given a set of data points $\{x_i\}_{i=1}^N, x_i\in\mathbb{R}^n$, the objective of kernel PCA is to find projected variables $e=\Phi_c w$ with maximal variance together with minimizing a regularization term.

$$\begin{aligned} \max_{w,e} J_p(w,e) &= \gamma\frac{1}{2}e^Te - \frac{1}{2}w^Tw \\ \text{such that} &\quad e=\Phi w \end{aligned} \tag{14}$$

where Φ is the $N\times n_h$ feature matrix:

$$\Phi_c = \begin{bmatrix} \varphi(x_1)^T - \hat{\mu}_\varphi^T \\ \varphi(x_2)^T - \hat{\mu}_\varphi^T \\ \vdots \\ \varphi(x_N)^T - \hat{\mu}_\varphi^T \end{bmatrix},$$

γ is the regularization parameter, $\hat{\mu}_\varphi = (1/N)\sum_{i=1}^N\varphi(x_i)$ and $\varphi(\cdot):\mathbb{R}^n\to\mathbb{R}^{n_h}$ is the mapping to a high dimensional feature space of dimension n_h (which can be infinite dimensional).

The Lagrangian for this constrained optimization problem is:

$$\mathcal{L}(w,e;\alpha) = \frac{\gamma}{2}e^Te - \frac{1}{2}w^Tw - \alpha^T(e-\Phi_c w)$$

with conditions for optimality given by:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial w} = 0 &\quad\to\quad w = \Phi_c^T\alpha \\ \frac{\partial\mathcal{L}}{\partial e} = 0 &\quad\to\quad \alpha = \gamma e \\ \frac{\partial\mathcal{L}}{\partial\alpha} = 0 &\quad\to\quad e = \Phi_c w. \end{aligned}$$

Eliminating e,w and defining $\lambda = 1/\gamma$ leads to:

$$\Phi_c\Phi_c^T\alpha = \lambda\alpha.$$

Applying the kernel trick $\varphi(x)^T\varphi(y) = K(x,y)$ one obtains the following eigenvalue problem:

$$\Omega_c\alpha = \lambda\alpha$$

where Ω_c is the centered kernel matrix with kl-entry: $\Omega_{c,kl} = K(x_k,x_l) - \frac{1}{N}\sum_{r=1}^N K(x_k,x_r) - \frac{1}{N}\sum_{r=1}^N K(x_l,x_r) + \frac{1}{N^2}\sum_{r=1}^N\sum_{s=1}^N K(x_r,x_s)$.

The projected variables of a test data point x (also called the score variables) become:

$$\begin{aligned} z_i(x) = \sum_{l=1}^N \alpha_l^{(i)}\Bigg(K(x_l,x) - \frac{1}{N}\sum_{r=1}^N K(x_r,x) \\ - \frac{1}{N}\sum_{r=1}^N K(x_r,x_l) + \frac{1}{N^2}\sum_{r=1}^N\sum_{s=1}^N K(x_r,x_s)\Bigg) \end{aligned} \tag{15}$$

where $\alpha^{(i)}$ denotes the i-th eigenvector.

B. Weighted Kernel PCA

A weighting scheme to extend kernel PCA to general loss functions was proposed in [2]. This scheme was used as a fast and efficient technique to impose robustness and sparseness in kernel PCA. The weights are chosen in such a way that outlying data points are downweighted. In this paper we consider a different kind of weighting.

By extending the formulation to a symmetric positive definite weight matrix V, the following primal problem can be formulated:

$$\begin{aligned} \max_{w,e} J_p(w,e) &= \gamma\frac{1}{2}e^TVe - \frac{1}{2}w^Tw \\ \text{such that} &\quad e=\Phi w \end{aligned} \tag{16}$$

with $\Phi = [\varphi(x_1)^T;\varphi(x_2)^T;\dots;\varphi(x_N)^T]$, $V=V^T>0$.

The Lagrangian becomes:

$$\mathcal{L}(w,e;\alpha) = \frac{\gamma}{2}e^TVe - \frac{1}{2}w^Tw - \alpha^T(e-\Phi w)$$

with conditions for optimality given by:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial w} = 0 &\quad\to\quad w = \Phi^T\alpha \\ \frac{\partial\mathcal{L}}{\partial e} = 0 &\quad\to\quad \alpha = \gamma Ve \\ \frac{\partial\mathcal{L}}{\partial\alpha} = 0 &\quad\to\quad e = \Phi w. \end{aligned}$$

By elimination of w and e, one obtains the following non-symmetric eigenvalue problem:

$$V\Omega\alpha = \lambda\alpha \tag{17}$$

where Ω is the kernel matrix with kl-entry: $\Omega_{kl} = K(x_k, x_l)$, the ordered eigenvalues are $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$ and corresponding eigenvectors $\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(N)}$. The matrix $V\Omega$ may not be symmetric but since V and Ω are positive definite, the product $V\Omega$ is also positive definite (see, [9]).

The i-th score variables of a test data point x become:

$$z_i(x) = w^T\varphi(x) = \sum_{l=1}^{N} \alpha_l^{(i)} K(x_l, x). \tag{18}$$

In general, a kernel function can be regarded as a similarity measure between two data points. Therefore, one can use the kernel matrix as an *affinity* matrix.

Note that (17) corresponds to the alignment (13) if $V = I_N$. In the same way, if $V = D^{-1}$ then weighted kernel PCA corresponds to the random walks algorithm (7). Hence, by changing V it is possible to solve the spectral clustering methods described earlier using weighted kernel PCA. Using this kernel PCA interpretation, it is also possible to use the variance associated to the eigenvector solution as a model selection criterion to find the kernel parameters.

Table I shows the relation between weighted kernel PCA and some spectral clustering methods. The last column indicates which eigenvector of (17) is the relaxed solution of the spectral method, given V.

TABLE I

RELATION BETWEEN WEIGHTED KERNEL PCA AND SOME SPECTRAL CLUSTERING METHODS

Method	Original Problem	V	Relaxed Solution
Alignment	$\Omega q = \lambda q$	I_N	$\alpha^{(1)}$
NCut	$Lq = \lambda Dq$	D^{-1}	$\alpha^{(2)}$
Random walks	$D^{-1}Wq = \lambda q$	D^{-1}	$\alpha^{(2)}$
NJW	$D^{-\frac{1}{2}}WD^{-\frac{1}{2}}q = \lambda q$	D^{-1}	$D^{\frac{1}{2}}\alpha^{(2)}$

The NCut and the random walks algorithms lead to the same equivalent problem as shown in (8). The NJW algorithm also leads to the same generalized eigenvalue problem but the relaxed solution has to be normalized.

Note that for the last three methods, the corresponding eigenvalue problem is $D^{-1}\Omega\alpha = \lambda\alpha$. Premultiplying by D (assumed to be full rank) leads to the generalized eigenvalue problem $\Omega\alpha = \lambda D\alpha$ which is numerically more stable to be solved.

C. Centering by including a bias term

Considering a bias term into the LS-SVM formulation to kernel PCA leads to a centering of the kernel matrix as shown in [16],[15]. Introducing a bias term into the weighted kernel PCA formulation leads to the following primal problem:

$$\begin{aligned} \max_{w,e,b} J_p(w,e) &= \gamma\frac{1}{2}e^TVe - \frac{1}{2}w^Tw \\ \text{such that} &\quad e = \Phi w + b1_N. \end{aligned} \tag{19}$$

The Lagrangian becomes:

$$\mathcal{L}\{w, e; \alpha\} = \frac{\gamma}{2}e^TVe - \frac{1}{2}w^Tw - \alpha^T(e - \Phi w - b1_N)$$

with conditions for optimality given by:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial w} = 0 &\quad\rightarrow\quad w = \Phi^T\alpha \\ \frac{\partial\mathcal{L}}{\partial e} = 0 &\quad\rightarrow\quad \alpha = \gamma Ve \\ \frac{\partial\mathcal{L}}{\partial b} = 0 &\quad\rightarrow\quad 1_N^T\alpha = 0 \\ \frac{\partial\mathcal{L}}{\partial \alpha} = 0 &\quad\rightarrow\quad e = \Phi w + b1_N. \end{aligned}$$

The bias term becomes:

$$b = -\frac{1}{1_N^TV1_N}1_N^TV\Omega\alpha \tag{20}$$

eliminating w and e and defining $\lambda = 1/\gamma$, one obtains the following eigenvalue problem:

$$M\Omega\alpha = \lambda\alpha \tag{21}$$

where M is defined as:

$$M = V - \frac{1}{1_N^TV1_N}V1_N1_N^TV. \tag{22}$$

Note that if $V = I$ then M becomes the centering matrix: $I - \frac{1}{N}1_N1_N^T$. This shows that including a bias term in weighted kernel PCA leads to a different centering. The weight matrix V can be chosen such that $M = D^{-1}$. In this way, it is possible to link the weighted kernel PCA formulation with bias term to the spectral clustering methods already discussed. However, this problem leads to solving a set of nonlinear equations.

As in (18), the i-th score variables become:

$$z_i(x) = w^T\varphi(x) + b = \sum_{l=1}^{N} \alpha_l^{(i)} K(x_l, x) + b.$$

D. Out-of-Sample Extension

Out-of-sample extensions for several unsupervised learning algorithms have been discussed in [4]. Using these extensions, it is possible to apply the trained clustering model to out-of-sample data points (test points). The Nyström method is used to evaluate the test points in the approximated eigenfunction.

It is important to mention that using the weighted kernel PCA formulation to spectral clustering, the extension to out-of-sample data points is straightforward. This is due to the fact that it is possible to perform an exact evaluation of the new data points in the trained weighted kernel PCA model using (15). Therefore, an approximation of the eigenfunction like the Nyström method is not needed. A new point x can be evaluated as follows:

$$q(x) = \text{sign}(w^T\varphi(x) - \theta) = \text{sign}\Big(\sum_{l=1}^{N}\alpha_l K(x_l, x) - \theta\Big). \quad (23)$$

IV. Empirical Results

In this section, experimental results are presented. Emphasis is given on the out-of-sample extension. The kernel alignment and the variance criterion were used to obtain the RBF kernel parameter σ^2. We performed 10 randomizations of the training and validation sets and the criterion was evaluated on validation data. For the threshold θ the alignment was used. We used the adjusted Rand index [10] as a external validation criterion to compare the clustering results with the true labels. The expected value of the original Rand index for two random partitions does not take a constant value. The adjusted Rand index corrects this bias and its range varies from 0 to 1. An adjusted Rand index of 1 means perfect match between the true labels and the clustering results. We also compared the Nyström out-of-sample extension described in [4] with the proposed method.

A. Toy Problem

The first experiment consisted of two linearly separable Gaussian clusters. The training scenario consisted of 300 data points for training and 100 data points for validation. Figure 1 shows the tuning curves for the alignment and the variance criterion. The obtained σ^2 was 1.25 and 2.80 for the variance and the alignment respectively. The adjusted Rand index and the alignment can be seen in Figure 2. This plot shows that the maximal alignment corresponds to high adjusted Rand index. It is important to mention that only the alignment was used to find the threshold. The adjusted Rand index is plotted just to confirm that the alignment is a good criterion to tune θ. Figure 3 shows the clustering results for test data points sampled from the same cluster distributions. Figure 4 shows the results for test points sampled from different distributions. This shows the generalization of new points far away from the clusters. The training data is depicted in gray and the test data points in black.

B. Iris dataset

In this experiment, we used 100 datapoints from the UCI Iris dataset. The non-linearly separable classes were used. The training scenario consisted of 60 data points for training, 20 data points for validation and 20 samples were left out for testing. Figure 5 shows the tuning curves used to obtain the kernel parameter. Figure 6 show the alignment used to find the threshold. The kernel matrices before and after sorting the data points using the relaxed solution are shown in Figure 7. The sorted matrix exhibits a block structure. Table II shows the adjusted Rand index for the 20 test data points, indicating the results obtained using the out-of-sample extension discussed in [4] and the proposed approach.

TABLE II

Iris dataset - Comparison of the adjusted Rand index of the Nyström out-of-sample extension and the proposed approach

Method	Nyström RBF k.	Weighted KPCA RBF k.
Alignment	1	1
NCut	0.58	0.62
NJW	0.58	0.62

C. Golub microarray dataset

We report clustering results on the Golub microarray dataset. The training scenario consisted of 22 samples (11 positive, 11 negative) for training and 14 samples for validation. The test set consisted of 14 samples (also balanced). The tuning curves for the kernel parameter were similar to the previous experiments. Figure 8 shows the alignment with respect to the threshold. The adjusted Rand index for the test data is shown in Table III.

TABLE III

Golub microarray dataset - Comparison of the adjusted Rand index of the Nyström out-of-sample extension and the proposed approach

Method	Nyström RBF k.	Weighted KPCA RBF k.
Alignment	0.6	0.6
NCut	0.63	0.72
NJW	0.72	0.86

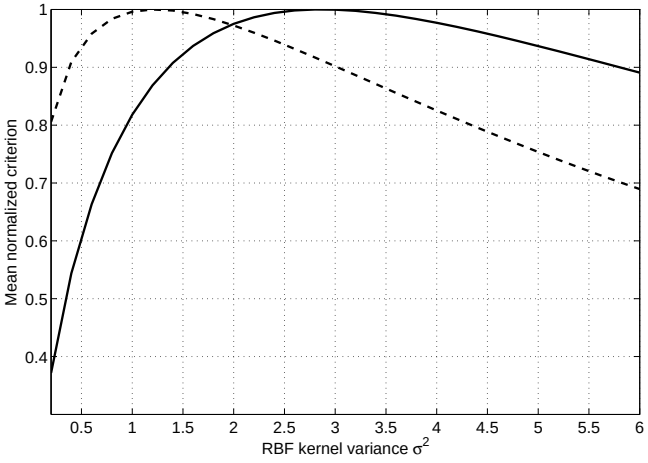


Fig. 1. Toy problem - tuning the RBF kernel parameter using validation data. Variance criterion (solid line), alignment (dashed line)

V. Conclusions

A new weighted kernel PCA formulation was derived as a unifying view of spectral clustering. It was shown that techniques such as the kernel alignment, the normalized cut and the NJW algorithm are particular cases of weighted kernel PCA. Using this interpretation it is possible to evaluate new test points in an exact way, which is important for predictive purposes. A maximal variance criterion can also

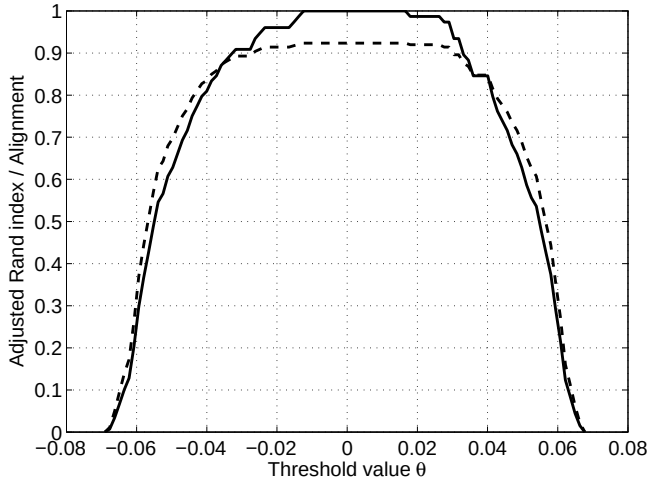


Fig. 2. Toy problem - tuning the threshold. Adjusted Rand index (solid line), alignment (dashed line). Maximal alignment corresponds to high Adjusted Rand index.

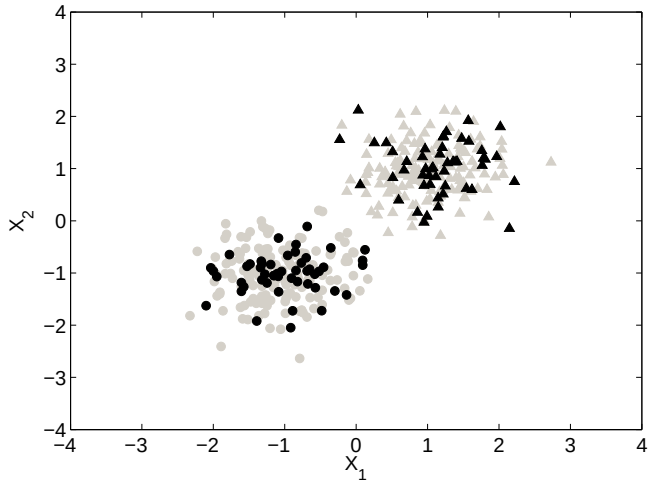


Fig. 3. Toy problem - clustering results. Test data sampled from the same cluster distributions. Training data depicted in gray, test data in black.

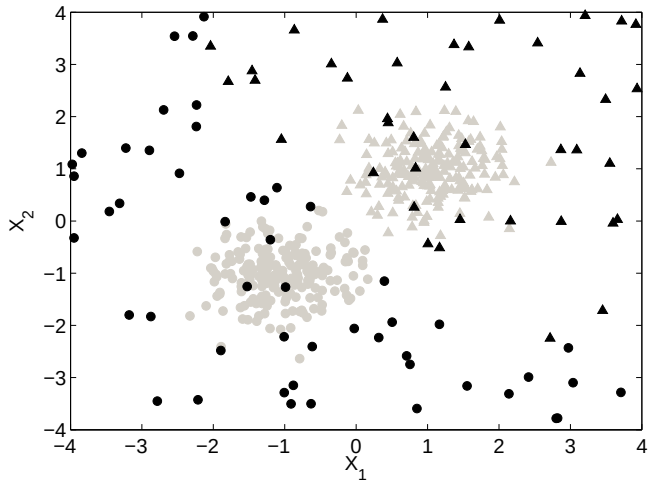


Fig. 4. Toy problem - clustering results. Test data sampled from different cluster distributions. Training data depicted in gray, test data in black.

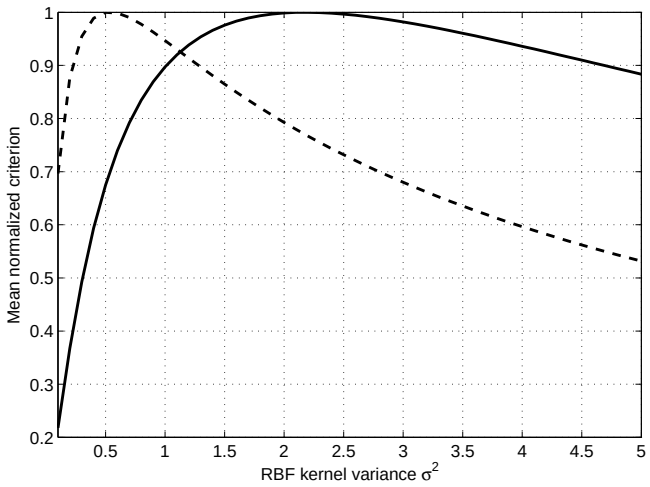


Fig. 5. Iris dataset - tuning the RBF kernel parameter using validation data. Variance criterion (solid line), alignment (dashed line).

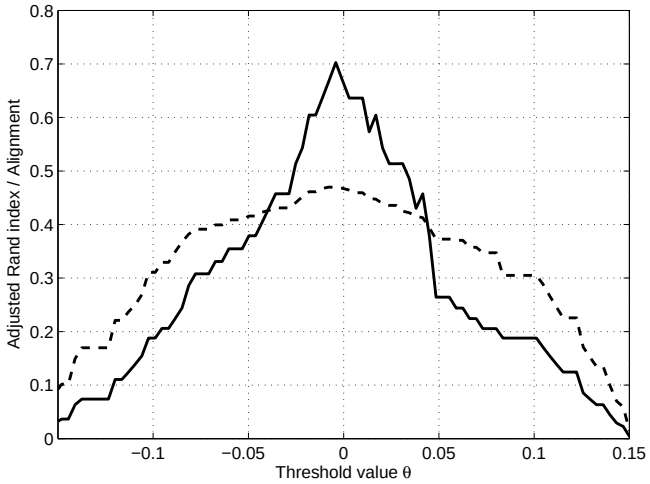
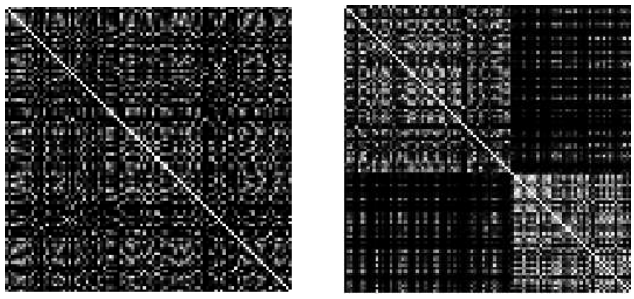


Fig. 6. Iris dataset - tuning the threshold. Adjusted Rand index (solid line), alignment (dashed line). Maximal alignment corresponds to high Adjusted Rand index.



(a) Original Iris data

(b) Iris data after sorting using the relaxed solution.

Fig. 7. Kernel matrices of the Iris data before and after sorting revealing cluster block structure. Whiter points correspond to high similarity.

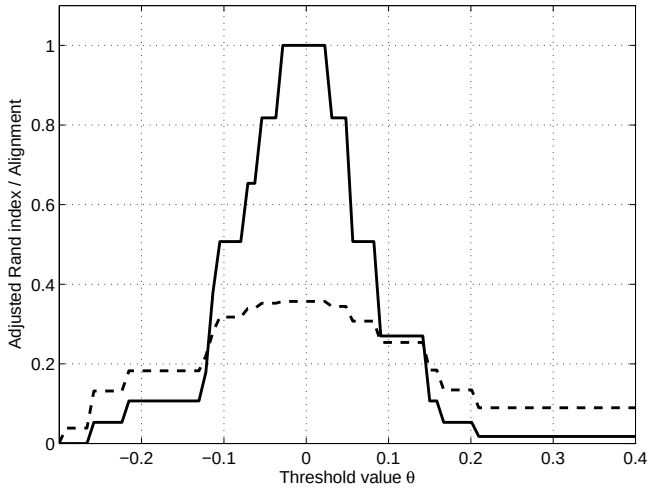


Fig. 8. Golub microarray dataset - tuning the threshold. Adjusted Rand index (solid line), alignment (dashed line).

be used to perform model selection in order to find the kernel parameters. We also include a bias term into the weighted formulation, showing that it leads to a different centering of the kernel matrix. From simulation results, the proposed method performed better compared to a existing out-of-sample extension which is an approximation of the underlying eigenfunction.

ACKNOWLEDGMENT

This work was supported by grants and projects for the Research Council K.U.Leuven (GOA-Mefisto 666, GOA-Ambiorics, several PhD / Postdocs & fellow grants), the Flemish Government FWO: PhD / Postdocs grants, projects G.0240.99, G.0407.02, G.0197.02, G.0080.01, G.0141.03, G.0491.03, G.0120.03, G.0452.04, G.0499.04, ICCoS, ANMMM; AWI;IWT:PhD grants, GBOU (McKnow) Soft4s, the Belgian Federal Government (Belgian Federal Science Policy Office: IUAP V-22; PODO-II (CP/01/40), the EU(FP5-Quprodis;ERNSI, Eureka 2063-Impact;Eureka 2419-FLiTE) and Contracts Research/Agreements (ISMC/IPCOS, Data4s, TML, Elia, LMS, IPCOS, Mastercard). Johan Suykens is an associate professor at the K.U.Leuven, Belgium. The scientific responsibility is assumed by its authors.

REFERENCES

[1] C. Alzate and J. A. K. Suykens. Extending kernel principal component analysis to general underlying loss functions. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'05)*, pages 214–219, 2005.

[2] C. Alzate and J. A. K. Suykens. Kernel principal component analysis using an epsilon insensitive robust loss function. Internal report 06-03. Submitted for publication, ESAT-SISTA, K. U. Leuven, 2006.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 585–591, Cambridge, MA, 2002. MIT Press.

[4] Y. Bengio, J. F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

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

[6] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[7] N. Cristianini, J. Shawe-Taylor, and J. Kandola. Spectral kernel methods for clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 649–655, Cambridge, MA, 2002. MIT Press.

[8] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory. *Czech. Math. J.*, 25(100):619–633, 1975.

[9] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[10] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, pages 193–218, 1985.

[11] M. Meila and J. Shi. Learning segmentation by random walks. In T.K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*. MIT Press, 2001.

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

[13] B. Schölkopf, A. Smola, and K. R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[14] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

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

[16] J. A. K. Suykens, T. Van Gestel, J. Vandewalle, and B. De Moor. A support vector machine formulation to PCA analysis and its kernel version. *IEEE Trans. Neural Networks*, 14(2):447–450, March 2003.