Noname manuscript No.
(will be inserted by the editor)

A Novel Dimensionality Reduction Technique based on Kernel Optimization Through Graph Embedding

N. Vretos, A. Tefas and I. Pitas

the date of receipt and acceptance should be inserted later

Abstract In this paper, we propose a new method for kernel optimization in kernel based dimensionality reduction techniques such as Kernel Principal Components Analysis (KPCA) and Kernel Discriminant Analysis (KDA). The main idea is to use the graph embedding framework for these techniques and, therefore, by formulating a new minimization problem to simultaneously optimize the kernel parameters and the projection vectors of the chosen dimensionality reduction method. Experimental results are conducted in various data sets, varying from real world publicly available databases for classification benchmarking to facial expressions and face recognition databases. Our proposed method outperforms other competing ones in classification performance. Moreover, our method provides a systematic way to deal with kernel parameters whose calculation was treated rather superficially so far and/or experimentally, in most of the cases.

1 Introduction

Dimensionality reduction techniques try to reduce the data dimensionality in a way that, in the reduced space, the data are better separated than in the original space. Dimensionality reduction techniques have attracted much attention in computer vision as well as in pattern classification, due to their implementation simplicity and classification performance. The most used linear dimensionality reduction techniques are Principal Component Analysis (PCA) [33] and Linear Discriminant Analysis [13]. Other methods exist as well, such as the Locality Preserving Projection (LPP) [25] and others. A good review of dimensionality reduction methods can be found in [31]. Most of these methods have their kernel counterparts. The well known kernel trick [24] can be applied to most of them in order to create their nonlinear version. In [28], [23] a kernel version of the original PCA (KPCA) and the kernel discriminant analysis (KDA) was proposed respectively. Moreover, other kernel methods exist like ISOMAP [1], Laplacian Eigenmaps [3] and others.

Kernel optimization and kernel learning refer to techniques that try either to learn a kernel matrix from the training data or to optimize the kernel function based on a task-dependent criterion. There is a vast amount of research works for kernel learning and kernel optimization. Most of the proposed methods try to create a new kernel, either as a combination of data dependent kernels, as in [34] and maximization of the so called kernel alignment criterion [4], or by means of data-dependent criteria, such as the Fischer ratio [13], which optimizes the class separability of the data. In the first case, Cristiannini *et al* [11] have proposed a method for kernel optimization, based on kernel alignment maximization, towards a base kernel on the labels of the data. In [18] Lackriet *et al* proposed a semidefinite programming problem (SDP), to optimize a linear combination of kernels under their positive semidefinite assumption on these kernels. To find the kernels coefficients, the support vector machine margin was minimized using an SDP with a constant trace constraint.

In the kernel function optimization case, the kernel function (i.e., the kernel hyper parameters) is optimized by means of a specific criterion. The most popular crite-

N. Vretos, A. Tefas and I. Pitas
Department of Informatics, University of Thessaloniki
Thessaloniki 54124, Greece Tel,Fax: +30-2310996304
E-mail: vretos@iti.gr E-mail: {tefas, pitas}@aiia.csd.auth.gr
The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 248434 (MOBISERV)

rion is the Fisher ration. In [8], the authors proposed to minimize the so-called J_4 criterion, which, in the case of Gaussian kernels, provides an analytical expression to find the scale parameter, as proven in [32]. Another method exist that tries to optimize the same criterion in order to find the optimal parameters for a specific kernel function has been proposed in [20].

In this paper, we propose a novel technique for optimizing dimensionality reduction criteria that depend on the selection of the appropriate kernel. The main novelty of this paper is the proposal of a systematic way to create new objective functions for kernel optimization, depending on the specific dimensionality reduction approach. Therefore, the objective functions to be minimized are variants of the matrix condition number, whose minimization can be made through solvable semidefinite programming (SDP) problems. Moreover, we give evidence that, if the kernel function satisfies Mercer conditions [24], the solution of the SDP can be extended to the hyper parameters of the kernel function. In this way, we achieve a generalization of kernel optimization, both in the sense of kernel matrix optimization as well as of kernel function optimization (i.e., kernel hyper parameters optimization).

Our approach is based on the graph embedding framework, firstly proposed in [35], which is a unified framework for all dimensionality reduction techniques such as PCA, LDA, LLE and others. Moreover, means were provided to easily transform the initial linear problems to their corresponding kernel and tensor versions. In this work, we provide the means to go a step beyond in the kernel version of dimensionality reduction methods, in such a way that we shall incorporate the kernel matrix (and, hence, the kernel hyper parameters) in the optimization problem and solve it with respect to both optimal embedding vectors and hyper-parameters. In [35], only the kernel versions of PCA and LDA are formulated and solved, thus, we shall investigate these two techniques. Our approach, does not handle cases where the kernel is embedded in the graph weights as will be detailed later on.

Our main novel idea is that, in the proper mathematical framework, the graph embedding optimization can be transformed to equivalent optimization problems, whose objective functions involve the matrix condition number. Moreover, since graph embedding provides a framework for specific dimensionality reduction techniques (based on the use of an intrinsic and a penalty graph), our method inherits this property and thus the new optimization function is specific for the dimensionality reduction technique described from the graph embedding problem. As we shall see later on, in order to connect such problems we need to make use of the congruence relation of matrices which is apparent in the optimization problems, defined in the kernel version of the graph embedding framework for KPCA and KDA.

The paper is organized as follows: in Section 2, the solution of the optimization problem of dimensionality reduction in the graph embedding framework is proven and a way is presented to extend it to the calculation of the optimal kernel hyper-parameters. In Section 3, we present the semidefinite programing problem (SDP) we use to solve the proposed optimization problem. Dimensionality reduction techniques in the new optimized kernel framework are presented in Section 4. In Section 5, we demonstrate results of the use of this dimensionality reduction in classification problems. Finally, conclusions are drawn and future work is discussed, in Section 6.

2 Kernel Optimization in Graph Embedding

Graph embedding is defined as the algorithm to find an optimal low-dimensional vector, representing the relations among the vertices of a similarity graph G [35]. In the simple one-dimensional case, the graph embedding optimization problem can be formulated as follows. Let $\mathbf{x}_i \in \mathbb{R}^M$, $i = 1, .., N$ be vectorial data (e.g. signals, images etc). Their similarities can be described by a similarity matrix $\mathbf{W}$ or a similarity graph $G = \{V, E\}$, where V the set of graph vertices each representing a vector $\mathbf{x}_i$ and E is the set of edges each representing the similarity of a pair of vertices (v_i, v_j). Various similarity measures have been used so far in order to calculate $\mathbf{W}$ depending on the information $\mathbf{x}_i$ stored in each graph vertex [10]. In spectral graph analysis the Laplacian Matrix $\mathbf{L}$ of a graph is defined as $L_{ij} = D_{ii} - W_{ij}$, where $D_{ii} = \sum_{j=1}^{N} W_{ij}$ and N is the number of vertices in G. Based on these definitions, the optimization problem for graph embedding can be formulated as follows $\arg\min_{\mathbf{y}^T\mathbf{y}=d} \mathbf{y}^T\mathbf{L}\mathbf{y}$, where d a scaling factor and $\mathbf{y}$ the low-dimensional representation vector of the vertices of G, such that, each vector $\mathbf{x}_i$ associated with vertex v_i, has y_i as its one-dimensional representation. It is easy to verify that the solution to the above optimization problem is given by $\mathbf{L}\mathbf{y}^* = \lambda\mathbf{y}^*$. The trivial solution $\mathbf{y}^* = \mathbf{1}$, for the eigenvalue $\lambda = 0$ is generally omitted [35]. In our case, we use the kernel version of the before mentioned problem. Let $\phi : \mathbb{R}^M \to \mathcal{F}$ be a function mapping the data to a high, possibly infinitely, dimensional space $\mathcal{F}$ and $\mathbf{K}$ be the kernel Gram matrix $K_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$. Under the assumption that the embedding direction can be written as $\mathbf{w} = \sum_i^N \alpha_i\phi(\mathbf{x}_i)$, the optimization problem can be written as:

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}=d} \boldsymbol{\alpha}^T\mathbf{K}^T\mathbf{L}\mathbf{K}\boldsymbol{\alpha}, \quad (1)$$

In our case, as we want to incorporate the kernel hyper parameters in the process, we assume that $\mathbf{K}$ also depends on a matrix $\boldsymbol{\Sigma}$ with $\boldsymbol{\Sigma}_{ij} = \sigma_{ij}$. Therefore the optimization problem becomes:

$$\boldsymbol{\alpha}^* = \arg \min_{\boldsymbol{\alpha}^T \mathbf{K}(\boldsymbol{\Sigma})\boldsymbol{\alpha}=d} \boldsymbol{\alpha}^T \mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{L} \mathbf{K}(\boldsymbol{\Sigma})\boldsymbol{\alpha}. \quad (2)$$

Alternatively, we can define a more complex optimization problem, where the orthogonality of the optimal vectors $\boldsymbol{\alpha}$ is conditioned by a matrix $\mathbf{B}$

$$\boldsymbol{\alpha}^* = \arg \min_{\boldsymbol{\alpha}^T \mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{B} \mathbf{K}(\boldsymbol{\Sigma})\boldsymbol{\alpha}=d} \boldsymbol{\alpha}^T \mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{L} \mathbf{K}(\boldsymbol{\Sigma})\boldsymbol{\alpha}. \quad (3)$$

It is well known that the above optimization problem is simply the generalized eigenvalue problem:

$$\mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{L} \mathbf{K}(\boldsymbol{\Sigma})\mathbf{v} = \lambda \mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{B} \mathbf{K}(\boldsymbol{\Sigma})\mathbf{v}. \quad (4)$$

The solution of this problem is given by the eigenvectors of:

$$\left(\mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{B} \mathbf{K}(\boldsymbol{\Sigma})\right)^{-1} \mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{L} \mathbf{K}(\boldsymbol{\Sigma}). \quad (5)$$

In the case of the graph embedding framework for kernel dimensionality reduction, we are interested in solving (3) as in [35]. The optimal $\boldsymbol{\alpha}^*$ is equal to the eigenvector corresponding to the smallest eigenvalue of the generalized eigenvalue problem. The solution of (4) is equivalent to minimizing the Generalized Reyleigh quotient :

$$r(\boldsymbol{\alpha}, \boldsymbol{\Sigma}) = \frac{\boldsymbol{\alpha}^T \mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{L} \mathbf{K}(\boldsymbol{\Sigma})\boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{B} \mathbf{K}(\boldsymbol{\Sigma})\boldsymbol{\alpha}}, \quad (6)$$

respectively. We shall see that this problem can be transformed to an equivalent problem of the matrix condition number κ of a symmetric matrix $\mathbf{A}$, which is the absolute ratio of the minimum and maximum eigenvalue $\kappa(\mathbf{A}) = \left|\frac{\lambda_{max}(\mathbf{A})}{\lambda_{min}(\mathbf{A})}\right|$, as long as $\lambda_{min}(\mathbf{A}) \neq 0$.

Theorem 1 *The problem (3) is equivalent to:*

$$\arg\min_{\boldsymbol{\Sigma}} \kappa(\mathbf{K}(\boldsymbol{\Sigma})\mathbf{K}(\boldsymbol{\Sigma})^T), \quad (7)$$

respectively.

Proof. Before we structure the proof of Theorem 1, we state a useful theorem on the congruent matrices. Congruent matrices, are matrices that belong to the same equivalence class under a similarity transformation [22]. That is, for arbitrary matrices $\mathbf{A}$, $\mathbf{B}$ and an invertible matrix $\mathbf{P}$, $\mathbf{A}$ is congruent to $\mathbf{B}$, if $\mathbf{B} = \mathbf{P^T A P}$. Ostrowski theorem [30] states that, under the congruence relation the following relation holds:

$$\lambda_k(\mathbf{P}^T \mathbf{A} \mathbf{P}) = \theta_k \cdot \lambda(\mathbf{A}) \quad (8)$$

$$\lambda_{min}(\mathbf{P}\mathbf{P}^T) \leq \theta_k \leq \lambda_{max}(\mathbf{P}\mathbf{P}^T) \quad (9)$$

All the above, in addition to the Rayleigh quotient, suggest that the optimization problem (3) can be transformed into an equivalent matrix condition number optimization problem in the following way. First, we consider bounds on the Rayleigh quotient (6). Matrix $\mathbf{K}$ is symmetric and positive semi definite, since it is a kernel matrix under the Mercer conditions. The Laplacian matrix $\mathbf{L}$ of a graph is symmetric and positive semi-definite as well. Therefore, we can write:

$$r(\boldsymbol{\alpha}, \boldsymbol{\Sigma}) \leq \frac{\lambda_{max}(\mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{L} \mathbf{K}(\boldsymbol{\Sigma}))}{\lambda_{min}(\mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{B} \mathbf{K}(\boldsymbol{\Sigma}))} \quad (10)$$

The above is straight forward if one considers the Rayleigh quotient for the nominator and denominator separately. Therefore, if we apply the Ostrowski Theorem to the above equations we get:

$$r(\boldsymbol{\alpha}, \boldsymbol{\Sigma}) \leq \frac{\lambda_{max}(\mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{L} \mathbf{K}(\boldsymbol{\Sigma}))}{\lambda_{min}(\mathbf{K}(\boldsymbol{\Sigma})^T \mathbf{B} \mathbf{K}(\boldsymbol{\Sigma}))} = \\ = \frac{\theta_{max} \cdot \lambda_{max}(\mathbf{L})}{\theta_{min} \cdot \lambda_{min}(\mathbf{B})} \quad (11)$$

Finally, based on the inequality of the Ostrowski theorem, the fact that $\mathbf{K}(\boldsymbol{\Sigma})$ is symmetric and the definition of the matrix condition number we can write:

$$r(\boldsymbol{\alpha}, \boldsymbol{\Sigma}) \leq \kappa(\mathbf{K}(\boldsymbol{\Sigma})\mathbf{K}(\boldsymbol{\Sigma})^T) \cdot \\ \cdot \frac{\lambda_{max}(\mathbf{L})}{\lambda_{min}(\mathbf{B})} \quad (12)$$

Thus, the problem in (3) become:

$$P : \arg\min_{\boldsymbol{\Sigma}} \kappa(\mathbf{K}(\boldsymbol{\Sigma})\mathbf{K}(\boldsymbol{\Sigma})^T) \cdot \\ \cdot \frac{\lambda_{max}(\mathbf{L})}{\lambda_{min}(\mathbf{B})} \quad (13)$$

□

Since P is equivalent to (3), they share the same optimal solutions. Moreover, since P is only conditioned on $\boldsymbol{\Sigma}$, the solution for 1 $\boldsymbol{\alpha}$ can be found if one substitutes the optimal $\boldsymbol{\Sigma}$ to the generalized eigenvalue problems in (4). It is easy to prove that methods like the one in [20] can be adapted to the proposed framework.

3 Matrix Condition Number Optimization and Kernel Hyperparameters

There is a vast amount of research works on matrix condition number optimization [9]. In our case, since $\mathbf{K}(\boldsymbol{\Sigma})$ is conditioned with a variable we need to prove that the solution of the matrix condition number minimization may be extended to the matrix variable $\boldsymbol{\Sigma}$. For simplicity of the demonstration we set $\mathbf{A} = \mathbf{K}(\boldsymbol{\Sigma})$. If we consider a space of solutions $\Omega = \{\mathbf{Q} \in \mathcal{S}^n | \, |\mathbf{Q} - \mathbf{A}| < \eta\}$

the minimization problem of $\kappa(\mathbf{A})$ has a solution for a specific $\mathbf{A}^* \in \Omega$ where $\mathcal{S}^n$ the space of real symmetric matrices under the following semidefinitie programming problem (SDP).

$$\begin{aligned} &\min_{s,t,\mathbf{A}} \ s \\ &\text{subject to} \ \ (\ Q_{ij} - \eta)t \leq A_{ij} \leq (Q_{ij} + \eta)t \\ &\qquad t \geq 0, \eta > 0, \mathbf{I}_{n\times n} \preceq \mathbf{A} \preceq s \cdot \mathbf{I}_{n\times n}, \end{aligned} \tag{14}$$

for a given matrix $\mathbf{Q}$ and an approximation variable η, where $\mathbf{A} \preceq \mathbf{B}$ means that $\mathbf{B} - \mathbf{A}$ is positive semidefinite. Although, in our case, this optimization problem fits well, other SDP optimization problems, which provide a relaxed definition of the solution space can be used. For instance, in [21], they propose that $\Omega = conv\{\mathbf{Q_1}, \mathbf{Q_2}, .., \mathbf{Q_m}\}$ where $conv$ the convex hull of these points in $\mathcal{S}^n$ and $\mathbf{y} \in \mathcal{R}^m$. In this case, the optimization problem can be written as:

$$\begin{aligned} &\min_{s,\mathbf{A}} \quad s \\ &\text{subject to} \ \sum_{i=1}^{m} \mathbf{Q}_i y_i = \mathbf{A} \\ &\qquad \mathbf{I}_{n\times n} \preceq \mathbf{A} \preceq s \cdot \mathbf{I}_{n\times n}. \end{aligned} \tag{15}$$

This approach is better suited in cases, where we can assume that the optimal solution for kernel optimization is a linear combination of several kernels [19]. The handicap of this approach is that the optimization becomes intractable for even medium size problems (i.e., for kernel matrices corresponding to a few hundred data samples).

As for the kernel hyper parameters, we must verify that (14) can provide a solution for $\boldsymbol{\Sigma}$ as well. To do so we need to prove that the solution of (14) can be extended to $\boldsymbol{\Sigma}$. Let us define the function $f : \mathcal{T} \to \mathbb{R}$ as $f(\boldsymbol{\Sigma}) = \kappa(\mathbf{K}(\boldsymbol{\Sigma}))$ where $\mathcal{T} \subset \mathbb{R}^l$ with $l = 1$ for the case of one σ and $l > 1$ for the other case.

Theorem 2 *for f as defined above, the optimization problem:*

$$\begin{aligned} &\textit{minimize } f(\boldsymbol{\Sigma}) && (16) \\ &\textit{subject to } \boldsymbol{\Sigma} \in \mathcal{T}, && (17) \end{aligned}$$

has a solution in $\mathcal{T}$.

Proof. Since $\lambda_{max}(\mathbf{K})$ is a convex function of $\mathbf{K}$ and $\lambda_{min}(\mathbf{K})$ is concave function of $\mathbf{K}$, both are Lipschitz continuous functions of $\mathbf{K}$. The continuous differentiability of $\mathbf{K}(\boldsymbol{\Sigma})$ implies that $\lambda_{max}(\mathbf{K}(\boldsymbol{\Sigma}))$ and $\lambda_{min}(\mathbf{K}(\boldsymbol{\Sigma}))$ are Lipschitz continuous functions on $\boldsymbol{\Sigma}$. Moreover, if the matrix $\mathbf{K}(\boldsymbol{\Sigma})$ has rank $rank(\mathbf{K}(\boldsymbol{\Sigma})) = N$, we can find positive constants M and m, such that:

$$m \leq \lambda_{min}(\mathbf{K}(\boldsymbol{\Sigma})) \text{ and } \lambda_{max}(\mathbf{K}(\boldsymbol{\Sigma})) \leq M, \tag{18}$$

Hence f is Lipschitz continuous and satisfies:

$$1 \leq f(\boldsymbol{\Sigma}) \leq \frac{M}{m}, \ \forall \boldsymbol{\Sigma} \in \mathcal{T}, \tag{19}$$

The above suggest that (16) has a solution in $\mathcal{T}$. □

All the above suggest that if $\mathbf{A}^*$ is a solution of (14), then we can find $\boldsymbol{\Sigma}^*$ after solving the system of nonlinear equations:

$$\mathbf{K}(\boldsymbol{\Sigma}^*)\mathbf{K}(\boldsymbol{\Sigma}^*)^T = \mathbf{A}^*. \tag{20}$$

Although the above analysis guarantees that such a $\boldsymbol{\Sigma}^*$ exists, finding it can be a difficult task.

Depending on the kernel we choose, as well as on the space of matrices $\boldsymbol{\Sigma}^*$, the above equations can be solved either analytically or numerically. In what follows, we shall provide a methodology for (20). The matrix $\boldsymbol{\Sigma}$ may have several representations, depending on the problem under consideration. We assume that $\boldsymbol{\Sigma}$ contains variables controlling the influence region (i.e. the spread) of a data point, as is the scale parameter σ in the Gaussian distribution. If, for instance we assume an identical spread for all data points, the matrix $\boldsymbol{\Sigma}$ takes the form $\boldsymbol{\Sigma} = \sigma \cdot \mathbf{1}_{N\times N}$ where $\mathbf{1}_{N\times N}$ the all ones $N \times N$ matrix and σ the common spread. On the other hand, if we assume that each data point has a different scale parameter then $\boldsymbol{\Sigma} = \boldsymbol{\sigma}\boldsymbol{\sigma}^T$ where $\boldsymbol{\sigma}$ the vector whose each element σ_i is the spread parameter associated with each sample i. Finally, in the more general case we can assume that each interconnection between points has a different spread parameter: $[\boldsymbol{\Sigma}]_{ij} = \sigma_{ij}$. Finally, we may consider other interpretations of the matrix $\boldsymbol{\Sigma}$ depending on the problem. It would be beneficial for instance under a class driven problem to consider $\boldsymbol{\Sigma}$ block diagonal matrix in the following form:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & 0 & ... & 0 \\ 0 & \boldsymbol{\Sigma}_2 & ... & 0 \\ ..... & ..... & ..... & ..... \\ 0 & 0 & ... & \boldsymbol{\Sigma}_{N_c} \end{bmatrix} \tag{21}$$

where N_c the number of classes in the problem and each block $\boldsymbol{\Sigma}_i$ for $i = 1, .., N_c$ is of the form $\boldsymbol{\Sigma} = \sigma_i \cdot \mathbf{1}_{N_i\times N_i}$ with σ_i a common spread within each class i and N_i the cardinality of each class.

3.1 Different solutions for Different Variable σ cases

In the single variable σ case, the matrix $\boldsymbol{\Sigma}$ can be considered as a single variable σ and thus the equation (20) takes the following form $\mathbf{K}(\sigma)\mathbf{K}(\sigma)^T = \mathbf{A}^*$. We define the kernel function $\phi(\sigma) = \exp(-\frac{w_{ij}}{2\sigma^2})$, where w_{ij} is the

Euclidean distance between 2 samples $\mathbf{x}_i$, $\mathbf{x}_j$. Thus, we must solve $\phi(\sigma) = \sqrt{A^*_{ij}}$ for every $i = 1, .., n$, $j = i, .., n$ since $\mathbf{K}(\sigma)$ and $\mathbf{A}^*$ are symmetric. Obviously, this system does not possess a solution if for any different i, j, k, $l \in [1, .., n]$, $A^*_{ij} \neq A^*_{kl}$ and $w_{ij} \neq w_{kl}$. However, we may approximate the solution using any of the well known approximation algorithms for solving such systems, such as the Levenberg-Marquardt algorithm.

In the case of one σ_i per sample we define the vector-valued functions $\phi_i(\boldsymbol{\sigma})$ as $\phi_{ij}(\boldsymbol{\sigma}) = \exp(-\frac{w_{ij}}{2\sigma_i\sigma_j})$. As previously we must solve the equation $\exp(-\frac{w_{ij}}{2\sigma_i\sigma_j}) = \sqrt{A^*_{ij}}$. Thid system of equations is overdetermined, since we have $\frac{N^2}{2} - N$ equations for N unknowns. As before, we can find an approximate solution.

Finally, in the case where $\Sigma_{ij} = \sigma_{ij}$ for each interconnection we have a different value σ_{ij} and we define the function $\phi(\boldsymbol{\Sigma})$ as $\phi_{ij}(\boldsymbol{\Sigma}) = \exp(-\frac{w_{ij}}{\sigma^2_{ij}})$. This system of equations can be solved analytically, since we need to find one σ_{ij} for each interconnection. Unfortunately, this approach does not have a practical use in the case of dimensionality reduction problems as we shall see in the next Section.

4 Kernel Optimization in Dimensionality Reduction

As briefly mentioned before, the graph embedding framework can be used to provide solutions to dimensionality reduction techniques. In [35], the authors have proven that most of the well known dimensionality reduction techniques, not only, fit in to the graph embedding framework, but also they can be extended in view of different graph choices that are employed in the process. The main idea behind dimensionality reduction within the graph embedding framework is to create two graphs, the intrinsic and the penalty. A list of graphs and the corresponding dimensionality reduction techniques can be found in [35].

In our case, we can apply kernel optimization technique described above in all methods which use the kernel version of the graph embedding framework. Thus, we can optimize the kernel hyperparameters in various dimensionality reduction techniques. To the best of authors knowledge, such a generalized technique for kernel optimization does not exist. Most of the techniques for kernel optimization are either supervised (learning the kernel from the data) or are based on the well known Fisher discrimination ratio [27]. As for the later, it is much closer to our work since it is a a special case of the proposed framework.

4.1 Kernel Principal Components Analysis (KPCA)

KPCA is a well know dimensionality reduction technique [29]. In the graph embedding framework, the intrinsic graph for PCA is the N-clique graph with equal and normalized weights $\frac{1}{N}$, while the penalty graph is defined as the non-edge N graph. These two graphs correspond to the weight matrices $\mathbf{W}$ and $\mathbf{B}$ respectively, where $W_{ij} = \frac{1}{N}$ and $\mathbf{B} = \mathbf{I}_{N\times N}$. Following the same definitions as previously for the Laplacian matrix of a graph, KPCA corresponds to the optimization problem (13). Thus, in the case of KPCA the optimal kernel may be found by minimizing the kernel condition number. One might expect that, in the KPCA case, the optimized scale parameter should be trivial. Although this is true, in the optimization problem in (14) there is a non-singularity constraint inherited from Ostrowski theorem, which makes the problem bounded on the matrix rank. Thus the scale parameter, must not exceed a certain value to degenerate the matrix rank. Moreover, in the case of Gaussian kernels the matrix is always full rank [27]. This strongly supports the applicability of the Ostrowski theorem in the case of KPCA.

4.2 Kernel Discriminant Analysis KDA

In the case of KDA, the intrinsic graph is the one where all vertices of the same class are connected with an equal weight $\frac{1}{N_c}$ where N_c is the number of samples in class c and the penalty graph is the intrinsic graph used in PCA. In this case, the similarity matrix $\mathbf{W}$ is block diagonal matrix, where each block represents one class. In the case of KDA the minimization of the $\mathbf{K}(\boldsymbol{\Sigma})\mathbf{K}(\boldsymbol{\Sigma})^T$ matrix condition number must be performed, using the SDP in (14), a solution can be found. Moreover, since matrix K($\boldsymbol{\Sigma}$) is symmetric, it is easy to calculate the optimized kernel.

4.3 Out of sample extension

When a new test sample arrives we need to test it against the database, we project the test sample to the space formed by the new kernel. To do so, we need to calculate first the projection matrix. The matrix $\mathbf{Q}$ in (14) is invertible and thus the projection matrix can be approximated by a linear transform. Let $\mathbf{Z}$ be the projection matrix taking Q to the optimal $\mathbf{A}^*$ then $\mathbf{Z} = \mathbf{Q}^{-1}\mathbf{A}^*$. Therefore, it is easy to transform the calculated test kernel (in the $\mathbf{Q}$ sense) and therefore apply the transformation to take this kernel to the optimal space. Thus, the test kernel is projected through $\mathbf{Z}$ before testing.

5 Experimental Results

We have performed several experiments in order to provide evidence that the proposed method performs ef-

Table 1 classification accuracies of the UCI database for simple KPCA, KDA and optimized kernel matrix versions.

	KPCA	KDA	optimized KPCA	optimized KDA
iris	77.3%	82.2%	79.6%	84.7%
wine	86.3%	91.2%	90.2%	94%
heart disease	37.8%	51.3%	49.3%	61.4%

ficiently in several image analysis tasks. The dimensionality reduction is tested in classification tasks including general classification problems, facial expression and face recognition. We have used 2 databases for face recognition (Yale [2] and ORL [26]), 1 for facial expression recognition (Cohn-Kanade [17]) and several dataset from the University of California Irvin repository [12] for data classification. All the optimization problems described above have been solved using the `CVX` optimization package for Matlab [15,14], which is very efficient in the case of SDP problems. Moreover, in all our experiments, we have used Gaussian kernels although any other Mercer kernel can be used as explained earlier.

In all our test, we compare our approach to KDA and KPCA with heuristically calculated spread parameters. The heuristics used are the ones mentioned in [35]. Moreover, for the classification task, we have used the k-nn classifier.

5.1 UCI Repository

We have used 3 different dataset from the UCI database: the iris database, the wine database and the heart database. The wine dataset consists of 178 samples of dimension 13 and 3 different classes of wines. The iris dataset has 150 samples of dimension 4 and 3 different classes. Finally, the heart dataset contains 303 samples of 13 attributes and 5 different classes interpreted as scale of severity for a heart disease(0/4 denoting no/sever disease, respectively). Although most of the experiments reported on the heart database focus on distinguishing the no disease from the other 4 disease classes, we conducted experiments for all five classes. A five fold approach is used to evaluate the performance of the classification based on the dimensionality reduction optimization technique. In the base KPCA and KDA algorithms, the parameter σ has been chosen based on known heuristics like the ones reported in [35]. Tables 1-3, depicts classification accuracy results on all chosen UCI databases.

As can be seen, the proposed method performs much better in the case of vector variable $\boldsymbol{\sigma}$ version of the optimization and has a poor improvement with respect to the simple KPCA and KDA, in the case of single variable. This can be explained, since in the vector variable case, we relax the constraint of the nonlinear equation system problem, by introducing more free variables, to the system. Thus, the optimization of $\boldsymbol{\sigma}$ converges better than for the single variable case. As we shall see, the same observation holds for the other datasets as well.

Table 2 classification accuracies of the UCI database for simple KPCA, KDA and optimized kernel function for single variable σ versions.

	KPCA	KDA	optimized KPCA	optimized KDA
iris	77.3%	82.2%	77.6%	82.8%
wine	86.3%	91.2%	87.4%	91.3%
heart disease	37.8%	51.3%	37.8%	51.6%

Table 3 classification accuracies of the UCI database for simple KPCA, KDA and optimized kernel function for vector variable $\boldsymbol{\sigma}$ versions.

	KPCA	KDA	optimized KPCA	optimized KDA
iris	77.3%	82.2%	80.1%	84.8%
wine	86.3%	91.2%	91.1%	94.2%
heart disease	37.8%	51.3%	49.5%	61.2%

Table 4 Mean classification accuracies for Facial expression recognition for simple KPCA, KDA and kernel matrix optimization (KMO)versions.

	KPCA	KDA	optimized KPCA	optimized KDA
KMO	49.1%	70.4 %	52.2%	72.4%
single σ	49.1%	70.4 %	49.5%	70.4%
vector $\boldsymbol{\sigma}$	49.1%	70.4 %	52.7%	73.2%
matrix $\boldsymbol{\Sigma}$	49.1%	70.4 %	52.3%	74%

5.2 Facial Expression Recognition

We have performed facial expression recognition on the Cohn-Kanade database [17], using the optimized kernel and a k-nn classifier. We have extracted 407 different facial expression images (with different number of entries in each facial expression class) from 100 different individuals. In this database, people vary in age, race and sex. We have conducted a person-out based inverse five fold cross validation. More specifically, we train with 20% of the individuals present in these databases and therefore we test the classifiers with the remaining 80% of the individuals. This is done mainly because our method requires time consuming optimization procedures and our aim is to provide results on kernel optimization rather than on facial expression.

The initial dimensionality of the data is 30000 (i.e., 150×200 pixels). We perform simple KPCA and KDA on the data and therefore we classify the different facial expressions with a k-nn classifier. The same approach is then applied with optimized KPCA and KDA. Results are depicted in Table 4. The percentages are the averages over the 5 folds. As can be seen, depending on the matrix variable choices, results varying as in the

Table 5 Face Recognition database accuracies for simple KPCA and KDA and optimized kernel matrix versions. Reported results are mean values of the classification accuracies of the k-nn classification algorithm over the 50 random subsets.

	KPCA (90%)	KDA	optimized KPCA	optimized KDA
Yale [2]	34.2%	78.8%	54.6%	79.4%
ORL [26]	36.8%	79.4%	57.2%	81.3%

Table 6 Face Recognition database accuracies for simple KPCA and KDA and optimized kernel matrix versions. Reported results are mean values of the classification accuracies of the k-nn classification algorithm over the 50 random subsets.

	KPCA (90%)	KDA	optimized KPCA	optimized KDA
Yale [2]	34.2%	78.8%	54.6%	79.4%
ORL [26]	36.8%	79.4%	57.2%	81.3%

case of the UCI dataset. Moreover, in this case, we have implemented the block diagonal matrix variable version. It can be seen, that although in the case of KDA, the recognition performance increases using the block version this is not the case for KPCA. This can be attributed to the fact that the block matrix version employs better the class information. Moreover, the small performance degradation with respect to the vector case may be explained in that, for KPCA, this block form stands between the single and vector variable case.

5.3 Face Recognition

Two databases where used for face recognition. The YALE database [2] contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised and wink. We have used 2 images per person for training and the rest for testing. The ORL [26] contains 10 different images for 40 distinct persons. For some subjects, the images were taken at different times, with varying illumination, facial expressions and props (glasses / no glasses). In this test, we have taken 2 images per person for training and the rest for testing. Moreover, we have used the 50 random subsets collected by Deng Cai *et al.* mentioned in [6]-[16]. We have used the $k-$nn algorithm for classification after the dimensionality reduction. In Tables 5-8, results for face recognition accuracies are illustrated.

Same as before results are slightly better in the vector version. In Figure 1, the face recognition accuracies for Yale in the kernel matrix optimization version of the algorithm are depicted.

Table 7 Face Recognition database accuracies for simple KPCA and KDA and optimized kernel function of single variable σ versions. Reported results are mean values of the classification accuracies of the k-nn classification algorithm over the 50 random subsets.

	KPCA (90%)	KDA	optimized KPCA	optimized KDA
Yale [2]	34.2%	78.8%	50.2%	79.4%
ORL [26]	36.8%	79.4%	53.2%	80.6%

Table 8 Face Recognition database accuracies for simple KPCA and KDA and optimized kernel function of vector variable $\boldsymbol{\sigma}$ versions. Reported results are mean values of the classification accuracies of the k-nn classification algorithm over the 50 random subsets.

	KPCA (90%)	KDA	optimized KPCA	optimized KDA
Yale [2]	34.2%	78.8%	54.6%	80%
ORL [26]	36.8%	79.4%	57.2%	81.6%

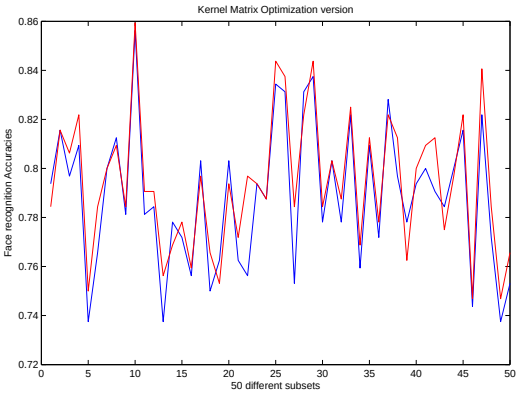

Fig. 1 Comparison of KDA and Optimal KDA. In red the optimal version, while in blue the original version of KDA with heuristic definition of the Gaussian scale.

6 Conclusions And Future Work

In this paper we propose a novel technique for kernel optimization. The proposed method is novel in two aspects. First we provide a framework through graph embedding for a general kernel optimization technique, with respect to any dimensionality reduction technique. Under the assumptions defined in this paper for the kernel matrix or the kernel function (i.e., the Mercer conditions) this framework may be applied to any dimensionality reduction technique that can be formulated as the kernel version of the graph embedding framework [35]. Moreover, as we have proven, the optimal solution of the minimization problem may be extended to the kernel function hyperparameters under strong mathematical assumptions. The main drawback of the proposed method is the use of highly expensive optimization procedures,which limits the applicability of the proposed method for data-bases of less than thousand of samples.

References

1. Balasubramanian, M., Schwartz, E.: The isomap algorithm and topological stability. Science **295**(5552), 7 (2002)
2. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. Pattern Analysis and Machine Intelligence, IEEE Transactions on **19**(7), 711–720 (1997)
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation **15**(6), 1373–1396 (2003)
4. Bousquet, O., Herrmann, D.: On the complexity of learning the kernel matrix. Advances in neural information processing systems pp. 415–422 (2003)
5. Cai, D., He, X., Han, J.: Spectral regression for efficient regularized subspace learning. In: Proc. Int. Conf. Computer Vision (ICCV'07) (2007)
6. Cai, D., He, X., Han, J., Zhang, H.J.: Orthogonal laplacianfaces for face recognition. IEEE Transactions on Image Processing **15**(11), 3608–3614 (2006)
7. Cai, D., He, X., Hu, Y., Han, J., Huang, T.: Learning a spatially smooth subspace for face recognition. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition Machine Learning (CVPR'07) (2007)
8. Chen, B., Liu, H., Bao, Z.: A kernel optimization method based on the localized kernel fisher criterion. Pattern Recognition **41**(3), 1098–1109 (2008)
9. Chen, X., Womersley, R., Ye, J.: Minimizing the condition number of a gram matrix. Preprint, The Department of Applied Mathematics, The Hong Kong Polytechnic University (2010)
10. Chung, F.: Spectral graph theory. 92. Amer Mathematical Society (1997)
11. Cristianini, N., Scholkopf, B.: Support vector machines and kernel methods: the new generation of learning machines. Ai Magazine **23**(3), 31 (2002)
12. Frank, A., Asuncion, A.: UCI machine learning repository (2010). URL http://archive.ics.uci.edu/ml
13. Fukunaga, K.: Introduction to statistical pattern recognition. Academic Pr (1990)
14. Grant, M., Boyd, S.: 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, pp. 95–110. Springer-Verlag Limited (2008)
15. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx (2011)
16. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.J.: Face recognition using laplacianfaces. IEEE Transactions on Pattern Anal. Mach. Intelligence **27**(3), 328–340 (2005)
17. Kanade, T., Cohn, J., Tian, Y.L.: Comprehensive database for facial expression analysis. In: Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), pp. 46 – 53 (2000)
18. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L., Jordan, M.: Learning the kernel matrix with semidefinite programming. The Journal of Machine Learning Research **5**, 27–72 (2004)
19. Lanckriet, G., Deng, M., Cristianini, N., Jordan, M., Noble, W., et al.: Kernel-based data fusion and its application to protein function prediction in yeast. In: Proceedings of the Pacific Symposium on Biocomputing, vol. 9, p. 2. World Scientific Singapore (2004)
20. Li, J., Pan, J., Lu, Z.: Kernel optimization-based discriminant analysis for face recognition. Neural computing & applications **18**(6), 603–612 (2009)
21. Lu, Z., Pong, T.: Minimizing condition number via convex programming. Tech. rep., Technical report, University of Washington, Seattle, WA, 2010. 8 (2010)
22. MacDuffee, C.: The theory of matrices. Dover Pubns (2004)
23. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.: Fisher discriminant analysis with kernels. In: Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop, pp. 41–48. IEEE (1999)
24. Muller, K., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.: An introduction to kernel-based learning algorithms. Neural Networks, IEEE Transactions on **12**(2), 181–201 (2001)
25. Niyogi, X.: Locality preserving projections. In: Advances in neural information processing systems 16: proceedings of the 2003 conference, vol. 16, p. 153. The MIT Press (2004)
26. Samaria, F., Harter, A.: Parameterisation of a stochastic model for human face identification. In: Proceedings of the Second IEEE Workshop on Applications of Computer Vision, pp. 138–142. IEEE (1994)
27. Schölkopf, B., Smola, A.: Learning with kernels: Support vector machines, regularization, optimization, and beyond. the MIT Press (2002)
28. Schölkopf, B., Smola, A., Müller, K.: Kernel principal component analysis. In: Proceeding of the Artificial Neural Networks, pp. 583–588. Springer (1997)
29. Schölkopf, B., Smola, A., Müller, K.: Kernel principal component analysis. Artificial Neural Networks?ICANN'97 pp. 583–588 (1997)
30. Varga, R.: Matrix iterative analysis, vol. 27. Springer (2010)
31. Wang, F., Sun, J.: Survey on distance metric learning and dimensionality reduction in data mining. Data Mining and Knowledge Discovery **29**(2), 534–564 (2014)
32. Wang, J., Lu, H., Plataniotis, K., Lu, J.: Gaussian kernel optimization for pattern classification. Pattern Recognition **42**(7), 1237–1247 (2009)
33. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemometrics and intelligent laboratory systems **2**(1-3), 37–52 (1987)
34. Xiong, H., Swamy, M., Ahmad, M.: Optimizing the kernel in the empirical feature space. IEEE Transactions on Neural Networks **16**(2), 460–474 (2005)
35. Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q., Lin, S.: Graph embedding and extensions: A general framework for dimensionality reduction. Pattern Analysis and Machine Intelligence, IEEE Transactions on **29**(1), 40–51 (2007)