Stochastic Discriminant Analysis

Mika Juuti*, Francesco Corona†‡ and Juha Karhunen§
*Aalto University, School of Science, Department of Mathematics and Systems Analysis
Espoo, Finland, Email: mika.juuti@aalto.fi
†Aalto University, School of Science, Department of Computer Science
Espoo, Finland, Email: francesco.corona@aalto.fi
‡Federal University of Ceara, Center of Technology
Department of Teleinformatics Engineering, 60455-760 Fortaleza, Brazil
§Aalto University, School of Science, Department of Computer Science
Espoo, Finland, Email: juha.karhunen@aalto.fi

***Abstract*— In this paper, we consider a linear supervised dimension reduction method for classification settings: Stochastic Discriminant Analysis. This method matches point adjacencies in the projection space with those in a response space. These adjacencies are represented by t-distributed probabilities. The matching is done by minimizing the Kullback-Leibler divergence between the two distributions. The performance of the algorithm is compared against state-of-the-art methods in supervised dimension reduction. We found that the performance of SDA is comparable to (and sometimes better than) state-of-the-art methods in supervised linear dimension reduction. In the presence of multiple classes, low-dimensional SDA projections led to higher classification accuracies.**

I. Introduction

Dimension reduction is related to the fundamental problem of determining what portion of the data is useful. Often thought to tackle the problem of which variables we want to preserve and what we can discard, in our setting, dimensionality reduction is combining variables into meaningful new variables that are useful for classification or regression.

The literature over Fisher's Linear Discriminant Analysis (LDA) [1] and its different modifications is vast. LDA produces a linear projection of the original data in a low-dimensional hyperplane. The cost function in LDA maximizes between-class scatter while minimizing the within-class scatter. LDA has problems with singular within-class scatter matrices, which is why it is often coupled with PCA in image recognition tasks [2]. Partial Least Squares regression is a supervised linear dimension reduction technique that tries to find subspaces in the input matrix that explain the largest amount of variance in the response matrix. When used with categorial response variables it is referred to as PLS-DA [3]. Kernel Dimension Reduction (KDR) [4] is a sufficient dimension reduction method [5] for classification and regression data. A sufficient dimension reduction contains all the regression information that the original space contained about the response variable. KDR tries to find the central subspace [5] for the input data, which in the intersection of all dimension reduction subspaces. The method is demanding in terms of computation and memory consumption. A gradient version of KDR has been developed for faster computation, called gKDR [6]. Supervised PCA by Barshan et al. is a regression technique that finds the principal components with maximum dependence on the given response variable. SPCA tries to find variables that are orthogonal in a kernel space of the response variable. A dual-space and kernel variant of SPCA (KSPCA) [7] have also been developed, extending the usage scenarios of the method.

All the methods discussed previously were supervised techniques. The following methods use no output information produce their embeddings. Neighborhood embedding techniques recreate a high-dimensional neighborhood structure in a low-dimensional space. The methods preserve point-to-point neighborhood relations. The low-dimensional embedding is created by defining probability mass functions based on point-to-point adjacencies in both high-dimensional and low-dimensional space. An information divergence between these two probability distributions is then iteratively decreased. The most common information divergence is the Kullback-Leibler divergence [8].

Some of the most popular and famous *point-to-point mappings* are t-SNE [9] and NeRV [10]. t-SNE describes high-dimensional point adjacencies as probabilities calculated from Gaussian kernels and low-dimensional adjacencies as probabilities calculated from t-distributed kernels. The motivation for the asymmetric matchup being that it solves the so called crowding problem. NeRV matches a convex combination of the divergences between the low-dimensional point adjacencies to the high-dimensional point adjacencies and vice versa. The proportion is hand-tuned, giving the user some control in penalizing precision and recall errors. *Parametric* methods provide a mapping of the data points. Amongst others, Parametric t-SNE learns a mapping by using a deep neural network.

In this paper, we present a supervised dimensionality reduction technique for classification. We are looking for a linear mapping of the data points from the high-dimensional space to the low-dimensional embedding by matching t-distributed point adjacencies. The matching is done using the Kullback-Leibler divergence. The method is similar to LDA in the regard that we want to maximize the between-class distances and minimize within-class distances, with a focus on extremely low-dimensional projections with multiple classes.

In what follows, Section II discusses the proposed method. Section III discusses how to minimize the cost function. Section IV evaluates the proposed method experimentally against some traditional and state-of-the-art approaches to dimension reduction. Finally, Section V summarizes the discussed topics.

II. Stochastic Discriminant Analysis

Formally, we are reducing the size of a data matrix containing n observations each with d_0 variables (dimensions): $\mathbf{X} = [\mathbf{x}_1\mathbf{x}_2...\mathbf{x}_n]^T \in \mathbb{R}^{n\times d_0}$. We reduce the amount of variables in $\mathbf{X}$ by finding a linear subspace of it: $\mathbf{Z} = [\mathbf{z}_1\mathbf{z}_2...\mathbf{z}_n]^T = \mathbf{XW}$, where $\mathbf{Z}$ is a $\mathbb{R}^{n\times d_t}$ matrix, $\mathbf{W} \in \mathbb{R}^{d_0\times d_t}$, and $d_t \ll d_0$. We are using class information from the response matrix $\mathbf{Y} = [\mathbf{y}_1\mathbf{y}_2...\mathbf{y}_n]^T \in \mathbb{I}^{n\times d_y}$ to find this projection. The response variables $\mathbf{y}_i$ are sequences of d_y binary numbers, specifying the class labels. The linear subspace is searched by matching point adjacencies (probability mass functions) in the embedding space with point adjacencies in the response space. The probabilities between points i and j in the $\mathbf{Z}$-space are:

$$q_{ij}(\mathbf{W}) = \frac{(1+\|\mathbf{z}_i-\mathbf{z}_j\|_2^2)^{-1}}{\sum_{k=1}^{n}\sum_{l=1,l\neq k}^{n}(1+\|\mathbf{z}_k-\mathbf{z}_l\|_2^2)^{-1}}, \quad (1)$$

where $\mathbf{z}_i = \mathbf{x}_i\mathbf{W}$ is the low-dimensional embedding coordinate. The elements q_{ij} are called t-distributed, because of the similarity with the probability density function of the t-distribution. The probabilities of response space are $p_{ij} = \bar{p}_{ij}/\sigma$, where the normalization term $\sigma = \sum_{ij}\bar{p}_{ij}$ and

$$\bar{p}_{ij} = \begin{cases} 1, & \text{if } \mathbf{y}_i = \mathbf{y}_j \\ \epsilon, & \text{otherwise} \end{cases}, \quad (2)$$

where $\epsilon > 0$ is any small number. The target probabilities define *ideal* distances. By setting $\epsilon \to 0$ we essentially have a stochastic version of the LDA principle: minimize within-class ($\mathbf{y}_i = \mathbf{y}_j$) and maximize between-class ($\mathbf{y}_i \neq \mathbf{y}_j$) distances.

The Kullback-Leibler divergence [11] is a measure of inefficiency when trying to encode a distribution P using another distribution Q. For two discrete probability mass functions P and Q, the divergence is $D_{KL}(P||Q) = \sum_{i=1}^{N} p_i \log(p_i/q_i)$, where $i = 1,\ldots,N$ is the index of the probability point masses. The Kullback-Leibler divergence is zero only in the case that $Q = P$. However, it is not a real distance because it does not fulfill the triangle inequality and it is not symmetric. We can write the cost function as:

$$J(\mathbf{W}) = \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij} log\frac{p_{ij}}{q_{ij}(\mathbf{W})} + \lambda\sum_{i=1}^{d_0}\sum_{j=1}^{d_t}\mathbf{W}_{ij}^2. \quad (3)$$

We are searching for the thin linear projection matrix $\mathbf{W}$ that minimizes the Kullback-Leibler divergence of approximating probability distribution P with Q. The inefficiency of encoding *ideal* distances in the response space using *realized* distances in the embedding space is measured. The t-distribution causes asymmetric distance penalties: the cost function is more sensitive to deviations in within-class distances than it is to between-class distances. Matching distances creates a regular simplex structure if the target dimension is high enough. This optimal structure is found already in $\nu - 1$ dimensional space, where ν is the number of classes in the dataset. If lower than this, the points are placed so as to maximize the space between classes. The expected effect is shown in Figure 1. In practice, the optimization criterion converges slowly with small values of ϵ. Therefore we choose $\epsilon = 1/\nu$ in general. We can also use an additive Tikhonov regularization term [12]. If the value of the regularization term λ is searched by cross-validation, we refer to the method as Regularized SDA, denoted as RSDA. Normally λ is set to zero. Tikhonov regularization is often applied to *ill-posed* [13] problems. In SDA, we have local solutions where the solution depends on the initialization. The initial solution in SDA is obtained with PCA, giving orthogonal vectors with maximum variance. In high-dimensional cases, regularization can help in moving past the initialization. Additionally, the optimization process can also be made smoother by constraining the elements of $\mathbf{W}$.

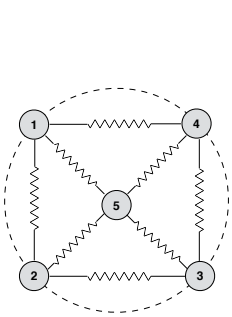

Fig. 1: **An ideal embedding of 5 classes into 2D.** Same-class elements are separated from other class elements.

III. Cost function

Equation (3) presented the cost function. The minimum of that cost function is approached by minimizing its gradient. The essential steps in obtaining the gradient are written here. We use the shorthand notation $q_{ij} = q_{ij}(\mathbf{W})$. We also write the distance in the embedding space as $D_{ij} = D_{ij}(\mathbf{W}) = \|\mathbf{z}_i(\mathbf{W})-\mathbf{z}_j(\mathbf{W})\|_2^2 = \boldsymbol{\tau}_{ij}\mathbf{W}\mathbf{W}^T\boldsymbol{\tau}_{ij}^T = (\mathbf{x}_i-\mathbf{x}_j)\mathbf{W}\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)^T$. The matrices $\mathbf{P}, \mathbf{Q}, \bar{\mathbf{Q}}$ and $\mathbf{D}$ are $\mathbb{R}^{n\times n}$ matrices. p_{ij}, q_{ij}, $\bar{q_{ij}}$ and D_{ij} denote their elements.

$$\begin{aligned}
\frac{dKL(P||Q(\mathbf{W}))}{d\mathbf{W}} &= \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}\frac{1}{q_{ij}}(-1)\frac{dq_{ij}}{d\mathbf{W}} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}(-1)\Big[\sum_{k=1}^{n}\sum_{l=1}^{n} q_{kl}\bar{q}_{kl}\frac{dD_{kl}}{d\mathbf{W}} - \bar{q}_{ij}\frac{dD_{ij}}{d\mathbf{W}}\Big] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}\bar{q}_{ij}\frac{dD_{ij}}{d\mathbf{W}} - \sum_{k=1}^{n}\sum_{l=1}^{n} q_{kl}\bar{q}_{kl}\frac{dD_{kl}}{d\mathbf{W}} \quad (4) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}(p_{ij}-q_{ij})\bar{q}_{ij}\frac{dD_{ij}}{d\mathbf{W}} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}(p_{ij}-q_{ij})\bar{q}_{ij}\boldsymbol{\tau}_{ij}^T\boldsymbol{\tau}_{ij}\mathbf{W},
\end{aligned}$$

since $\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}k = (\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij})k = k$, where k is an arbitrary constant. Here $(1+D_{ij})^{-1} = \bar{q}_{ij}$ denotes the unnormalized probability. Adding the regularization we get

$$\frac{dJ}{d\mathbf{W}} = \sum_{i=1}^{n}\sum_{j=1}^{n}(p_{ij}-q_{ij})\bar{q}_{ij}\boldsymbol{\tau}_{ij}^T\boldsymbol{\tau}_{ij}\mathbf{W} + 2\lambda\mathbf{W}. \quad (5)$$

In matrix form the expression becomes

$$\nabla_{\mathbf{W}}J = 2\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W} + 2\lambda\mathbf{W}, \quad (6)$$

where $\mathbf{L} = \mathbf{G}^+ - \mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ is calculated as

$$\begin{aligned} \mathbf{G} &= (\mathbf{P} - \mathbf{Q}) \odot \bar{\mathbf{Q}} \\ \mathbf{G}^+ &= \mathbf{G} + \mathbf{G}^T \\ \mathbf{\Lambda} &= \sum_j \mathbf{G}^+_{ij} \end{aligned} \tag{7}$$

Here $\odot$ denotes the Hadamard product and $\mathbf{G}^+$ is a symmetrized matrix of $\mathbf{G} \in \mathbb{R}^{n \times n}$ and $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ is a diagonal matrix containing the row sum of $\mathbf{G}^+$. The matrix $\mathbf{L}$ is the difference between two Laplacian matrices $\mathbf{L} = \mathbf{L_P} - \mathbf{L_Q}$, where $\mathbf{L_P}$ is calculated from the adjacency matrices $\mathbf{G_P} = \mathbf{P} \odot \bar{\mathbf{Q}}$ and $\mathbf{G_Q} = \mathbf{Q} \odot \bar{\mathbf{Q}}$. A Laplacian matrix is a symmetric diagonally dominant matrix and therefore positive definite, however $\mathbf{L}$ need not be positive semi-definite.

There are many ways of optimizing the cost function based on gradient information alone, for example Conjugate Gradient [14], [15], [16] methods and the Limited-memory BFGS [17], [15] algorithm are efficient at solving problems with a large number of variables. The partial Hessian $\mathbf{H}^+ = \mathbf{X}^T \mathbf{L_P} \mathbf{X}$ has been successfully used in neighborhood embedding methods, where it is called the *Spectral Direction* optimization method [18].

IV. Experimental evaluation

The experimental evaluation is divided into two parts. First, three case studies on different datasets are conducted in subsection IV-A. In the case studies, the classification accuracies for a range of target dimensionality values are calculated and 2D projections are visualized. We will also describe a regularization parameter search scheme for SDA in subsection IV-A1 and compare the runtime with different optimization algorithms in subsection IV-A4. In subsection IV-B, a comparison of the 2D projection qualities of state-of-the-art methods is conducted over a range of datasets. The utilized datasets are summarized in Table I.

We will define the hyperparameters used of various methods here. Our proposed method SDA is initialized with the PCA initial solution is all tests. In SPCA, we chose the delta kernel [7] for the response space. In the kernel version of SPCA, we selected the delta kernel for response space and a Gaussian kernel for the input space, setting the width of the Gaussian to the median value of the squared interpoint distances. gKDR was run in the partitioning mode (v) to reduce its memory requirements. The variables of each dataset were standardized: mean-centered and normalized to unit variance.

A. Case studies with three high-dimensional datasets

Three high-dimensional image datasets were chosen and analyzed: Olivetti faces, USPS and COIL-20. All datasets have multiple classes. The Olivetti face dataset (IV-A1) contains images of 40 persons, each photographed in ten pictures with both normal and unusual facial expressions. The input dimensionality is very high. The USPS dataset (IV-A2) contains a large number of hand-written digits in ten classes in a smaller space. COIL-20 (IV-A3) features 20 high-dimensional images of rotating objects photographed at fixed angle intervals.

1) The Olivetti faces dataset: The Olivetti faces (ORL / AT&T) [19] contains 40 persons shot with 10 different angles and facial expressions totaling 400 images. Each sample is a 64-by-64 pixel image, amounting to 4096 variables. In our tests, two thirds of the data was randomly selected as training data elements and one third as test elements. The random selection was repeated ten times to acquire error bars.

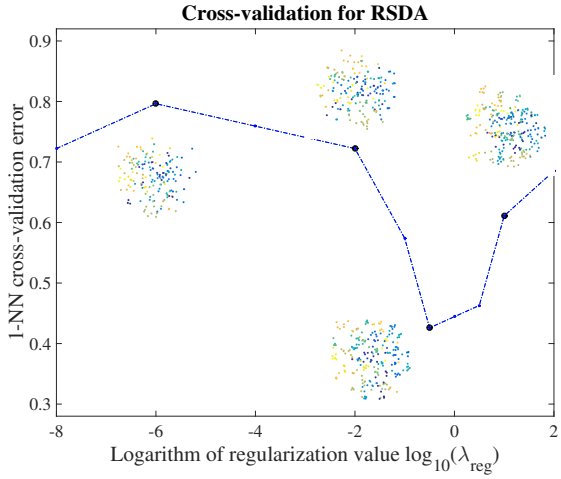

Fig. 2: **Tikhonov regularization parameter search.** The 2D embedding of the learning points are displayed for selected values of λ.

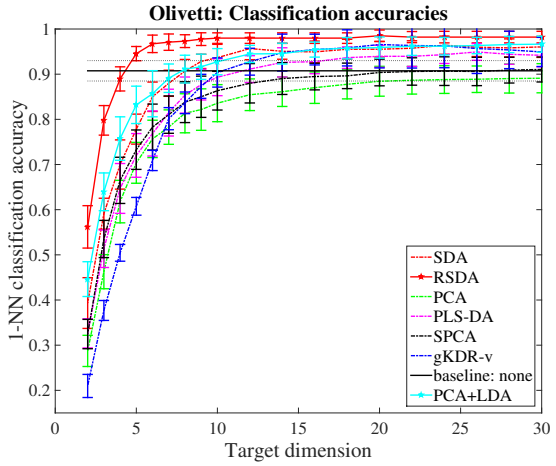

Fig. 3: **Olivetti dataset.** Classification accuracies after projection with different DR methods. The baseline is the classification accuracy in the original high-dimensional dataset.

Data set	Samples	Variables	Classes
USPS	9298	256	10
MNIST5k	5000	784	10
Phoneme	4509	256	5
Olivetti faces	400	4096	40
COIL-20	1440	16384	20
COIL-100	7200	16384	100
Iris	150	4	3
W. Breast cancer	683	9	2
Wine	178	13	3

TABLE I: Data sets used in this paper.

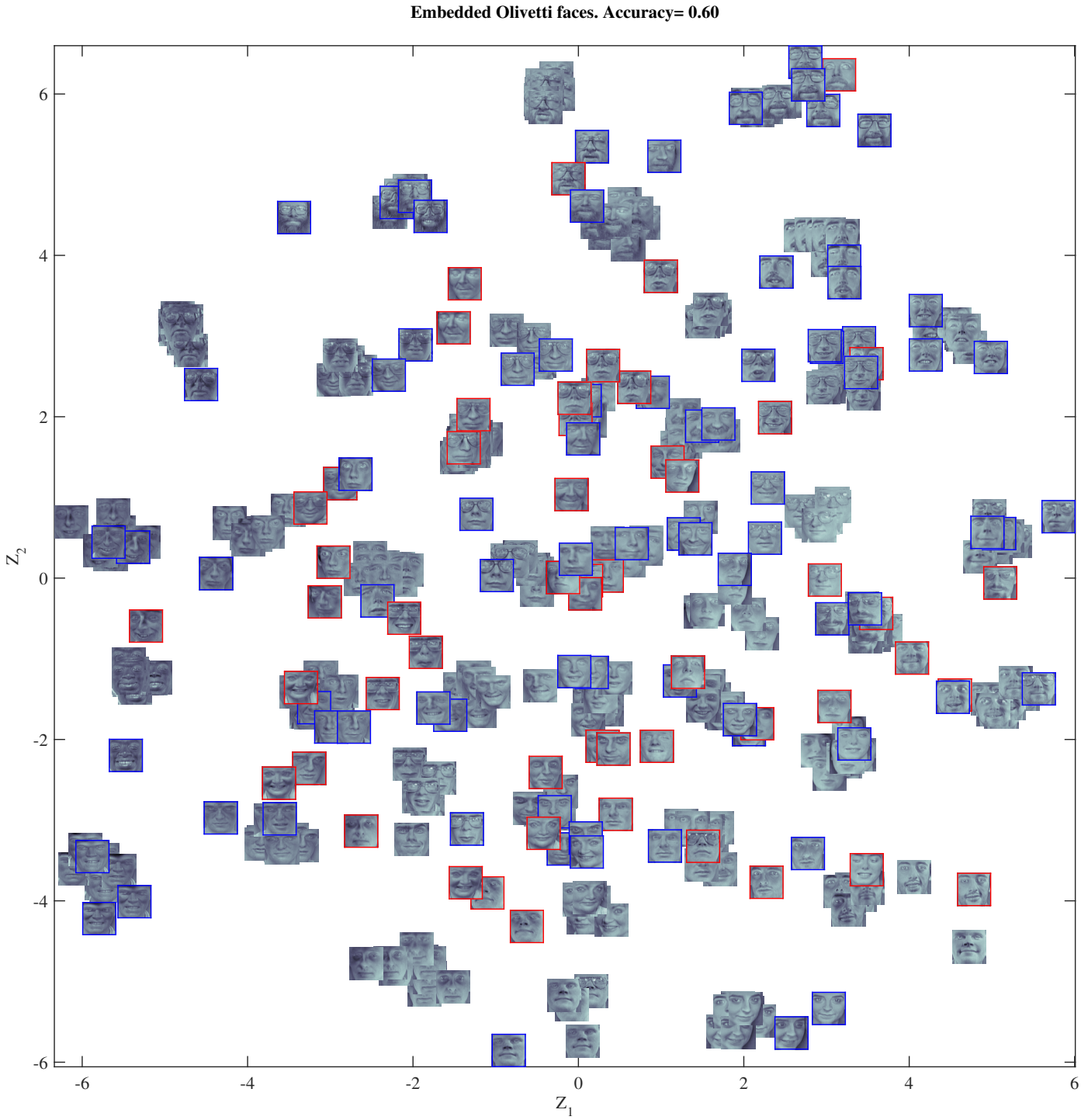

Fig. 4: **A representative RSDA linear embedding of the Olivetti faces dataset.** Colored borders denote projected test points. Red borders denote a misclassification, while blue borders denote a correct classification.

In the Olivetti dataset, Tikhonov regularization was used to guide the optimization process. The appropriate amount of regularization was searched by cross-validation. A random selection of 80% of the learning subset was used for training and 20% were used for cross-validation. The correct value is searched by trying six logarithmically intervalled values of λ from 10^2 to 10^{-8}. In total, ten regularization values are explored in the cross-validation search. Among these values, the one that gives the smallest 1-NN classification error is called λ^*, which is the regularization value used in the tests that follow. Figure 2 shows one regularization search procedure. The classification error is plotted against the logarithm of the regularization term. The dimension reduction in the figure is to 2D. We can observe that the search is magnified twice in the region $\lambda = 10^0$. Finally, the 1-NN classification error on the cross-validation dataset was found to be the smallest

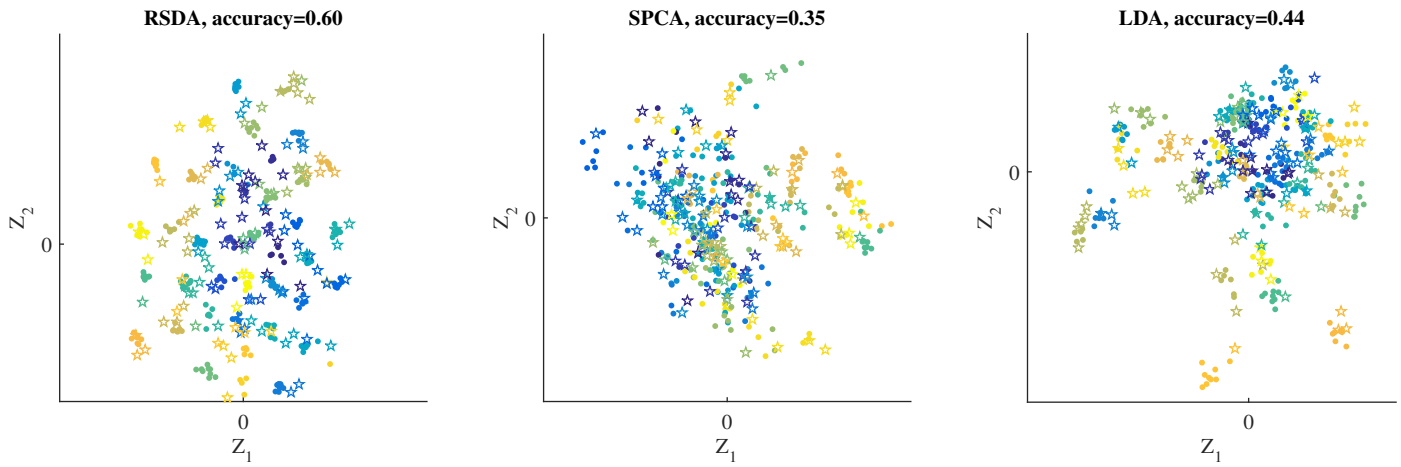

Fig. 5: **Three linear embeddings of the Olivetti faces dataset.** Dots denote projected learning points. Stars denote projected test points. The 1-NN classification accuracy resulting from this embedding is added to the title.

when $\lambda = 10^{-0.5} \approx 0.32$. This search was conducted until no progress could be made, evaluated at a tolerance 10^{-4}. The search procedure was fast, requiring approximately 3-4 seconds per value explored. The tolerance for optimality in the main algorithm was set at 10^{-5}.

Figure 3 shows the classification accuracy when learning the dimension reduction, using the λ search scheme described earlier. The error bars report the mean and standard deviation. The regularized algorithm shows the best performance here. The mean accuracy is highest among the methods and further, the error bars are among the narrowest. The method stabilizes at 98.0% 1-NN classification accuracy at 10D, above the 90.1% accuracy for using the whole input space.

An 2D RSDA embedding of the Olivetti faces is shown Figure 4. The correct classifications and misclassifications have been high-lighted in the figure. We can see for example that the face projected at $(3, 6)$ is projected a bit off as it should in fact have been projected at $(0.5, 4.5)$. The three best performing 2D projections in the Olivetti faces dataset have been compared in Figure 5. All figures show the 266 learning points and 134 test points for the same permutation. The same embedding is shown in both Figure 4 and Figure 5.

2) The USPS dataset: The US Postal Service [19] dataset contains 9298 hand-written images of digits. Each digits is represented by 16-by-16 pixel grey-scale images, giving 256 data dimensions. The data was divided randomly so that $2/3$ was used for training and $1/3$ was used for testing. The random selection was repeated ten times to obtain error bars.

The 1-NN classification accuracies are shown in Figure 6. SDA has the highest accuracies for small dimension reduction tasks. We can observe a saturation in LDA, SPCA and SDA. The saturation is related to the fact that the defined optimal simplex structure of the data is reached already at 9 dimensions. PCA, PLS-DA and gKDR-v approach or exceed the initial classification accuracy 96.3% in higher target dimensions.

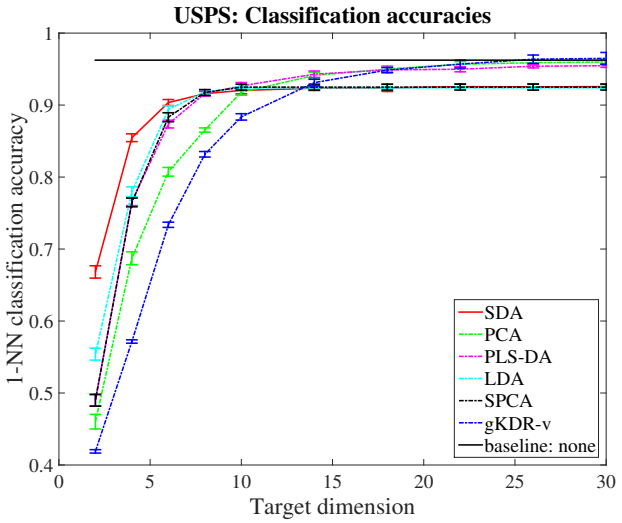

Fig. 6: **USPS dataset.** Classification accuracies for different DR methods. The baseline is the classification accuracy in the original high-dimensional dataset.

The three best performing 2D linear embedding of the data points are compared in Figure 7. In general, we can see that LDA and PLS-DA resemble multidimensional simplexes projected onto a subspace with too many classes crowding near the origin. Such projections are not ideal in the presence of multiple classes. On the contrary, SDA tends to fill a 2D circle, ultimately resulting in a higher class discrimination ability.

3) COIL-20 Object Images: The Columbia Object Image Library contains rotating images of 20 objects, photographed at 5 degree intervals [20]. The images are 128-by-128 pixel grey-scale images. Images include objects such as rubber ducks, toy cars and jars. In total, there are 1440 samples in 16384D.

Figure 8 shows classification accuracies for the previous techniques calculated over the dimensions two to five. The mean and error bars were calculated by leaving three elements out of each class at each round and repeating the runs 24 times, thus going through the whole data. The tolerance for the SDA algorithms was set at 10^{-5}. SDA and RSDA can in average

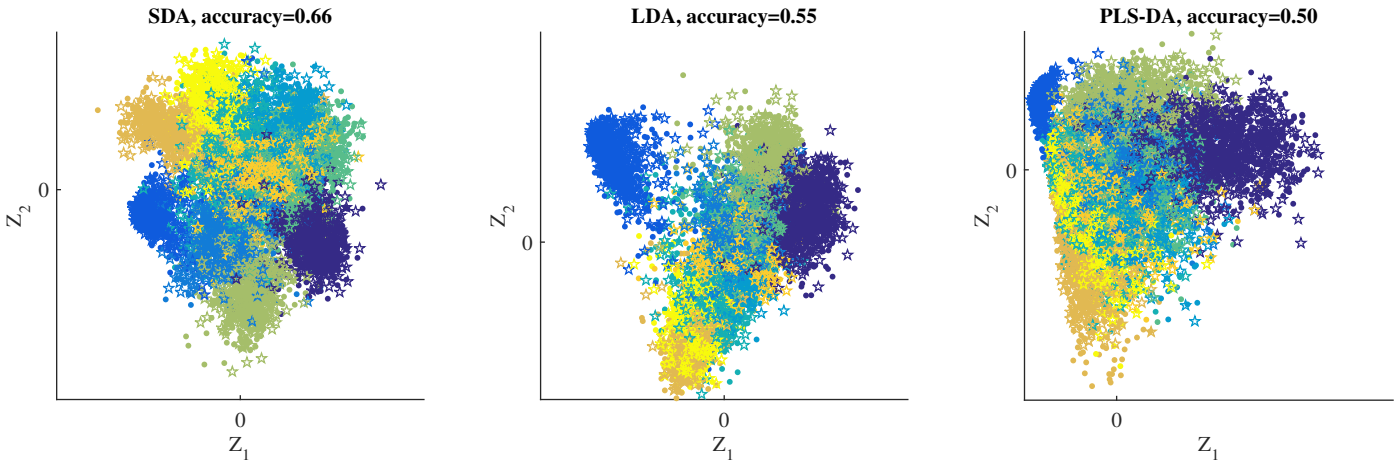

Fig. 7: **Three linear embeddings of the USPS dataset.** Dots denote projected learning points and stars denote projected test points. The 1-NN classification accuracy resulting from this embedding is added to the title.

identify over 90% of the classes in with two variables. At 5D, most algorithms perform similarly. The three best performing embeddings of the COIL-20 dataset are shown in Figure 9.

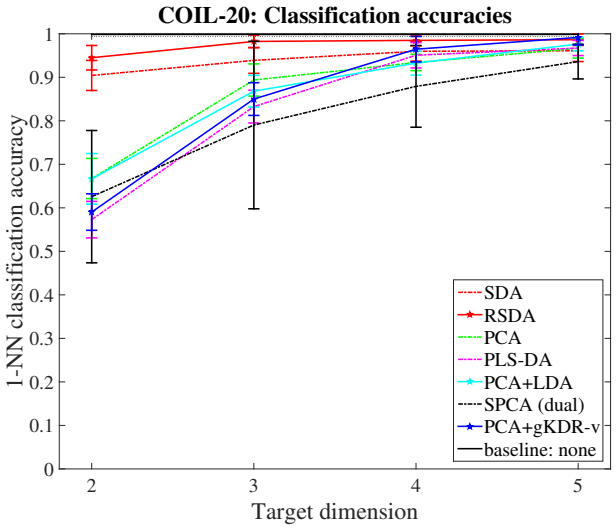

Fig. 8: **COIL-20 dataset.** Classification accuracies for different DR methods. The baseline is the classification accuracy with no DR at all.

4) Computational complexity and runtime comparison: The time complexity of SDA in gradient based methods is largely determined by the number of times the gradient in Equation (6) is evaluated. The matrix expression has the time complexity $O(dn^2+dDn))$, where D is the dimensionality of the input space, d is dimensionality of target space and n is the number of samples. As such, optimizers that require as few function evaluations as possible would be efficient choices.

The processing time of the algorithms in Table II is compared on the three featured datasets in Figure 10. The fastest algorithm differs depending on the characteristics of the dataset. The spectral direction method converges faster and at a lower level than the other algorithms in the USPS dataset. Convergence is reached in ca. 120s. The number of variables is still small enough so that the partial Hessian

Acronym	Method
GD:	Gradient descent [15]
BB:	GD with Barzilai and Borwein step length [15]
CG:	Conjugate gradient (Hestenes-Stiefel update) [15]
PCG:	Preconditioned CG (LBFGS preconditioning)[15]
RCG:	Conjugate gradient (Polak-Ribiere update) [16]
LBFGS:	Limited-memory BFGS [15]
SD:	Spectral direction (Modified Newton's method) [15]

TABLE II: Different optimizers compared.

information can be utilized cost-efficiently. The Olivetti and COIL-20 datasets contain a much larger number of variables. The Hessian matrix is of the size $dD \times dD$, resulting in a costly use of the Hessian information. In COIL-20, the attractive Hessian is re-evaluate only every 20 iterations to increase the performance. We can see that the LBFGS algorithm and different forms of the nonlinear conjugate gradient method are faster choices when doing dimensionality reduction for very high-dimensional spaces.

B. Comparison over multiple datasets

In this subsection we compare the proposed method with state-of-the-art linear embeddings especially for visualization settings (2D). The algorithms were run over three standard UCI datasets [21], three large datasets (more than 4000 data points) and three very high-dimensional datasets (more than 4000 input dimensions). In general, the algorithms were run for different selections of training and test points 10 times to obtain the confidence intervals. The COIL-20 and COIL-100 datasets were evaluated in the principle of leave-*three*-out, as discussed in subsection IV-A3. As a preprocessing step, the original color images in COIL-100 were transformed to gray-scale images and all datasets were normalized. [22] In the tables that follow, a distinction is made between different dimension reduction types: *none*, *supervised* and *unsupervised*. The different types are separated by horizontal lines.

UCI datasets: In the Iris dataset, morphological varieties of the iris flower are identified by quantitative measurements of flowers. In the Wine dataset, wine species are identified

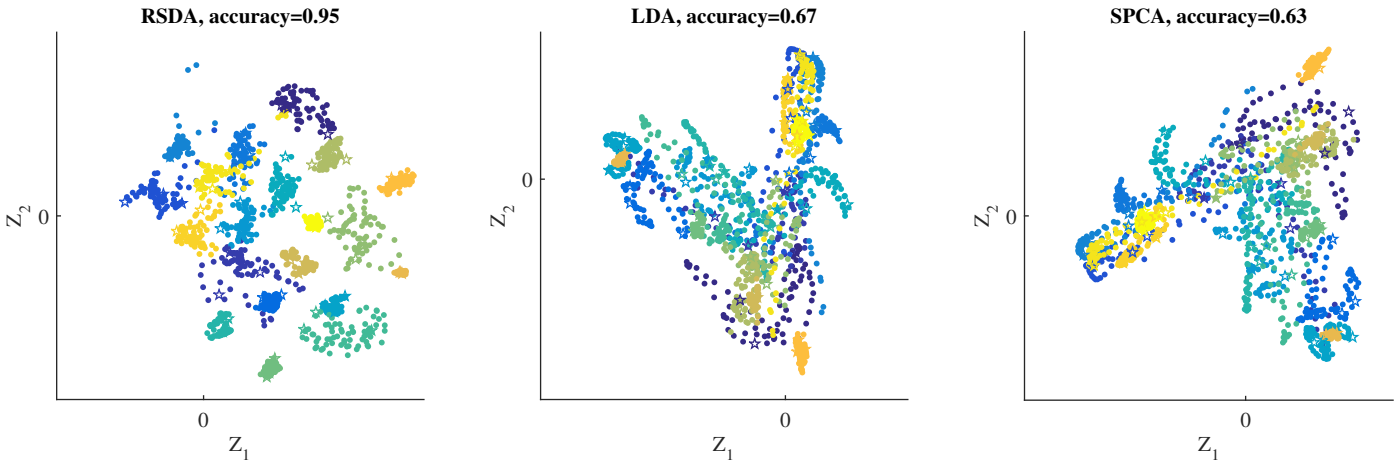

Fig. 9: **Three linear embeddings of the COIL-20 dataset.** Dots denote projected learning points and stars denote projected test points. The 1-NN classification accuracy resulting from this embedding is added to the title.

Method	Iris	Wine	W. Breast Cancer
None	0.941 ± 0.026	0.949 ± 0.026	0.957 ± 0.014
SDA	0.948 ± 0.030	$\mathbf{0.983 \pm 0.017}$	0.956 ± 0.009
LDA	$\mathbf{0.962 \pm 0.025}$	0.981 ± 0.016	$\mathbf{0.961 \pm 0.009}$
PLS-DA	0.879 ± 0.040	0.974 ± 0.021	0.957 ± 0.008
gKDR	0.960 ± 0.021	0.959 ± 0.030	0.956 ± 0.013
SPCA	0.892 ± 0.026	0.974 ± 0.018	$\mathbf{0.961 \pm 0.011}$
KSPCA	0.893 ± 0.047	0.971 ± 0.019	0.893 ± 0.087
PCA	0.860 ± 0.034	0.938 ± 0.024	$\mathbf{0.961 \pm 0.011}$

TABLE III: Three UCI datasets: 1-NN generalization accuracy (mean $\pm$ std) on test set. The datasets were reduced to 2D aside from *None*, in which no dimension reduction was done.

Method	Phoneme	MNIST5k	USPS
None	0.889 ± 0.010	0.936 ± 0.002	0.962 ± 0.002
SDA	$\mathbf{0.875 \pm 0.009}$	$\mathbf{0.557 \pm 0.006}$	$\mathbf{0.668 \pm 0.009}$
LDA	0.664 ± 0.010	$0.461 \pm 0.011^{\text{P}}$	0.554 ± 0.008
PLS-DA	0.779 ± 0.014	0.301 ± 0.006	0.490 ± 0.008
gKDR-v	0.809 ± 0.015	0.323 ± 0.024	0.453 ± 0.009
SPCA	0.780 ± 0.008	0.401 ± 0.008	0.490 ± 0.008
KSPCA	0.781 ± 0.009	0.401 ± 0.009	0.354 ± 0.010
PCA	0.765 ± 0.007	0.383 ± 0.006	0.460 ± 0.010

TABLE IV: Three large high-dimensional datasets, 1-NN generalization accuracy (mean $\pm$ std) on test set. The datasets were reduced to 2D aside from *None*, in which no dimension reduction was done.

based on chemical test results. In the Wisconsin Breast Cancer dataset, tumors are classified as benign or malignant based on physical measurements. The datasets are all standard small datasets with few input dimensions. The results of 2D projections are shown in Table III. In the UCI datasets, all methods are performing quite similarly. The tests were repeated 20 times to obtain the error bars.

Large datasets: Three large datasets were compared. Two datasets were optical number recognition tasks (MNIST, USPS) and one was a phoneme recognition dataset. The phoneme dataset contains two vowel pronunciations (aa,ao) and three consonants (dcl,iy,sh), in which the vowels are difficult to separate [23],[24]. In SDA, the optimality tolerances for the large datasets were set at 10^{-5} and the tests were repeated 10 times each. The results can be seen in Table IV. SDA performs favorably in all tests.

High-dimensional datasets: A face recognition dataset (Olivetti faces) and two object recognition datasets (COIL-20 and COIL-100) were compared. The regularized version of SDA was also calculated. The 1-NN out-of-sample classification accuracies are shown in Table V. The proposed regularized algorithm has the highest accuracy among the tested algorithms on all datasets. The tests were repeated 10 times to obtain the error bars. The tolerance for optimality was set at 10^{-5} in Olivetti and COIL-20 and at 10^{-4} in COIL-100. The tolerances for the regularization search were set at one magnitude higher than the final algorithm, at 10^{-4} resp. 10^{-3}. Optimizing with RSDA, including the λ search procedure, was in average faster than using no regularization ($\lambda = 0$) in COIL-100, with the median time 88 min vs. 215 min[B].

Method	Olivetti faces	COIL-20	COIL-100
None	0.908 ± 0.023	0.999 ± 0.005	0.988 ± 0.006
SDA	0.393 ± 0.056	0.904 ± 0.035	0.277 ± 0.024
RSDA	$\mathbf{0.562 \pm 0.047}$	$\mathbf{0.944 \pm 0.026}$	$\mathbf{0.605 \pm 0.026}$
LDA[P]	$0.446 \pm 0.039^{\text{P}}$	$0.656 \pm 0.079^{\text{P}}$	$0.300 \pm 0.054^{\text{P}}$
PLS-DA	0.310 ± 0.042	0.573 ± 0.042	0.481 ± 0.049
gKDR-v	0.210 ± 0.046	$0.565 \pm 0.057^{\text{P}}$	$0.142 \pm 0.038^{\text{P}}$
SPCA	0.325 ± 0.033	$0.623 \pm 0.152^{\text{D}}$	$0.437 \pm 0.061^{\text{D}}$
KSPCA	0.322 ± 0.037	0.567 ± 0.191	0.397 ± 0.055
PCA	0.289 ± 0.029	0.667 ± 0.046	$0.288 + 0.036$

TABLE V: Three very high-dimensional datasets, 1-NN generalization accuracy (mean $\pm$ std) on test set. The datasets were reduced to 2D aside from *None*, in which no dimension reduction was done.

[D] Dual formulation for SPCA used.
[P] Dimensionality reduction done in a PCA reduced space, with the size 100.
[A] Run on 4-core Intel i5-4570 CPU @ 3.20GHz
[B] Run on 6-core Intel Xeon X5650 @ 2.67GHz. We acknowledge the computational resources provided by Aalto Science-IT project.

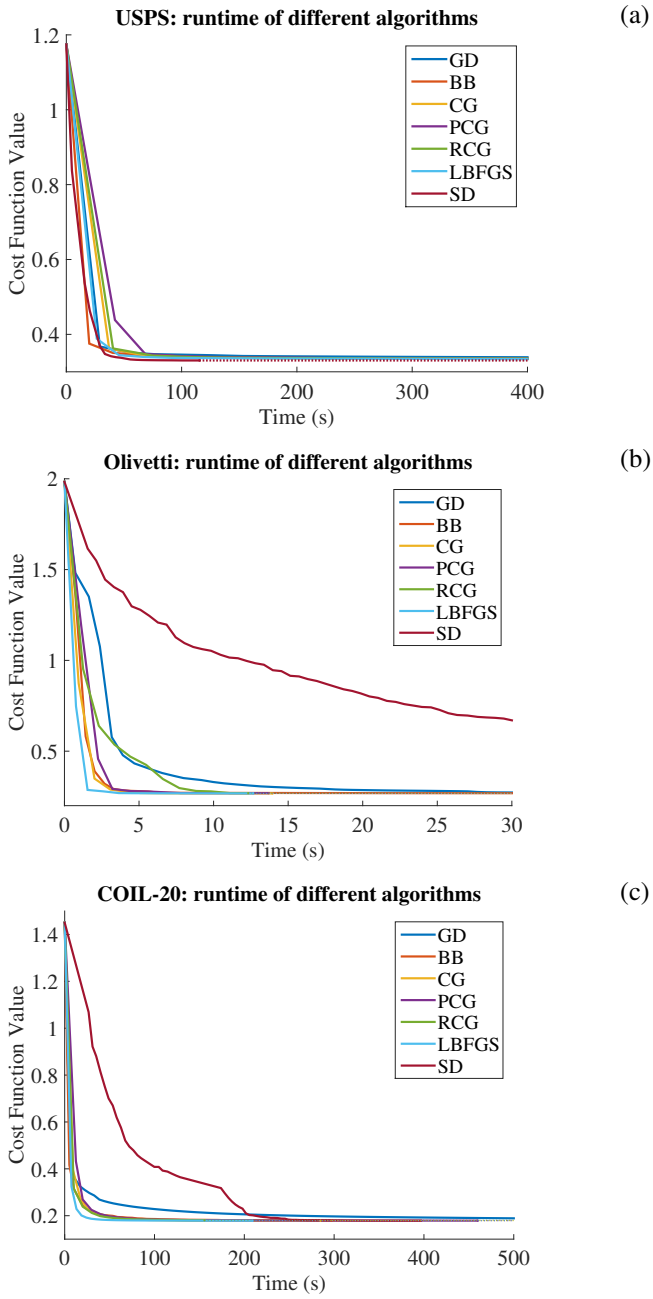

Fig. 10: **Runtimes with different optimization algorithms.** The fastest methods differ depending on the characteristics of the datasets.

V. Conclusion

The proposed method is useful especially at extremely low-dimensional projections of datasets with numerous classes that ordinary discriminant analysis algorithms manage poorly. The generalization ability of the method increases until the optimal structure is found in $\nu - 1$ dimensions, where ν is the number of classes. The method performs better than state-of-the-art linear projections in extremely low-dimensional projections. Tikhonov regularization was found to increase classification accuracies in very high-dimensional datasets.

References

[1] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis-a brief tutorial," *Institute for Signal and information Processing*, 1998.

[2] J. Yang and J. yu Yang, "Why can LDA be performed in PCA transformed space?" *Pattern Recognition*, vol. 36, no. 2, pp. 563 – 566, 2003, biometrics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320302000481

[3] M. Prez-Enciso and M. Tenenhaus, "Prediction of clinical outcome with microarray data: a partial least squares discriminant analysis (pls-da) approach," *Human Genetics*, vol. 112, no. 5-6, pp. 581–592, 2003. [Online]. Available: http://dx.doi.org/10.1007/s00439-003-0921-9

[4] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Kernel dimension reduction in regression," *The Annals of Statistics*, pp. 1871–1905, 2009.

[5] K. P. Adragni and R. D. Cook, "Sufficient dimension reduction and prediction in regression," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1906, pp. 4385–4405, 2009.

[6] K. Fukumizu and C. Leng, "Gradient-based kernel dimension reduction for supervised learning," *arXiv preprint arXiv:1109.0455*, 2011.

[7] E. Barshan, A. Ghodsi, Z. Azimifar, and M. Zolghadri Jahromi, "Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds," *Pattern Recognition*, vol. 44, no. 7, pp. 1357–1371, 2011.

[8] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[9] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.

[10] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski, "Information retrieval perspective to nonlinear dimensionality reduction for data visualization," *The Journal of Machine Learning Research*, vol. 11, pp. 451–490, 2010.

[11] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[12] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[13] "Encyclopedia of mathematics." [Online]. Available: http://www.encyclopediaofmath.org/index.php/Ill-posed_problems

[14] D. P. Bertsekas, "Nonlinear programming," 1999.

[15] M. Schmidt, "Minfunc," 2005. [Online]. Available: "http://www.cs.ubc.ca/ schmidtm/Software/minFunc.html"

[16] C. E. Rasmussen, "Matlab function: Nonlinear conjugate gradient minimizer." [Online]. Available: http://www.gatsby.ucl.ac.uk/ edward/code/minimize/example.html

[17] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming*. Springer, 2008, vol. 116.

[18] M. Vladymyrov and M. Carreira-Perpinan, "Partial-hessian strategies for fast learning of nonlinear embeddings," *arXiv preprint arXiv:1206.4646*, 2012.

[19] "Data for MATLAB hackers." [Online]. Available: http://www.cs.nyu.edu/ roweis/data.html

[20] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-20)," Tech. Rep., Feb 1996.

[21] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[22] S. K. Nayar, S. A. Nene, and H. Murase, "Columbia object image library (COIL-100)," *Department of Comp. Science, Columbia University, Tech. Rep. CUCS-006-96*, 1996.

[23] "Datasets for the elements of statistical learning." [Online]. Available: http://statweb.stanford.edu/ tibs/ElemStatLearn/data.html

[24] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The elements of statistical learning*. Springer, 2009, vol. 2, no. 1.