ADCAIJ: Advances in Distributed Computing and Articial Intelligence Journal
Regular Issue, Vol. 6 N. 1 (2017), 31-40
eISSN: 2255-2863
DOI: http://dx.doi.org/10.14201/ADCAIJ2017613140

Kernel-based framework for spectral dimensionality reduction and clustering formulation: A theoretical study

X. Blanco-Valencia[a], M. A. Becerra[b,c], A. E. Castro-Ospina[c], M. Ortega-Adarme[d], D. Viveros-Melo[d], J. C. Alvarado-Pérez[a,e], and D. H. Peluffo-Ordóñez[f]

[a]Universidad de Salamanca, Spain
[b]Institución Universitaria Salazar y Herrera, Colombia
[c]Research Center of the Instituto Tecnológico Metropolitano, Colombia
[d]Universidad de Nariño, Colombia
[e]Coorporación Universitaria Autónoma de Nariño, Colombia
[f]Universidad Técnica del Norte, Ecuador - dhpeluffo@utn.edu.ec

KEYWORD

Kernel PCA; Spectral clustering; Support vector machine.

ABSTRACT

This work outlines a unified formulation to represent spectral approaches for both dimensionality reduction and clustering. Proposed formulation starts with a generic latent variable model in terms of the projected input data matrix. Particularly, such a projection maps data onto a unknown high-dimensional space. Regarding this model, a generalized optimization problem is stated using quadratic formulations and a least-squares support vector machine. The solution of the optimization is addressed through a primal-dual scheme. Once latent variables and parameters are determined, the resultant model outputs a versatile projected matrix able to represent data in a low-dimensional space, as well as to provide information about clusters. Particularly, proposed formulation yields solutions for kernel spectral clustering and weighted-kernel principal component analysis.

1. Introduction

In pattern recognition, the term kernel is used to define a function that establishes the similarity among given input elements. Therefore, a kernel function enables learning methods to use similarities for representing the samples or data points, instead of using explicitly the input data matrix [Belanche Muñoz, 2013]. Kernel- based methods have been widely exploited for both supervised and unsupervised learning approaches showing their usability and versatility in several applications [Aldrich and Auret, 2013], such as image segmentation [Wu et al., 2015, Molina-Giraldo et al., 2012], time-varying data analysis and complex dynamic data clustering [Langone et al., 2013, Peluffo-Ordónez et al., 2013], and hypothesis testing [Harchaoui et al., 2013], among others. This article explores the benefit of using a kernel model within the design of spectral formulations of clustering and unsupervised dimensionality reduction methods.

On one side, kernel methods are of interest since they allow to incorporate prior knowledge into the clustering procedure [Filippone et al., 2008]. In case of unsupervised clustering methods (that is to say, when clusters are naturally formed by following a given partition criterion), a set of initial parameters should be properly selected to avoid any local optimum solution distant from the desired global optimum. Indeed, in spectral clustering (SC), such initial parameters are traditionally the number of clusters and the input kernel matrix itself. On the other side, the aim of dimensionality reduction (DR) is to extract a lower dimensional, relevant information from high-dim ensional data, being then a key stage for the design of pattern recognition systems. Indeed, when using adequate DR stages, the system performance can be enhanced as well as the data visualization can become more intelligible [Alvarado-Pérez and Peluffo-Ordóñez, 2015, Alvarado-Pérez et al., 2015]. Recent methods of DR are focused on the data topology preservation [Peluffo-Ordóñez et al., 2014b]. Mostly such a topology is driven by graph-based approaches where data are represented by a similarity matrix, and it is then susceptible to be expressed in terms of a kernel matrix [Ham et al., 2004], which means that a wide range of methods can be set within a kernel principal component analysis (KPCA) framework [Peluffo-Ordonez et al., 2014]. At the moment to choose a method for either SC or DR, aspects such as nature of data, complexity, aim to be reached and problem to be solved should be taken into consideration. In this regard, it must be quoted that there exists a variety of spectral methods making then the selection of a method a nontrivial task. In fact, some problems may require the combination of methods so that the properties of different methods are simultaneously exploited [Peluffo-Ordónez et al., 2015]. Some works have studied the benefit of taking advantage simultaneously of DR and SC techniques. For instance, in [Peluffo-Ordóñez et al., 2014a], a DR approach (linear feature extraction) is used to enhance the clustering performance by performing the grouping process over the projected data rather than over the original data. Other works are focused on generating variable relevance [Wolf and Bileschi, 2005, Peluffo Ordoñez et al., 2015] or data representation [Wolf and Shashua, 2005] criteria from conventional spectral clustering formulations.

In this work, the authors outline a unified formulation able to explain kernel approaches for both spectral clustering (SC) and unsupervised DR. Such a formulation starts with a latent variable model of a high-dimensional representation of the input data matrix, involving implicitly a mapping function. The model is incorporated within a quadratic functional, which along with an orthonormal constraint constitutes our optimization problem being a non-supervised version of a least-square-support-vector-machine (LS-SVM) formulation. Its solution is accomplished by relaxing the problem, and following a primal-dual scheme, which readily leads to a kernel representation given the quadratic nature of the functional. The proposed formulation represents a framework to easily understand the relationship between kernel-based approaches for SC and unsupervised DR. Also, the resultant model yields explicitly the solution of two well-known methods, namely the so-called kernel spectral clustering (KSC) proposed in [Alzate and Suykens, 2010], and a version of weighted kernel PCA (WKPCA) [Peluffo-Ordonez et al., 2014].

The rest of this paper is organized as follows: Section 2 presents a brief overview on kernels. Section 3 describes our unified formulation, and explains the SC and DR perspectives. Finally, some final remarks are drawn in section 4.

2. Overview on kernels

For following statements, let us consider the following notation: Let $\boldsymbol{Y} \in \mathbb{R}^{D\times N}$ be the input data matrix formed by N samples (data points), denoted by $\boldsymbol{y}_i \in \mathbb{R}^D$ with $i \in \{1,\dots,N\}$. As well, from another point of view, it is conformed by D variables such that $\boldsymbol{y}^{(\ell)} \in \mathbb{R}^N$ is the ℓ-th variable, with $\ell \in \{1,\dots,D\}$. Mathematically, kernels involve a mapping process from a d-dimensional input space representing a data set to a (d_h-) high-dimensional space, where $d_h >> D$. In terms of pattern recognition, the advantage of mapping the original data space onto a higher one lies in the fact that the latter space may provide a better data representation regarding cluster separability. Furthermore, it must be taken into account that the mapping is done before carrying out any clustering process. Then, the success of the data clustering task can be partly attributed to the kernel-matrix-building function when grouping algorithms are directly associated with the chosen kernel.

Currently, kernels with special structure aimed to attend particular interests have been proposed. For instance, in [Seeland et al., 2012], a structural clustering kernel is introduced by incorporating similarities induced by a structural clustering algorithm to improve graph kernels recommended by literature. Mercer kernels have been used for solving multi-cluster problems [Domeniconi et al., 2011]. In [Belanche Muñoz, 2013], different kernels (generative, convolution, and covariance kernels, among others) are explained as well as important developments on how to construct kernels from a generating function are described.

In terms of human learning theory, one of the fundamental problems is the discrimination among elements or objects. Consider the following instance: We have a set of objects formed by two different classes; then, when a new object appears the classification and/or visualization task is to determine to which class such an object belongs. This is usually done by taking into account the object's properties as well as similarities and differences with regards to the two previously known classes. According to the above, and regarding kernel theory, we need to create or choose a similarity or affinity measure to compare the data. Since such similarities are non-negative, kernel functions are positive-definite. A kernel function can be defined in the form:

$$\begin{aligned} \mathcal{K}(\cdot,\cdot) : \mathbb{K}^D \times \mathbb{K}^D &\longrightarrow \mathbb{K} \\ \boldsymbol{y}_i, \boldsymbol{y}_j \quad &\longmapsto \mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_j), \end{aligned} \tag{1}$$

where $\mathbb{K} = \mathbb{C}$ or $\mathbb{R}$. Note that in this case we have assumed elements $\boldsymbol{y}_i$ to be real and D-dimensional. Then, for a total of N data points, we can arrange the kernel function values into a $N \times N$ matrix $\boldsymbol{K}$ with entries $k_{ij} = \mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_j)$, called Gram matrix or kernel matrix as well. Such a matrix is positive-semidefinite, i.e., a $N \times N$ complex matrix satisfying $\sum_{i=1}^{N}\sum_{j=1}^{N} c_i \bar{c}_j k_{ij} \geq 0$, for all $c_i \in \mathbb{C}$, being $\bar{c}_i$ the complex conjugate of c_i. Similarly, a real symmetric $N \times N$ matrix $\boldsymbol{K}$ satisfying the same condition given for all $c_i \in \mathbb{R}$ is also called positive-semidefinite. In terms of spectral matrix analysis, a symmetric matrix is positive-semidefinite if and only if all its eigenvalues are non-negative. In the literature, a number of different terms are used for positive-definite kernels, such as reproducing kernel, Mercer kernel, admissible kernel, support vector kernel, non-negative definite kernel and covariance.

2.1 Kernel trick

Now, let us consider a function to map from the D-dimensional space to that d_h dimensional one is in the form $\phi(\cdot)$, such that: $\boldsymbol{\phi}(\cdot) : \mathbb{R}^D \longrightarrow \mathbb{R}^{d_h}, \boldsymbol{y}_i \longmapsto \phi(\boldsymbol{y}_i)$. The matrix $\boldsymbol{\Phi} = \left[\boldsymbol{\phi}(\boldsymbol{y}_1)^\top, \ldots, \boldsymbol{\phi}(\boldsymbol{y}_N)^\top\right]$, $\boldsymbol{\Phi} \in \mathbb{R}^{d_h \times N}$, is a high dimensional representation of the input data matrix $\boldsymbol{Y}$. A sagittal diagram of the mapping function is shown in Figure 1.

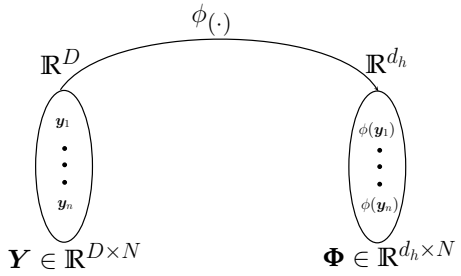

Figure 1: Mapping function to a high dimensional space

An interesting property of the kernel functions is the so-called *kernel trick*. In topology, a kernel function can be seen as an inner product in the domain of Hilbert space $\mathcal{H}$, as follows: $\mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \phi(\boldsymbol{y}_i), \phi(\boldsymbol{y}_i) \rangle_{\mathcal{H}}$. Kernel trick allows for performing the mapping and the inner product simultaneously by defining an associated kernel function. Then, we can estimate the kernel matrix without knowing the mapping function. This property gains importance in kernel theory, since it permits to replace a positive-definite kernel with another kernel that is finite and approximately positive-definite. For instance, from a given algorithm formulated in terms of a

positive-definite kernel $\mathcal{K}$, we can construct an alternative algorithm by replacing it by another positive-definite kernel $\tilde{\mathcal{K}}$ [Schölkopf and Smola, 2002], in such a manner that $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top = \boldsymbol{K}$. Then, in this case, kernel trick has served to estimate $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top$ as $\boldsymbol{K}$. In the domain of $\mathcal{H}$, $\boldsymbol{K}$ holds the inner product of the mapped data points (rows of matrix $\boldsymbol{\Phi}$), or -from another point of view- the outer product of the mapped variables (columns of matrix $\boldsymbol{\Phi}$).

2.2 Types of kernel functions

Radial basis function (RBF) kernels are those that can be written in terms of similarity or dissimilarity measure, in the form:

$$\mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_j) = f(\mathrm{d}(\boldsymbol{y}_i, \boldsymbol{y}_j)), \tag{2}$$

where $\mathrm{d}(\cdot,\cdot)$ is a measure on the domain of $\boldsymbol{Y}$, in this case $\mathbb{R}^D$, so:

$$\begin{aligned} \mathrm{d}(\cdot,\cdot) :& \mathbb{R}^D \times \mathbb{R}^D \longrightarrow \mathbb{R}^+ \\ & \boldsymbol{y}_i, \boldsymbol{y}_j \longmapsto \mathrm{d}(\boldsymbol{y}_i, \boldsymbol{y}_j) \end{aligned} \tag{3}$$

and f is a function defined on $\mathbb{R}^+$. Usually, such measure arises from the inner product; $\mathrm{d}(\boldsymbol{y}_i, \boldsymbol{y}_j) = \left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\| = \sqrt{\langle \boldsymbol{y}_i - \boldsymbol{y}_j, \boldsymbol{y}_i - \boldsymbol{y}_j \rangle}$. In Table 1, some common kernels recommended by the state of the art are described.

Table 1: Some kernel functions

Kernel name	Definition	Domain
lineal	$\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle$	$\mathbb{R}^D$
Polynomial	$\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle^D$	$\mathbb{R}^D$
Rational quadratic	$1 - \frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2}{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 + \sigma}, \sigma \in \mathbb{R}^+$	$\mathbb{R}^d$
Exponential	$\exp\left(-\frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|}{2\sigma^2}\right), \sigma \in \mathbb{R}^+$	$\mathbb{R}^D$
Gaussian	$\exp\left(-\frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2}{2\sigma^2}\right), \sigma \in \mathbb{R}^+$	$\mathbb{R}^D$

2.2.1 *Special kernels*

- *Scaled Gaussian kernel matrix*

An alternative to the Gaussian kernel is a local scaled version regarding the data point neighborhood as follows:

$$k_{ij} = \exp\left(-\frac{||\boldsymbol{y}_i - \boldsymbol{y}_j||^2}{\sigma_i \sigma_j}\right), \tag{4}$$

where σ_i is the scaling parameter defined as $\sigma_i = ||\boldsymbol{y}_i - \boldsymbol{y}_i(m)||$ being $\boldsymbol{y}_i(m)$ the m-th nearest neighbor to data point $\boldsymbol{y}_i$. The parameter m is established regarding the nature of the input data. This kernel is widely explained in [Zelnik-manor and Perona, 2004].

2.2.2 *Multiple-kernel learning*

Multiple kernel learning (MKL) approaches have emerged to deal with different issues in machine learning, mainly, regarding support vector machines (SVM) [González et al., 2012,Huang et al., 2012]. The intuitive idea of MKL is that learning can be enhanced when using different kernels instead of an unique kernel. Indeed, local analysis provided by each kernel is of benefit to examine the structure of the whole data. Herein, we consider a MKL approach, in which each dimension of matrix $\boldsymbol{X}$ is considered as independent data matrix and then the

resultant kernel is a linear combination of the set of obtained kernels [Molina-Giraldo et al., 2012]. We will denote the ℓ-th variable (column vector) as $\boldsymbol{y}^{(\ell)} = [y_1^{(\ell)}, \ldots, y_N^{(\ell)}]^\top$. A basic multiple kernel learning (MKL) approach can be expressed as a linear combination of variable-related kernels. In particular, for a Gaussian kernel $\boldsymbol{K} \in \mathbb{R}^{N \times N}$, we have:

$$\boldsymbol{K} = \sum_{\ell=1}^{d} \rho_\ell \boldsymbol{K}_\ell \tag{5}$$

where $\boldsymbol{K}_\ell$ is the kernel associated to variable ℓ being each entry

$$k_{ij}^{(\ell)} = \exp\left(-\frac{|y_i^{(\ell)} - y_j^{(\ell)}|^2}{2\sigma^2}\right), \quad \forall i, j \in [N] \tag{6}$$

and $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_D]$ is the vector of coefficients. As explained in [Molina-Giraldo et al., 2012], vector $\boldsymbol{\rho}$ can be estimated under a variable-relevance criterion, for example, a PCA-derived one as follows:

$$\boldsymbol{\rho} = \sum_{\ell=1}^{D} \lambda_\ell \boldsymbol{v}_\ell \circ \boldsymbol{v}_\ell \tag{7}$$

where λ_ℓ and $\boldsymbol{v}_\ell$ are respectively the ℓ-th eigenvalue and eigenvector of the covariance matrix of $\boldsymbol{Y}$ and $\circ$ stands for the Hadamard (element-wise) product.

3. Generalized kernel formulation

This section is aimed at formulating a model and cost function for a multipurpose data representation. To establish our model, let us consider an output data matrix $\boldsymbol{X} \in \mathbb{R}^{d \times N}$, being $d \leq D$ formed by N data points denoted by $\boldsymbol{x}_i \in \mathbb{R}^d$, with $i \in \{1, \ldots, N\}$, as well as by d variables denoted as $\boldsymbol{x}^{(\ell)} \in \mathbb{R}^N$ with $\ell \in \{1, \ldots, d\}$. Also, let us assume an orthonormal projection matrix $\boldsymbol{W} \in \mathbb{R}^{D_h \times d}$, such that $\boldsymbol{W} = [\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(d)}]$ and $\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_d$, where $\boldsymbol{w}^{(\ell)} \in \mathbb{R}^{D_h}$ and $\boldsymbol{I}_d$ is a d-dimensional identity matrix. Since $\boldsymbol{W}$ is orthonormal, elements $\boldsymbol{w}^{(\ell)}$ represent a d-dimensional base and can then generate a new space by means of a linear combination in the form: $\boldsymbol{x}^{(\ell)} = \boldsymbol{w}^{(\ell)} \boldsymbol{\Phi}$. So, the output matrix becomes $\boldsymbol{X} = \boldsymbol{W}^\top \boldsymbol{\Phi}$. Here, in order to add an offset effect, we consider a whole latent variable model as $\boldsymbol{x}^{(\ell)} = \boldsymbol{w}^{(\ell)} \boldsymbol{\Phi} + b_\ell \boldsymbol{1}_N$. Such a model can be expressed in matrix terms as:

$$\boldsymbol{X} = \boldsymbol{W}^\top \boldsymbol{\Phi} + \boldsymbol{b} \otimes \boldsymbol{1}_N^\top, \tag{8}$$

where b_l is a bias term, and $\boldsymbol{b} = [b_1, \ldots, b_d]$, $\otimes$ denotes Kronecker product, and $\boldsymbol{1}_N$ accounts for a N-dimensional all ones vector. Both PCA and SVM, in their simplest formulations, involve an energy term regarding the data matrix. Unlike conventional formulations that starts with a known input matrix, we pose a latent variable model, being unknown both variables (output and mapped data matrix) as well parameters (bias term and projection matrix). By incorporating a weighting matrix $\boldsymbol{\Delta} = \text{Diag}(\delta_1, \ldots, \delta_N)$, the energy term regarding $\boldsymbol{X}$ can be written as $\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top$. Then, a functional in terms of the generalized matrix M-norm [Peluffo Ordoñez et al., 2015] can be expressed as:

$$\frac{1}{N} \text{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top) = ||\boldsymbol{X}||^2_{(1/N)\boldsymbol{\Delta}}. \tag{9}$$

From another point of view, if we define a weighted output data matrix as $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{d \times N}$ as

$$\widetilde{\boldsymbol{X}} = \boldsymbol{X}\ \text{Diag}(\delta_1^{1/2}, \ldots, \delta_N^{1/2}), \tag{10}$$

the functional $\text{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top)$ can also be directly seen as an energy term, so: $\text{tr}(\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^\top)$. Our model can be determined by means of a primal-dual formulation as described below.

Primal formulation: Recalling the functional given in equation (9) and the othonormality condition of projection matrix, we can write the following optimization problem:

$$\max_{\boldsymbol{X},\boldsymbol{W},\boldsymbol{b}} \quad \frac{1}{N}\,\text{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top), \quad \text{s. t.} \quad \boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}_d, \quad \boldsymbol{X} = \boldsymbol{\Phi}\boldsymbol{W} + \boldsymbol{b}\otimes\boldsymbol{1}_N^\top, \tag{11}$$

which can be relaxed as

$$\max_{\boldsymbol{X},\boldsymbol{W},\boldsymbol{b}} \quad \frac{1}{2N}\,\text{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top\boldsymbol{\Gamma}) - \frac{1}{2}\,\text{tr}(\boldsymbol{W}^\top\boldsymbol{W}), \text{s. t.} \quad \boldsymbol{X} = \boldsymbol{W}^\top\boldsymbol{\Phi} + \boldsymbol{b}\otimes\boldsymbol{1}_N^\top, \tag{12}$$

where $\boldsymbol{\Gamma} = \text{Diag}([\gamma_1,\ldots,\gamma_d])$ is a diagonal matrix holding regularization parameters. **Dual formulation:** To solve problem (12), we form the corresponding Lagrangian of problem stated in equation (12), as follows:

$$\mathcal{L} = \frac{1}{2N}\,\text{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top\boldsymbol{\Gamma}) - \frac{1}{2}\,\text{tr}(\boldsymbol{W}^\top\boldsymbol{W}) - \text{tr}(\boldsymbol{A}^\top(\boldsymbol{X} - \boldsymbol{W}^\top\boldsymbol{\Phi} - \boldsymbol{b}\otimes\boldsymbol{1}_N^\top)), \tag{13}$$

where matrix $\boldsymbol{A}\in\mathbb{R}^{N\times n_e}$ holds the Lagrange multiplier vectors, that is, $\boldsymbol{A} = [\boldsymbol{\alpha}^{(1)},\cdots,\boldsymbol{\alpha}^{(n_e)}]$, being $\boldsymbol{\alpha}^{(l)}\in\mathbb{R}^N$ the l-th vector of Lagrange multipliers. Solving the Karush-Kuhn-Tucker (KKT) conditions on (13), we get:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{X}} = \boldsymbol{0}\Rightarrow\boldsymbol{X} = N\boldsymbol{\Delta}^{-1}\boldsymbol{A}\boldsymbol{\Gamma}^{-1}, \quad \frac{\partial\mathcal{L}}{\partial\boldsymbol{W}} = \boldsymbol{0}\Rightarrow\boldsymbol{W} = \boldsymbol{\Phi}\boldsymbol{A},$$

Therefore, by applying Lagrange multipliers and eliminating the primal variables from the initial problem (11), the following eigenvector-based dual solution is obtained: $\boldsymbol{A}\boldsymbol{\Lambda} = \boldsymbol{A}\boldsymbol{\Delta}(\boldsymbol{I}_N + (\boldsymbol{1}_N\otimes\boldsymbol{b}^\top)(\boldsymbol{K}\boldsymbol{\Lambda})^{-1})\boldsymbol{K}$, where $\boldsymbol{\Lambda} = \text{Diag}(\boldsymbol{\lambda})$, $\boldsymbol{\Lambda}\in\mathbb{R}^{N\times N}$, $\boldsymbol{\lambda}\in\mathbb{R}^N$ is the vector of eigenvalues with $\lambda_l = N/\gamma_l$, $\lambda_l\in\mathbb{R}^+$. Again, $\boldsymbol{K}\in\mathbb{R}^{N\times N}$ is a given kernel matrix, satisfying the Mercer's theorem such that $\boldsymbol{\Phi}^\top\boldsymbol{\Phi} = \boldsymbol{K}$. In order to pose a quadratic dual formulation satisfying the condition $\boldsymbol{b}^\top\boldsymbol{1}_N = 0$ by centering vector $\boldsymbol{b}$ (i.e. with zero mean), the bias term is chosen in the form $b_l = -1/(\boldsymbol{1}_N^\top\boldsymbol{\Delta}\boldsymbol{1}_N)\boldsymbol{1}_N^\top\boldsymbol{\Delta}\boldsymbol{K}\boldsymbol{\alpha}^{(l)}$. Therefore, the solution of problem (12) is reduced to the following eigenvector-related problem:

$$\boldsymbol{A}\boldsymbol{\Lambda} = \boldsymbol{\Delta}\boldsymbol{H}\boldsymbol{K}\boldsymbol{A}, \tag{14}$$

where matrix $\boldsymbol{H}\in\mathbb{R}^{N\times N}$ is the centering matrix that is defined as $\boldsymbol{H} = \boldsymbol{I}_N - (1/(\boldsymbol{1}_N^\top\boldsymbol{V}\boldsymbol{1}_N))\boldsymbol{1}_N\boldsymbol{1}_N^\top\boldsymbol{\Delta}$. Imposing a linear independency constraint on Lagrangian vector multipliers, $\boldsymbol{A}$ might be chosen as an orthonorma matrix. In consequence, a feasible solution is to estimate $\boldsymbol{A}$ and $\boldsymbol{\Lambda}$ as the spectral decomposition of a centered weighted kernel matrix $\boldsymbol{\Delta}\boldsymbol{H}\boldsymbol{K}$ -eigenvector and eigenvalue diagonal matrix, respectively. Finally, the output data matrix can be calculated as follows:

$$\boldsymbol{X} = \boldsymbol{A}^\top\boldsymbol{K} + \boldsymbol{b}\otimes\boldsymbol{1}_N^\top. \tag{15}$$

Given this, the solution is determined by the spectrum of a centered weighted kernel matrix and a bias vector defined so that the centering condition is ensured. In the following sections, we show how this solution can be applied for both dimensionality reduction and spectral clustering.

3.1 Dimensionality reduction perspective

Latent data matrix $\boldsymbol{X}$ is given by the linear model $\boldsymbol{W}^\top \boldsymbol{\Phi} + \boldsymbol{b} \otimes \mathbf{1}_N^\top$, which clearly involves a linear combination. If we seek for a low-dimensional representation of input data $\boldsymbol{Y}$,just estimation $\boldsymbol{X}$ with a low-rank version of $\boldsymbol{W}$. Such a estimation of the reduced matrix can be performed on the dual problem solution by using some eigenvectors from $\boldsymbol{A}$.

Weighted kernel PCA: Given that the optimization is done under a maximization criterion, the eigenvectors associated with the largest eigenvalues should be selected. In this sense, final dimension d indicates how many eigenvectors are to be considered. Indeed, the eigenvalues of the centered weighted kernel defines the explained variance, so that the final dimension can be estimated with respect to it. Then, our generalized kernel model represents a weighted kernel PCA formulation when using a low-rank representation of matrix $\boldsymbol{W}$, being then able to embed a D-dimensional data matrix $\boldsymbol{Y}$ into a low-dimensional resulting matrix $\boldsymbol{X}$.

Kernel PCA: To yield conventional kernel PCA, the model should be considered as linear projection in the form $\boldsymbol{X} = \boldsymbol{W}^\top \boldsymbol{\Phi}$. Since d is clearly less than d_h, a low-rank version of $\boldsymbol{\Phi}$ is then $\widehat{\boldsymbol{\Phi}} = \boldsymbol{W}\boldsymbol{X}$. So, we can write a functional to be minimized as $\frac{1}{N}||\boldsymbol{\Phi} - \widehat{\boldsymbol{\Phi}}||_F^2$, which has a dual problem given by:

$$\max_{\boldsymbol{X}} \ \mathrm{tr}(\boldsymbol{X}^\mathsf{T}\boldsymbol{K}\boldsymbol{X}), \quad \text{s. t.} \quad \boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}_d, \tag{16}$$

as widely explained in [Peluffo-Ordonez et al., 2014]. Therefore, a feasible solution is when $\boldsymbol{X}$ are the eigenvectc associated with the d largest eigenvalues. As well, this formulation can be seen as a generalized Weighted PCA when using a Mahalanobis distance regarding any positive-semidefinite matrix [Peluffo-Ordóñez et al., 2014, Peluffo-Ordonez et al., 2014]. Since kernel PCA is derived under the assumption that matrix $\boldsymbol{\Phi}$ has zero mean, centering becomes necessary. To satisfy this condition, we can normalize the kernel matrix with:

$$\begin{aligned} \boldsymbol{K} \leftarrow & \boldsymbol{K} - \frac{1}{N}\boldsymbol{K}\mathbf{1}_N\mathbf{1}_N^\top - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top\boldsymbol{K} + \frac{1}{N^2}\mathbf{1}_N\mathbf{1}_N^\top\boldsymbol{K}\mathbf{1}_N\mathbf{1}_N^\top \\ & = (\boldsymbol{I}_N - \mathbf{1}_N\mathbf{1}_N^\top)\boldsymbol{K}(\boldsymbol{I}_N - \mathbf{1}_N\mathbf{1}_N^\top). \end{aligned} \tag{17}$$

3.2 Clustering perspective

Notice that the primal formulation given in (12) can be seen as a least-squares SVM. Then, our model should be able to provide information about the clusters immersed in data matrix. Since no supervised information is used, grouping process is fully unsupervised.

Kernel spectral clustering: Suppose that the output holds non-encoded information about centroids or prototypes for each cluster. Then, output data points should be represented in low dimension $d = K - 1$, being K the assumed number of clusters. Because each cluster is represented by a single point in the $K-1$-dimensional eigenspace, such that those single points are always in different orthants due also to the KKT conditions, we can encode the eigenvectors considering that two points are in the same cluster if they are in the same orthant in the corresponding eigenspace [Alzate and Suykens, 2010]. Then, a code book can be obtained from the rows of the matrix containing the $K-1$ binarized leading eigenvectors in the columns, by using $\mathrm{sign}(\boldsymbol{x}^{(\ell)})$. Then, matrix $\overline{\boldsymbol{X}} = \mathrm{sgn}(\boldsymbol{X})$ is the code book being each row a codeword. Finally, clusters are formed according to the minimal Hamming distance between codewords within the space of $\overline{\boldsymbol{X}}$. This clustering approach is so-called kernel spectral clustering (KSC), introduced in [Alzate and Suykens, 2010]. Figure 2 depicts graphically the effect of cluster assignment when using a Hamming encoding.

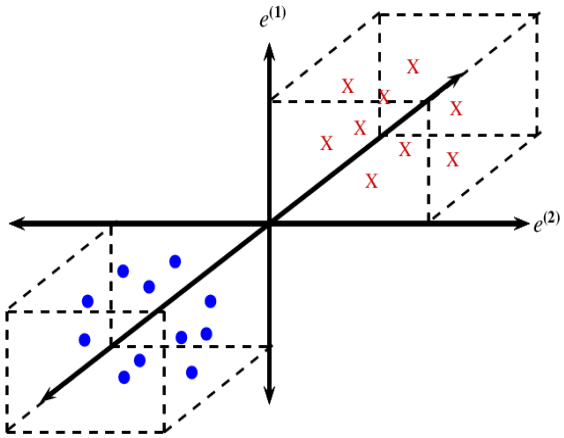

Figure 2: Encoding $\boldsymbol{E} = \text{sign}\,(\boldsymbol{E})$*. The example shows a two clusters problem. Since each single cluster is located in a different orthant, a feasible encoding is by using the sign function and Hamming distance becomes a proper measure to assign elements to a cluster according to the minimal distance*

Out-of-samples extension: The big advantage of this approach is that it can be extended to out-of-samples analysis without re-clustering the whole data to determine the assignment cluster membership for new testing data [Alzate and Suykens, 2010]. In particular, defining $\boldsymbol{z} \in \mathbb{R}^d$ as the projection vector of a testing data point $\boldsymbol{y}_{\text{test}}$, and by taking into consideration the training clustering model, the testing projections can be computed as $\boldsymbol{z} = \boldsymbol{A}^\top \boldsymbol{K}_{\text{test}} + \boldsymbol{b}$, where $\boldsymbol{K}_{\text{test}} \in \mathbb{R}^d$ is the kernel vector such that$\boldsymbol{K}_{\text{test}} = [K_{\text{test}_1}, \ldots, K_{\text{test}_N}]^\top$, where $K_{\text{test}_i} = \mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_{\text{test}})$. Once, the test projection vector $\boldsymbol{z}$ is computed, a decoding stage is carried out that consists of comparing the binarized projections with respect to the codewords in the code book $\overline{\boldsymbol{X}}$ and assigning cluster membership based on the minimal Hamming distance [Alzate and Suykens, 2010].

4. Final remarks

The aim of this paper is to state a generalized formulation able to explain the close relationship between spectral clustering and dimensionality reduction, within a kernel-based framework. Specifically, it has been shown that a least-square-support-vector-machine optimization problem, involving a latent variable model in terms of a high- dimensional representation of input data matrix, yields solutions containing information for encoding cluster assignment, and in turn for representing data matrix embedded in a lower-dimensional space. Furthermore, our formulation provides researchers on spectral, unsupervised pattern recognition methods with a fully matrix notation and formulation to easily understand kernel-based approaches such as KSC and KPCA.

5. References

Aldrich, C. and Auret, L., 2013. Statistical Learning Theory and Kernel-Based Methods. In *Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods*, pp. 117-181. Springer.

Alvarado-Pérez, J. C. and Peluffo-Ordóñez, D. H., 2015. Artificial and Natural Intelligence Integration. In *12th International Conference on Distributed Computing and Artificial Intelligence (DCAI 2016)*, p-p. 167-173. Springer.

Alvarado-Pérez, J. C., Peluffo-Ordóñez, D. H., and Therón, R., 2015. Bridging the gap between human knowledge and machine learning. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 4 (1):54-64.

Alzate, C. and Suykens, J. A. K., 2010. Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32 (2):335-347. Belanche Muñoz, L. A., 2013. Developments in kernel design. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013), Bruges, Belgium*, pp. 369-378.

Domeniconi, C., Peng, J., and Yan, B., 2011. Composite kernels for semi-supervised clustering. *Knowledge and information systems*, 28 (1):99-116.

Filippone, M., Camastra, F., Masulli, F., and Rovetta, S., 2008. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41 (1):176-190.

González, F., Bermeo, D., Ramos, L., and Nasraoui, O., 2012. On the Robustness of Kernel-Based Clustering. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 122-129.

Ham, J., Lee, D. D., Mika, S., and Schölkopf, B., 2004. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on Machine learning*, p. 47. ACM.

Harchaoui, Z., Bach, F., Cappé, O., and Moulines, E., 2013. Kernel-based methods for hypothesis testing: a unified view. *IEEE Signal Processing Magazine*, 30 (4):87-97.

Huang, H., Chuang, Y., and Chen, C., 2012. Multiple Kernel Fuzzy Clustering. *Fuzzy Systems, IEEE Transactions on*, 20 (1):120-134.

Langone, R., Alzate, C., and Suykens, J. A., 2013. Kernel spectral clustering with memory effect. *Physica A: Statistical Mechanics and its Applications*.

Molina-Giraldo, S., Álvarez-Meza, A., Peluffo-Ordoñez, D., and Castellanos-Domínguez, G., 2012. Image Segmentation Based on Multi-Kernel Learning and Feature Relevance Analysis. *Advances in Artificial Intelligence–IBERAMIA 2012*, pp. 501-510.

Peluffo-Ordóñez, D. H., Lee, J. A., Verleysen, M., Rodrıguez, J. L., and Castellanos-Domınguez, G., 2014. Unsupervised relevance analysis for feature extraction and selection. In *ICPRAM 2014*, pp. 310-315.

Peluffo-Ordónez, D., Garcia-Vega, S., Langone, R., Suykens, J., Castellanos-Dominguez, G. et al., 2013. Kernel spectral clustering for dynamic data using multiple kernel learning. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pp. 1-6. IEEE.

Peluffo-Ordonez, D. H., Aldo Lee, J., and Verleysen, M., 2014. Generalized kernel framework for unsupervised spectral methods of dimensionality reduction. In *2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pp. 171-177. IEEE.

Peluffo-Ordónez, D. H., Alvarado-Pérez, J. C., Lee, J. A., and Verleysen, M., 2015. Geometrical homotopy for data visualization. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2015)*.

Peluffo-Ordóñez, D. H., Alzate, C., Suykens, J. A., and Castellanos-Domínguez, G., 2014a. Optimal Data Projection for Kernel Spectral Clustering. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 553-558.

Peluffo-Ordóñez, D. H., Lee, J. A., and Verleysen, M., 2014b. Recent methods for dimensionality reduction: A brief comparative analysis. In *European Symposium on Artificial Neural Networks (ESANN)*. Citeseer.

Peluffo Ordoñez, D. H., Lee, J. A., Verleysen, M., Rodriguez, J. L., Castellanos-Dominguez, G. et al., 2015. Unsupervised relevance analysis for feature extraction and selection. A distance-based approach for feature relevance. In *3rd International Conference on Pattern Recognition Applications and Methods (ICPRAM 2014)*.

Schölkopf, B. and Smola, A. J., 2002. *Learning with Kernels*.

Seeland, M., Karwath, A., and Kramer, S., 2012. A structural cluster kernel for learning on graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 516-524. ACM.

Wolf, L. and Bileschi, S., 2005. Combining variable selection with dimensionality reduction. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pp. 801--806. IEEE.

Wolf, L. and Shashua, A., 2005. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *Journal of machine learning*, 6:1855-1887.

Wu, Y., Ma, W., Gong, M., Li, H., and Jiao, L., 2015. Novel Fuzzy Active Contour Model With Kernel Metric for Image Segmentation. *Applied Soft Computing*.

Zelnik-manor, L. and Perona, P., 2004. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*, pp. 1601-1608. MIT Press.