2014 22nd International Conference on Pattern Recognition

Laplacian Support Vector Analysis for Subspace Discriminative Learning

Nikolaos Arvanitopoulos
School of Computer and Communication Sciences
EPFL, Switzerland

Dimitrios Bouzas
BETA CAE Systems S.A.

Anastasios Tefas
Department of Computer Science,
Aristotle University of Thessaloniki, Greece

***Abstract*—In this paper we propose a novel dimensionality reduction method that is based on successive Laplacian SVM projections in orthogonal deflated subspaces. The proposed method, called Laplacian Support Vector Analysis, produces projection vectors, which capture the discriminant information that lies in the subspace orthogonal to the standard Laplacian SVMs. We show that the optimal vectors on these deflated subspaces can be computed by successively training a standard SVM with specially designed deflation kernels. The resulting normal vectors contain discriminative information that can be used for feature extraction. In our analysis, we derive an explicit form for the deflation matrix of the mapped features in both the initial and the Hilbert space by using the kernel trick and thus, we can handle linear and non-linear deflation transformations. Experimental results in several benchmark datasets illustrate the strength of our proposed algorithm.**

I. Introduction

In many applications of pattern classification one has to deal with data of high dimensionality. However, many dimensions of the feature vectors are often highly correlated and the data can usually admit a low-dimensional representation. Furthermore, dealing with high-dimensional data has disadvantages, both in terms of complexity and classification performance. Due to the above reasons, the problem of *Dimensionality Reduction* has attracted much interest in the scientific community. Many algorithms, which handle both linear and non-linear projections, have been proposed so far. Among the most popular and successful linear ones are the *Principal Component Analysis (PCA)* [6], *Linear Discriminant Analysis (LDA)* [4] and *Locality Preserving Projections (LPP)* [5]. By taking advantage of the *kernel trick* [11], we can reformulate the above approaches as kernel algorithms. The corresponding algorithms are the *Kernel Principal Component Analysis (KPCA)* [10], *Kernel Discriminant Analysis* [9] and *Laplacian Eigenmaps (LE)* [2]. Another approach that has been proposed and uses successive projections using a standard SVM is the *Margin Maximizing Discriminant Analysis (MMDA)* and the corresponding *Kernel Margin Maximizing Discriminant Analysis (KMMDA)* [7], which applies the kernel trick on MMDA.

Manifold Regularization [1] was recently proposed as a general framework for learning from labeled and unlabeled data. A regularization term that takes into account the geometry of the data distribution has been proposed. For this purpose, the *graph-Laplacian* of a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ has been used, with vertex set the data points and similarity matrix $\mathbf{W}$. The whole framework has been developed in a *Reproducing Kernel Hilbert Space (RKHS)* setting and a new *Representer Theorem* has been proved. As a result of the above framework, novel algorithms, such as *Laplacian Regularized Least Squares (LapRLS)* and *Laplacian Support Vector Machines (LapSVM)* have been proposed.

Furthermore, in [12] a discriminative semi-supervised feature selection model was proposed based on Manifold Regularization. As in [1], an extended SVM formulation is proposed by using an additional regularization term based on the graph Laplacian. This problem formulation selects features that are most discriminative in terms of the classification margin and at the same time exploits the geometry of the data distribution that generates both labeled and unlabeled data.

In this paper, we propose a novel technique for dimensionality reduction that integrates the geometry of the data distribution into the optimization problem of a SVM. The proposed approach is inspired by the Laplacian SVMs in [1], where the manifold regularization term ensures smoothness along the underlying manifold of the initial space. The main idea is to use the discriminative information contained in the subspace (i.e., the hyperplane) that is orthogonal to the initial Laplacian SVM vector for successive feature extraction. The maximum margin formulation guarantees the discriminative ability of the additional projection vectors that are orthogonal to the successive hyperplanes. In order to extract the additional discriminative dimensions we use an iterative deflation procedure that allows us to compute projections using the deflated samples. The contribution of our work is summarized in the following:

1) We propose a novel Laplacian SVM formulation in deflated subspaces that incorporates knowledge of the geometry of the data based on manifold regularization and we extend it to the non-linear case.
2) We use the discriminative information of the constructed hyperplanes of the SVM optimization problems to successively generate orthogonal projection directions onto deflated subspaces for Dimensionality Reduction. We also extend the method to non-linear deflation transformations.
3) We incorporate the above deflation procedures, linear and non-linear, into the SVM formulations to obtain the final projection vectors.
4) We extend our approach to the multiclass case.

The manuscript is organized as follows. The proposed Laplacian Support Vector Analysis (LSVA) in deflated subspaces is described in detail in Section II. In Section III we extend our approach to address problems that include multiple data

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.285

classes. Experimental results in several benchmark datasets are given in Section IV. Finally, conclusions are drawn in Section VI.

II. Laplacian SVM Analysis

In this Section we introduce initially the linear deflation procedure in Subsection II-B and the corresponding Linear LSVA (LLSVA) formulation on deflated subspaces in Subsection II-C. We begin our analysis for the linear case in order to better present and clarify the proposed framework and then we proceed to the more general non-linear case in Subsections II-D and II-E respectively.

A. Notation

We assume that we have in total n data points assembled in a matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top \in \mathbb{R}^{n\times d}$ with label vector $\mathbf{y} = [y_1, \ldots, y_n]^\top \in \mathbb{R}^n$ and label set $\mathcal{Y} = \{-1, +1\}$. In the remaining we denote by $\boldsymbol{\mathcal{L}} = \mathbf{D} - \mathbf{W}$ the graph Laplacian that corresponds to a graph with nodes the data samples in $\mathbf{X}$, weight matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ and diagonal degree matrix $\mathbf{D}$ with entries $D_{ii} = \sum_{j=1}^n W_{ij}$. We denote by $\phi : \mathcal{X} \to \mathcal{H}$ the non-linear mapping from the initial data space to a RKHS and by $\mathbf{K}$ the corresponding Mercer kernel. The kernel function is denoted by $k(\mathbf{x}, \mathbf{y}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. With slight abuse of notation we use the same characters $\phi, \mathbf{K}, k(\cdot,\cdot)$ for different kinds of mappings, kernels and kernel functions throughout the paper. However, their forms will be clear from the context.

B. Linear Deflation Transformation

It is well-known that any SVM formulation proposed in the literature produces a projection vector that optimizes a specific objective function. Usually, this projection is the only one used for discriminality and the subspace that is orthogonal to this projection is discarded by the classification procedure. However, it is obvious that this orthogonal subspace (called hereafter deflated subspace) may contain also discriminant information for the given task. Thus, it would be desirable to have an algorithm that can successively extract all the projection vectors in these deflated subspaces. That is, in order to project our data into successive orthogonal hyperplanes we use a deflation transformation algorithm [7]. If $\widetilde{\mathbf{w}}_k = \mathbf{w}_k / \|\mathbf{w}_k\|$ is the orthonormal vector of the hyperplane in iteration k of the algorithm, the projection matrix that projects (i.e., deflates) the data samples on the corresponding hyperplane is given by $\mathbf{P}_{\widetilde{\mathbf{w}}_k} = \mathbf{I}_{d\times d} - \widetilde{\mathbf{w}}_k \widetilde{\mathbf{w}}_k^\top$, where $\mathbf{I}_{d\times d} \in \mathbb{R}^{d\times d}$ is the identity matrix. Now, the deflated data along the direction of $\widetilde{\mathbf{w}}_k$ are $\mathbf{X}_k = \mathbf{X}\mathbf{P}_{\widetilde{\mathbf{w}}_k}$. That is, in each iteration we project the data onto the subspace described by $\widetilde{\mathbf{w}}_k$. In order to obtain more discriminative projection directions, we proceed in an iterative manner: In each iteration k, we construct a new normal vector $\widetilde{\mathbf{w}}_k$, which is orthogonal to all the previously constructed vectors $\widetilde{\mathbf{w}}_j$, $j < k$. That is, $\langle \widetilde{\mathbf{w}}_k, \widetilde{\mathbf{w}}_j \rangle = 0$, $j < k$. The above statement can be justified by taking into acount the fact that $\widetilde{\mathbf{w}}_k, \widetilde{\mathbf{w}}_j$ belong to subspaces that are orthogonal to each other and consequently, it holds that $\widetilde{\mathbf{w}}_k, \widetilde{\mathbf{w}}_j$ are orthogonal $\forall k \neq j$. The multiple deflation can now be done in a successive way by projecting the data onto new subspaces, that is $\mathbf{X}_1 = \mathbf{X}\mathbf{P}_{\widetilde{\mathbf{w}}_1}, \ldots, \mathbf{X}_k = \mathbf{X}_{k-1}\mathbf{P}_{\widetilde{\mathbf{w}}_k}$. However, this iterative procedure is equivalent to directly applying the deflation matrix $\mathbf{P}_k = \mathbf{P}_{\widetilde{\mathbf{w}}_1} \cdots \mathbf{P}_{\widetilde{\mathbf{w}}_k}$ to the initial data $\mathbf{X}$ and thus there is no need to explicitly deflate all the data. From the orthogonality property of the normal vectors, the final projection matrix can also be written as $\mathbf{P}_k = \mathbf{I}_{d\times d} - \sum_{j=1}^k \widetilde{\mathbf{w}}_j \widetilde{\mathbf{w}}_j^\top$. Let us note here that the matrix $\mathbf{P}_k$ is symmetric and not invertible, since for a projection matrix it holds $\mathbf{P}_k^2 = \mathbf{P}_k$.

C. Linear Laplacian Support Vector Analysis

We are now in a position to propose a new LSVA formulation in deflated subspaces. In this new formulation, we integrate an additional regularization term which incorporates information about the geometry of the data distribution. This information is encoded in the graph Laplacian $\boldsymbol{\mathcal{L}}$ of a specific graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ with nodes the data points $\mathbf{X}$ and edges defined by a weight matrix $\mathbf{W}$. Furthermore, we assume that our data are deflated onto already computed subspaces from $k-1$ previous iterations of our algorithm, that is our current data have the form $\mathbf{X}_k = \mathbf{X}\mathbf{P}_{k-1}$. The final combined optimization criterion solved in each iteration k is

$$\min_{\mathbf{w}_k} \frac{1}{2}\mathbf{w}_k^\top \mathbf{w}_k + \frac{\lambda}{2}\mathbf{w}_k^\top \mathbf{P}_{k-1}\mathbf{X}^\top \boldsymbol{\mathcal{L}}\mathbf{X}\mathbf{P}_{k-1}\mathbf{w}_k + C\sum_{i=1}^n \xi_i, \quad (1)$$
$$\text{s.t. } y_i(\mathbf{w}_k^\top \mathbf{P}_{k-1}\mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i = 1, \ldots, n.$$

The solution to the above problem is given by the saddle point of the Lagrangian

$$L(\mathbf{w}_k, b, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}_k^\top (\mathbf{I} + \lambda \mathbf{P}_{k-1}\mathbf{X}^\top \boldsymbol{\mathcal{L}}\mathbf{X}\mathbf{P}_{k-1})\mathbf{w}_k +$$
$$C\sum_{i=1}^n \xi_i - \sum_{i=1}^n \gamma_i (y_i(\mathbf{w}_k^\top \mathbf{P}_{k-1}\mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^n \beta_i \xi_i.$$

Defining the matrix

$$\mathbf{A}_{k-1} = (\mathbf{I} + \lambda \mathbf{P}_{k-1}\mathbf{X}^\top \boldsymbol{\mathcal{L}}\mathbf{X}\mathbf{P}_{k-1}), \quad (2)$$

the modified Karush-Kuhn-Tucker (KKT) conditions are

$$\begin{aligned}
&\nabla_{\mathbf{w}_k} L(\mathbf{w}_{k,o}, b_o, \boldsymbol{\gamma}_o, \boldsymbol{\beta}_o, \boldsymbol{\xi}_o) = \mathbf{0} \\
&\Leftrightarrow \mathbf{A}_{k-1}\mathbf{w}_{k,o} = \sum_{i=1}^n \gamma_{i,o} y_i \mathbf{P}_{k-1}\mathbf{x}_i, \\
&\frac{\partial}{\partial b} L(\mathbf{w}_{k,o}, b_o, \boldsymbol{\gamma}_o, \boldsymbol{\beta}_o, \boldsymbol{\xi}_o) = 0 \Leftrightarrow \sum_{i=1}^n \gamma_{i,o} y_i = 0, \\
&\frac{\partial}{\partial \xi_i} L(\mathbf{w}_{k,o}, b_o, \boldsymbol{\gamma}_o, \boldsymbol{\beta}_o, \boldsymbol{\xi}_o) = 0 \Leftrightarrow \beta_{i,o} = C - \gamma_{i,o}, \\
&y_i(\mathbf{w}_{k,o}^\top \mathbf{P}_{k-1}\mathbf{x}_i + b_o) - 1 + \xi_i \geq 0, \\
&0 \leq \gamma_{i,o} \leq C,\ \beta_{i,o} \geq 0,\ \xi_i \geq 0,\ \beta_{i,o}\xi_{i,o} = 0,\ i = 1, \ldots, n, \\
&\gamma_{i,o}(y_i(\mathbf{w}_{k,o}^\top \mathbf{P}_{k-1}\mathbf{x}_i + b_o) - 1 + \xi_i) = 0, \quad i = 1, \ldots, n.
\end{aligned} \quad (3)$$

By using the KKT conditions the dual becomes

$$\max_{\boldsymbol{\gamma}} \sum_{i=1}^n \gamma_i - \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \gamma_i \gamma_j y_i y_j \mathbf{x}_i^\top \mathbf{P}_{k-1}\mathbf{A}_{k-1}^{-1}\mathbf{P}_{k-1}\mathbf{x}_j, \quad (4)$$
$$\text{s.t. } 0 \leq \gamma_i \leq C \text{ and } \sum_{i=1}^n \gamma_i y_i = 0.$$

The optimal weight vector is given by

$$\mathbf{w}_{k,o} = \mathbf{A}_{k-1}^{-1} \sum_{i=1}^n \gamma_{i,o} y_i \mathbf{P}_{k-1}\mathbf{x}_i, \quad (5)$$

and the corresponding separating hyperplane by

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w}_{k,o}^\top \mathbf{P}_{k-1}\mathbf{x} + b) = \text{sgn}(\sum_{i=1}^{n} \gamma_{i,o} y_i \mathbf{x}_i^\top \mathbf{P}_{k-1}\mathbf{A}_{k-1}^{-1}\mathbf{P}_{k-1}\mathbf{x} + b). \quad (6)$$

We can easily see that the above dual corresponds to a standard SVM formulation using the deflation kernel matrix

$$\boldsymbol{\mathcal{K}} = \mathbf{X}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^{-1}\mathbf{P}_{k-1}\mathbf{X}^\top. \quad (7)$$

The resulting vector $\mathbf{w}_k$ has to be normalized, that is $\widetilde{\mathbf{w}}_k = \mathbf{w}_k / \|\mathbf{w}_k\|$. The new projection matrix that adds an additional discriminative projection direction is given by $\mathbf{P}_k = \mathbf{P}_{k-1}(\mathbf{I}_{d\times d} - \widetilde{\mathbf{w}}_k\widetilde{\mathbf{w}}_k^\top)$. The above procedure is repeated until we reach the desired dimensionality d'. We have now obtained an orthogonal matrix $\widetilde{\mathbf{W}} \in \mathbb{R}^{n\times d'}$ whose columns contain the orthonormal vectors $\widetilde{\mathbf{w}}_j$, $j = 1, \ldots, d'$. These vectors provide discriminative information and can be used for feature extraction. Finally, the resulting projected data are given by $\mathbf{X}_{d'} = \mathbf{X}\widetilde{\mathbf{W}} \in \mathbb{R}^{n\times d'}$. The complete algorithm for linear projections is described in Algorithm 1.

Algorithm 1 Linear Laplacian Support Vector Analysis (LLSVA)

1: **Input**: Data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top \in \mathbb{R}^{n\times d}$ and label vector $\mathbf{y} = [y_i, \ldots, y_n]^\top \in \{-1, +1\}^n$. Graph Laplacian $\mathcal{L} \in \mathbb{R}^{n\times n}$. Target dimensionality d'.
2: $\mathbf{P}_0 = \mathbf{I}_{d\times d}$.
3: $\widetilde{\mathbf{W}} = []$.
4: **for** $k = 1$ **to** d' **do**
5: Compute the deflation kernel matrix $\boldsymbol{\mathcal{K}} = \mathbf{X}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^{-1}\mathbf{P}_{k-1}\mathbf{X}^\top$, with $\mathbf{A}_{k-1}$ as in (2).
6: Train a SVM using the above kernel.
7: Compute the normal vector $\mathbf{w}^k$ from (5) and normalize it as $\widetilde{\mathbf{w}}_k = \mathbf{w}_k / \|\mathbf{w}_k\|$.
8: $\widetilde{\mathbf{W}} = [\widetilde{\mathbf{W}}; \widetilde{\mathbf{w}}_k]$.
9: $\mathbf{P}_k = \mathbf{P}_{k-1}(\mathbf{I}_{d\times d} - \widetilde{\mathbf{w}}_k\widetilde{\mathbf{w}}_k^\top)$.
10: **end for**
11: Final projected data $\mathbf{X}_{d'} = \mathbf{X}\widetilde{\mathbf{W}}$.

D. Non-linear Deflation Transformation

We now extend our approach to handle non-linear projections. That is, we map the data to a RKHS in which we assume that they are linearly deflated. This deflation corresponds to a non-linear deflation in the original space. The new non-linear projection matrix along the direction of the mapped normalized vector $\widetilde{\mathbf{w}}_k^{(\phi)}$ can be written as $\mathbf{P}_{\widetilde{\mathbf{w}}^{(\phi)}} = \mathbf{I}_{m\times m} - \widetilde{\mathbf{w}}_k^{(\phi)}\widetilde{\mathbf{w}}_k^{(\phi)\top}$, where m is the unknown (and potentially infinite) dimensionality of the feature map ϕ. Analogously, the final matrix that contains all the orthonormal vectors is $\mathbf{P}_k^{(\phi)} = \mathbf{I}_{m\times m} - \sum_{j=1}^{k} \widetilde{\mathbf{w}}_j^{(\phi)}\widetilde{\mathbf{w}}_j^{(\phi)\top}$. The deflated data are $\boldsymbol{\Phi}_k = \boldsymbol{\Phi}\mathbf{P}_k^{(\phi)}$, where $\boldsymbol{\Phi}$ is the matrix of the mapped data points through the mapping ϕ. It is clear that $\mathbf{P}_k^{(\phi)}$ cannot be computed explicitly by using the above form. Therefore, we have to take advantage of the kernel trick in order to handle the unknown dimensionality of the feature map. To do so, we safely assume that the projection vector can be restricted to be in the range of $\boldsymbol{\Phi}_k$, since $\widetilde{\mathbf{w}}_k^{(\phi)} \in \mathbb{R}^m$, which is the column space of $\boldsymbol{\Phi}_k$. Therefore, we can write $\widetilde{\mathbf{w}}_k^{(\phi)} = \mathbf{P}_{k-1}^{(\phi)} \sum_{i=1}^{n} \widetilde{\alpha}_{k,i}\phi(\mathbf{x}_i) = \mathbf{P}_{k-1}^{(\phi)}\boldsymbol{\Phi}^\top\widetilde{\boldsymbol{\alpha}}_k$. The projection vector is now written as a linear combination of the already deflated data, with coefficients $\widetilde{\alpha}_{k,i} \in \mathbb{R}$, along the previous orthonormal directions in the feature space. By using this form for the projection vectors we guarantee that the normalized vector $\widetilde{\mathbf{w}}_k^{(\phi)} = \mathbf{w}_k^{(\phi)} / \left\|\mathbf{w}_k^{(\phi)}\right\|$ is orthogonal to the previously computed projection vectors, that is $\left\langle \widetilde{\mathbf{w}}_k^{(\phi)}, \widetilde{\mathbf{w}}_j^{(\phi)} \right\rangle = 0$, $j < k$. It is easy to show that, taking into account the orthogonality property, the projection matrix can be written as:

$$\mathbf{P}_k^{(\phi)} = \mathbf{I}_{m\times m} - \mathbf{P}_{k-1}^{(\phi)}\boldsymbol{\Phi}^\top(\sum_{j=1}^{k} \widetilde{\boldsymbol{\alpha}}_j\widetilde{\boldsymbol{\alpha}}_j^\top)\boldsymbol{\Phi}\mathbf{P}_{k-1}^{(\phi)}. \quad (8)$$

By multipling equation (8) from the left side with $\boldsymbol{\Phi}$ and from the right side with $\boldsymbol{\Phi}^\top$ we end to the following relation:

$$\mathbf{K}_k = \mathbf{K} - \mathbf{K}_{k-1}(\sum_{j=1}^{k-1} \widetilde{\boldsymbol{\alpha}}_j\widetilde{\boldsymbol{\alpha}}_j^\top)\mathbf{K}_{k-1}, \quad (9)$$

where $\mathbf{K}_k = \boldsymbol{\Phi}\mathbf{P}_k^{(\phi)}\boldsymbol{\Phi}^\top$ denotes the k-th deflated kernel. In equation (9) it is shown that the k-th deflated kernel of the data can be expressed as a subtraction between the kernel of the data in the feature space and a sum of all the previous deflated kernels. For the norm of the projection vector we have

$$\left\|\mathbf{w}_k^{(\phi)}\right\| = \sqrt{\mathbf{w}_k^{(\phi)\top}\mathbf{w}_k^{(\phi)}} = \sqrt{\boldsymbol{\alpha}_k^\top\boldsymbol{\Phi}\mathbf{P}_{k-1}^{(\phi)}\mathbf{P}_{k-1}^{(\phi)}\boldsymbol{\Phi}^\top\boldsymbol{\alpha}_k} = \sqrt{\boldsymbol{\alpha}_k^\top\boldsymbol{\Phi}\mathbf{P}_{k-1}^{(\phi)}\boldsymbol{\Phi}^\top\boldsymbol{\alpha}_k} = \sqrt{\boldsymbol{\alpha}_k^\top\mathbf{K}_{k-1}\boldsymbol{\alpha}_k}. \quad (10)$$

The new representation does not involve the explicitly mapped feature vectors, but only the corresponding deflated kernel. Now, for the normalized projection vector $\widetilde{\boldsymbol{\alpha}}_k$ it holds

$$\widetilde{\boldsymbol{\alpha}}_k = \frac{\boldsymbol{\alpha}_k}{\sqrt{\boldsymbol{\alpha}_k^\top\mathbf{K}_{k-1}\boldsymbol{\alpha}_k}}. \quad (11)$$

Finally, our regularization term becomes

$$\widetilde{\mathbf{w}}_k^{(\phi)\top}\mathbf{P}_{k-1}^{(\phi)}\boldsymbol{\Phi}^\top\boldsymbol{\mathcal{L}}\boldsymbol{\Phi}\mathbf{P}_{k-1}^{(\phi)}\widetilde{\mathbf{w}}_k^{(\phi)} = \widetilde{\boldsymbol{\alpha}}_k^\top\boldsymbol{\Phi}\mathbf{P}_{k-1}^{(\phi)}\boldsymbol{\Phi}^\top\boldsymbol{\mathcal{L}}\boldsymbol{\Phi}\mathbf{P}_{k-1}^{(\phi)}\boldsymbol{\Phi}^\top\widetilde{\boldsymbol{\alpha}}_k = \widetilde{\boldsymbol{\alpha}}_k^\top\mathbf{K}_{k-1}\boldsymbol{\mathcal{L}}\mathbf{K}_{k-1}\widetilde{\boldsymbol{\alpha}}_k. \quad (12)$$

E. Kernel Laplacian Support Vector Analysis

Proceeding as in the linear case, we formulate a Laplacian SVM on the non-linearly deflated data. The SVM optimization problem at iteration k takes the form

$$\min_{\boldsymbol{\alpha}_k} \frac{1}{2}\boldsymbol{\alpha}_k^\top\mathbf{K}_{k-1}\boldsymbol{\alpha}_k + \frac{\lambda}{2}\boldsymbol{\alpha}_k^\top\mathbf{K}_{k-1}\boldsymbol{\mathcal{L}}\mathbf{K}_{k-1}\boldsymbol{\alpha}_k + C\sum_{i=1}^{n}\xi_i, \quad (13)$$

$$\text{s.t} \quad y_i(\boldsymbol{\alpha}_k^\top\mathbf{k}_{k-1}^{(i)} + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n.$$

The Lagrangian is given by

$$L(\boldsymbol{\alpha}_k, b, \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\xi}) = \frac{1}{2}\boldsymbol{\alpha}_k^\top(\mathbf{K}_{k-1} + \lambda\mathbf{K}_{k-1}\boldsymbol{\mathcal{L}}\mathbf{K}_{k-1})\boldsymbol{\alpha}_k + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\mu_i(y_i(\boldsymbol{\alpha}_k^\top\mathbf{k}_{k-1}^{(i)} + b) - 1 + \xi_i) - \sum_{i=1}^{n}\nu_i\xi_i. \quad (14)$$

By defining the matrix

$$\mathbf{B}_{k-1} = \mathbf{K}_{k-1} + \lambda \mathbf{K}_{k-1} \boldsymbol{\mathcal{L}} \mathbf{K}_{k-1} = \mathbf{K}_{k-1}(\mathbf{I} + \lambda \boldsymbol{\mathcal{L}} \mathbf{K}_{k-1}), \tag{15}$$

the modified Karush-Kuhn-Tucker (KKT) conditions are

$$\begin{aligned}
&\nabla_{\boldsymbol{\alpha}_k} L(\boldsymbol{\alpha}_{k,o}, b_o, \boldsymbol{\mu}_o, \boldsymbol{\nu}_o, \boldsymbol{\xi}_o) = \mathbf{0} \\
&\Leftrightarrow \mathbf{B}_{k-1}\boldsymbol{\alpha}_{k,o} = \sum_{i=1}^{n} \mu_{i,o} y_i \mathbf{k}_{k-1}^{(i)}, \\
&\frac{\partial}{\partial b} L(\boldsymbol{\alpha}_{k,o}, b_o, \boldsymbol{\mu}_o, \boldsymbol{\nu}_o, \boldsymbol{\xi}_o) = 0 \Leftrightarrow \sum_{i=1}^{n} \mu_{i,o} y_i = 0, \\
&\frac{\partial}{\partial \xi_i} L(\boldsymbol{\alpha}_{k,o}, b_o, \boldsymbol{\mu}_o, \boldsymbol{\nu}_o, \boldsymbol{\xi}_o) = 0 \Leftrightarrow \nu_{i,o} = C - \mu_{i,o}, \\
&y_i(\boldsymbol{\alpha}_{k,o}^{\top} \mathbf{k}_{k-1}^{(i)} + b_o) - 1 + \xi_i \geq 0, \\
&0 \leq \mu_{i,o} \leq C,\ \nu_{i,o} \geq 0,\ \xi_i \geq 0,\ \nu_{i,o}\xi_{i,o} = 0, \quad i = 1, \ldots, n, \\
&\mu_{i,o}(y_i(\boldsymbol{\alpha}_{k,o}^{\top} \mathbf{k}_{k-1}^{(i)} + b_o) - 1 + \xi_i) = 0, \quad i = 1, \ldots, n.
\end{aligned} \tag{16}$$

By using the KKT conditions the dual becomes

$$\begin{aligned}
&\max_{\boldsymbol{\mu}} \sum_{i=1}^{n} \mu_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mu_i \mu_j y_i y_j (\mathbf{k}_{k-1}^{(i)})^{\top} \mathbf{B}_{k-1}^{-1} \mathbf{k}_{k-1}^{(j)}, \\
&\text{s.t. } \mu_i \geq 0 \text{ and } \sum_{i=1}^{n} \mu_i y_i = 0.
\end{aligned} \tag{17}$$

The optimal weight vector is given by

$$\boldsymbol{\alpha}_{k,o} = \mathbf{B}_{k-1}^{-1} \sum_{i=1}^{N} \mu_{i,o} y_i \mathbf{k}_{k-1}^{(i)}, \tag{18}$$

and the corresponding separating hyperplane is

$$\begin{aligned}
f(\mathbf{x}) &= \text{sgn}(\boldsymbol{\alpha}_{k,o}^{\top} \mathbf{k}_{\mathbf{x}} + b) \\
&= \text{sgn}(\sum_{i=1}^{n} \mu_{i,o} y_i (\mathbf{k}_{k-1}^{(i)})^{\top} \mathbf{B}_{k-1}^{-1} \mathbf{k}_{\mathbf{x}} + b),
\end{aligned} \tag{19}$$

where $\mathbf{k}_{\mathbf{x}} = [k(\mathbf{x}_1, \mathbf{x}), \ldots, k(\mathbf{x}_n, \mathbf{x})]^{\top}$. We can easily see that the above dual corresponds to a standard SVM formulation using the deflation kernel matrix

$$\boldsymbol{\mathcal{K}}' = \mathbf{K}_{k-1} \mathbf{B}_{k-1}^{-1} \mathbf{K}_{k-1}. \tag{20}$$

The resulting vector $\boldsymbol{\alpha}_k$ is normalized using (11) to obtain $\widetilde{\boldsymbol{\alpha}}_k$. The above procedure is repeated until we reach the desired dimensionality n'. A matrix $\widetilde{\mathbf{A}} \in \mathbb{R}^{n \times n'}$ is constructed, whose columns are the vectors $\widetilde{\boldsymbol{\alpha}}_k$, $k = 1, \ldots, n'$. The resulting projected data are then given by $\mathbf{K}_{n'} = \mathbf{K}\widetilde{\mathbf{A}} \in \mathbb{R}^{n \times n'}$. The complete algorithm for the non-linear case is described in Algorithm 2.

Algorithm 2 Kernel Laplacian Support Vector Analysis (KLSVA)

1: **Input**: Data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^{\top} \in \mathbb{R}^{n \times d}$ and label vector $\mathbf{y} = [y_i, \ldots, y_n]^{\top} \in \{-1, +1\}^n$. Graph Laplacian $\mathcal{L} \in \mathbb{R}^{n \times n}$. Target dimensionality n'.
2: Compute the kernel matrix $\mathbf{K}$ from $\mathbf{X}$.
3: $\mathbf{K}_0 = \mathbf{K}$.
4: $\widetilde{\mathbf{A}} = []$.
5: **for** $k = 1$ **to** n' **do**
6: Compute the kernel matrix $\boldsymbol{\mathcal{K}}' = \mathbf{K}_{k-1}\mathbf{B}_{k-1}^{-1}\mathbf{K}_{k-1}$, with $\mathbf{B}_{k-1}$ as in (15).
7: Train a SVM using the above kernel.
8: Compute the normal vector $\widetilde{\boldsymbol{\alpha}}_k$ from (18) and (11).
9: $\widetilde{\mathbf{A}} = [\widetilde{\mathbf{A}}; \widetilde{\boldsymbol{\alpha}}_k]$.
10: $\mathbf{K}_k = \mathbf{K} - \mathbf{K}_{k-1}(\sum_{j=1}^{k-1} \widetilde{\boldsymbol{\alpha}}_j \widetilde{\boldsymbol{\alpha}}_j^{\top})\mathbf{K}_{k-1}$.
11: **end for**
12: Final projected data $\mathbf{K}_{n'} = \mathbf{K}\widetilde{\mathbf{A}}$.

III. Extending to Multi-Class Datasets

The most straightforward approach to extend our proposed technique to the multi-class case is to use the One versus All approach. That is, in each deflated subspace (i.e., iteration of our algorithm) we consider a different binary problem setting, which involves one class against all the others. Obviously, by adopting this approach, each one of the created projection vectors will discriminate the current unitary class involved against the remaining ones. Intuitively, the process of successively projecting our data onto the created projection vectors can be thought of as a process of successively separating the classes of our initial data. The orthogonality between the projection vectors is guaranteed, since in each step of our process the data are already projected onto a deflated subspace that is orthogonal to all the previous deflated subspaces considered. Thus, by using the One versus All multi-class extension in each iteration of the Algorithms 1 and 2 the class labels change to $+1$ for the unitary class considered and to -1 for all the remaining classes. After considering the n_c problems, where n_c is the number of classes, the procedure can continue by training the SVM on one random chosen binary problem of the n_c available, or by choosing one binary setting with some measurable characteristic like the number of samples of the unitary class or other.

IV. Experimental Results

We compare our proposed linear deflation procedure LLSVA with several state-of-the-art linear Dimensionality Reduction methods, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Locally Preserving Projections (LPP) and Margin Maximizing Discriminant Analysis (MMDA). Our non-linear deflation procedure is compared against Kernel Principal Component Analysis (KPCA), Kernel Discriminant Analysis (KDA), Laplacian Eigenmaps (LE) and a modified Kernel Margin Maximizing Discriminant Analysis (KMMDA) that we have derived using our analysis in the previous Section. We should note that the authors in [7] have not derived the kernel MMDA analytically. To evaluate the above methods we use 19 benchmark datasets from the UCI [3] and Statlog [8] repositories, 9 of which refer to binary classification problems and the other 10 refer to multi-class classification problems. All the features of each dataset are scaled to the interval $[-1, +1]$.

To evaluate the test error in each experiment we use 5-fold Cross Validation. That is, in each fold we compute the projection vectors based on the training set, then we project the feature vectors of the fold's training and test sets onto the already computed projection vectors. Finally, we use the 1-Nearest Neighbor classifier for the final classification of the

TABLE I: Error rates of Linear Dimensionality Reduction methods on binary problems.

Dataset	Method					
	1-NN	**LLSVA**	**PCA**	**LDA**	**LPP**	**MMDA**
Australian (14)	20.15 %	**16.82 % (3)**	18.10 % (5)	19.43 % (1)	17.68 % (5)	17.69 % (12)
Breast (10)	4.68 %	**2.49 % (5)**	3.22 % (6)	4.40 % (1)	3.95 % (3)	3.66 % (8)
Diabetes (8)	29.17 %	**26.96 % (8)**	28.78 % (6)	31.12 % (1)	29.17 % (8)	29.69 % (4)
German (24)	32.8 %	**27.20 % (2)**	30.90 % (4)	29.70 % (1)	31.70 % (8)	29.80 % (6)
Heart (13)	24.44 %	**18.89 % (2)**	23.70 % (9)	25.19 % (1)	21.85 % (8)	20.74 % (10)
Ionosphere (34)	13.69 %	**6.83 % (7)**	10.26 % (18)	21.64 % (1)	8.83 % (11)	8.83 % (22)
Liver (6)	38.26 %	**32.75 % (4)**	38.26 % (6)	40.58 % (1)	37.97 % (6)	36.52 % (1)
Sonar (60)	12.49 %	16.97 % (45)	**10.60** % (18)	31.76 % (1)	13.85 % (14)	20.71 % (22)
Transfusion (4)	32.09 %	**25.54 % (3)**	27.68 % (2)	26.74 % (1)	26.60 % (1)	27.81 % (3)

TABLE II: Error rates of Kernel Dimensionality Reduction methods on binary problems.

Dataset	Method					
	1-NN	**KLSVA**	**KPCA**	**KDA**	**LE**	**KMMDA**
Australian (14)	20.15 %	**14.93 % (1)**	19.86 % (11)	17.25 % (1)	25.36 % (14)	16.24 % (7)
Breast (10)	4.68 %	**2.78 % (3)**	3.22 % (10)	3.66 % (1)	4.24 % (1)	3.80 % (7)
Diabetes (8)	29.17 %	**23.43 % (1)**	30.74 % (7)	27.87 % (1)	36.47 % (7)	23.83 % (4)
German (24)	32.8 %	28.60 %	31.50 % (21)	29.10 % (1)	28.40 % (15)	**28.30 % (4)**
Heart (13)	24.44 %	**18.89 % (4)**	29.63 % (11)	22.22 % (1)	20.74 % (11)	19.26 % (2)
Ionosphere (34)	13.69 %	**5.13 % (3)**	6.27 % (8)	5.71 % (1)	6.27 % (1)	5.99 % (11)
Liver (6)	38.26 %	26.96 % (1)	42.03 % (4)	36.81 % (1)	40.00 % (3)	**26.67 % (4)**
Sonar (60)	12.49 %	**10.58 % (4)**	25.89 % (30)	13.01 % (1)	12.03 % (5)	12.99 % (1)
Transfusion (4)	32.09 %	**16.66 % (2)**	28.88 % (3)	27.41 % (1)	29.95 % (4)	20.72 % (1)

TABLE III: Error rates of Linear Dimensionality Reduction methods on multi-class problems.

Dataset	Method				
	1-NN	**LLSVA**	**PCA**	**LDA**	**LPP**
Balance (4)	20.00 %	**8.81 % (2)**	23.84 % (4)	10.40 % (2)	23.52 % (4)
Ecoli (7)	17.07 %	**12.67 % (6)**	17.07 % (7)	14.62 % (7)	16.22 % (6)
Glass (9)	30.30 %	**24.79 % (7)**	30.31 % (8)	42.52 % (3)	29.87 % (7)
Iris (4)	4.00 %	**2.67 % (1)**	4.00 % (4)	3.33 % (1)	4.00 % (2)
Soy (35)	8.46 %	**4.56 % (25)**	7.17 % (30)	8.81 % (2)	7.50 % (6)
Tae (5)	35.10 %	**29.79 % (1)**	31.10 % (1)	33.16 % (2)	34.47 % (3)
Thyroid (5)	4.65 %	2.79 % (3)	**2.32 % (3)**	5.12 % (1)	3.72 % (3)
Vehicle (18)	29.32 %	**20.10 % (18)**	29.32 % (16)	25.42 % (3)	21.75 % (14)
Vowel (10)	1.61 %	1.52 % (9)	1.61 % (10)	**1.31 % (10)**	1.92 % (10)
Wine (13)	4.56 %	**0 % (8)**	3.88 % (3)	1.72 % (2)	1.16 % (3)

TABLE IV: Error rates of Kernel Dimensionality Reduction methods on multi-class problems.

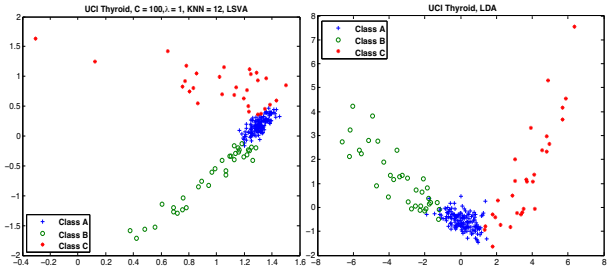
Dataset	Method				
	1-NN	**KLSVA**	**KPCA**	**KDA**	**LE**
Balance (4)	20.00 %	**2.24 % (3)**	19.51 % (4)	4.48 % (2)	15.19 % (1)
Ecoli (7)	17.07 %	15.69 % (7)	15.73 % (6)	**14.34 % (7)**	22.39 % (7)
Glass (9)	30.30 %	**25.62 % (6)**	33.16 % (7)	27.97 % (5)	35.06 % (9)
Iris (4)	4.00 %	**2.67 % (3)**	4.00 % (3)	4.00 % (2)	4.00 % (2)
Soy (35)	8.46 %	**5.55 % (11)**	11.08 % (26)	13.37 % (2)	8.48 % (12)
Tae (5)	35.10 %	**31.81 % (5)**	46.99 % (5)	52.22 % (2)	43.70 % (4)
Thyroid (5)	4.65 %	**1.40 % (3)**	2.79 % (2)	4.18 % (2)	12.09 % (5)
Vehicle (18)	29.32 %	**15.83 % (17)**	36.40 % (18)	**15.83 % (3)**	19.27 % (10)
Vowel (10)	1.61 %	**0.81 % (10)**	2.12 % (10)	1.41 % (10)	1.01 % (10)
Wine (13)	4.56 %	2.22 % (5)	2.30 % (13)	**1.70 % (2)**	2.81 % (2)

projected feature vectors. For our method and MMDA the SVM cost parameter is fixed to $C = 100$. The regularization parameter λ of the proposed method is optimized in the interval $\lambda \in [0.1, 100]$ by grid search on 30 values. Furthermore, the graph Laplacian is contructed from a KNN graph with $K = 10$ and weights given by $W_{ij} = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2}$ with $\sigma = 1$. The results for the linear methods on binary problems are shown in Table I whereas for the multi-class problems are shown in Table III. In each column we present the minimum classification error obtained by each method and in the parentheses the dimensionality of the feature vectors for which this error is achieved. We observe that in the linear case, for both the binary and multi-class datasets, our approach achieves higher accuracy than the competing methods in almost all of the available datasets. In Table II we present the results for the non-linear Dimensionality Reduction methods on binary problems and in Table IV the respective results on multi-class problems. Our method is denoted by KLSVA. As in the linear case, the experimental setting is the same. The situation here is again similar to the linear case. In all except two datasets for the binary problems (i.e., German and Liver) and two for the multi-class problems (i.e., Ecoli and Wine) we obtain

better results than the other investigated methods. It is also worth mentioning that in most cases our method outperforms the other approaches with more than 2% difference in the classification error, while in the non-winning datasets our method loses slightly with less than 1% difference in the classification error attained by the winning method.

V. Visualization

In Figures 1 and 2 we provide some 2-D visualization results on the Thyroid dataset. We compare our linear method with LDA, PCA and LPP. Our kernel method is compared with KDA, KPCA and LE. In the linear case, we observe that our method works very similar to LDA, where the goal is to make the classes as compact and as far from each other as possible. In the non-linear setting, we observe that our approach is able to successfully separate all the classes. Similar results are obtained with LE, however our method produces more compact clusters.



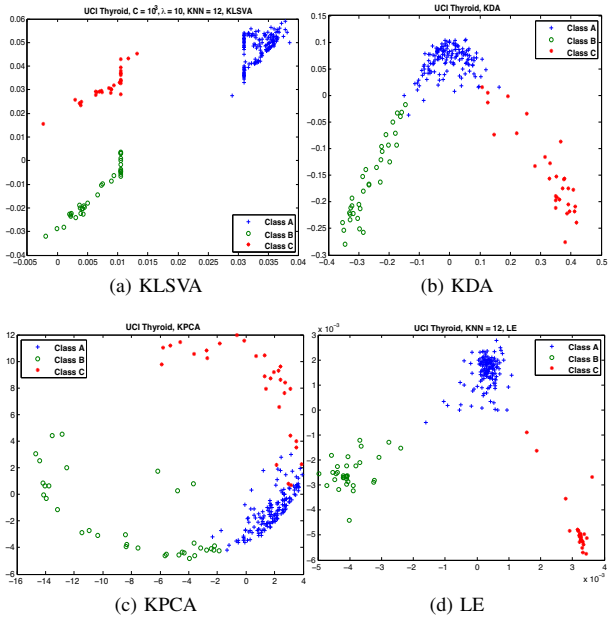
(a) LSVA (b) LDA

(c) PCA (d) LPP

Fig. 1: UCI Thyroid Linear Methods

VI. Conclusions

In this paper a novel dimensionality reduction method based on successive Laplacian SVM on deflated subspaces has been proposed. The resulting normal vectors of the trained hyperplanes are used to project the initial data points in lower dimension and therefore, they provide new discriminative information for feature extraction. We have shown that these vectors can be computed by solving a standard SVM with a specially designed deflation kernel. In our theoretical analysis we have investigated both the linear and the non-linear case. For the non-linear case we have provided an explicit form for the deflation matrix on the mapped feature vectors onto the RKHS. Experiments in benchmark datasets have illustrated the strength of the proposed approach against the state-of-art in Dimensionality Reduction.



(a) KLSVA (b) KDA

(c) KPCA (d) LE

Fig. 2: UCI Thyroid Non-Linear Methods

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[2] Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396, 2003.

[3] A. Frank and A. Asuncion. UCI Machine Learning Repository, 2010.

[4] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition, October 1990.

[5] Xiaofei He and Partha Niyogi. Locality Preserving Projections. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

[6] I. Jolliffe. *Principal Component Analysis*. Encyclopedia of Statistics in Behavioral Science. October 2005.

[7] András Kocsor, Kornél Kovács, and Csaba Szepesvári. Margin Maximizing Discriminant Analysis. In *In Proceedings of the 15th European Conference on Machine Learning*, pages 227–238, 2004.

[8] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, editors. *Machine Learning, Neural and Statistical Classication*. Ellis Horwood Series in Artificial Intelligence. Prentice Hall, 1994.

[9] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Schölkopf, and Klaus Robert Müller. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop*, pages 41–48, August 1999.

[10] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, 1998.

[11] Bernhard Schölkopf and Alexander J. Smola. *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, December 2001.

[12] Zenglin Xu, Irwin King, Michael Rung-Tsong Lyu, and Rong Jin. Discriminative Semi-Supervised Feature Selection via Manifold Regularization. *IEEE Transactions on Neural Networks*, 21(7):1033–1047, July 2010.