Sparse Representation Based Projections

Radu Timofte[1]
http://homes.esat.kuleuven.be/~rtimofte

Luc Van Gool[1,2]
Luc.VanGool@esat.kuleuven.be

[1]ESAT-PSI-VISICS / IBBT
Katholieke Universiteit
Leuven, Belgium

[2]BIWI
ETH
Zurich, Switzerland

Abstract

In dimensionality reduction most methods aim at preserving one or a few properties of the original space in the resulting embedding. As our results show, preserving the sparse representation of the signals from the original space in the (lower) dimensional projected space is beneficial for several benchmarks (faces, traffic signs, and handwritten digits). The intuition behind is that taking a sparse representation for the different samples as point of departure highlights the important correlations among the samples that one then wants to exploit to arrive at the final, effective low-dimensional embedding. We explicitly adapt the LPP and LLE techniques to work with the sparse representation criterion and compare to the original methods on the referenced databases, and this for both unsupervised and supervised cases. The improved results corroborate the usefulness of the proposed sparse representation based linear and non-linear projections.

1 Introduction

A large effort has already been spent on techniques to find low-dimensional projections for (very) high dimensional spaces. The latter become widespread nowadays in fields such as computer vision or pattern recognition. Through such projections, additional properties can be enhanced, especially when groundtruth labels are available for the training data.

Taxonomy. The dimensionality reduction techniques can be split into linear and non-linear techniques, according to whether a projection matrix exists or not between the original and the projected space. Each technique aims at preserving or enhancing one or a few properties of the original data in their projection. The best known and most often used linear algorithms are Principal Component Analysis (PCA)[28] (enhances the variance) and Linear Discriminant Analysis (LDA)[12, 18] (enhances the ratio of the between-class scatter and the within-class scatter). Among the non-linear algorithms, Multidimensional Scaling (MDS)[3] preserves pair-wise distances, Locally Linear Embedding (LLE)[19] preserves the neighborhood reconstruction property, Laplacian Eigenmaps (LE)[2] preserves the distances to the nearest neighbors, and Isometric Mapping (ISOMap)[21] works on geodesic distances (instead of pair-wise distances as MDS does) thereby embedding the intrinsic geometry. Starting from these algorithms many variants have been proposed, each one intended to fix some drawbacks. For example, Locality Preserving Projections (LPP)[15] is a linear version

BMVC 2011 http://dx.doi.org/10.5244/C.25.61

of LE, thereby fixing the out-of-sample problem. Another example is the kernelisation of the above methods, widening the dataset that these methods can successfully handle.

Solution frameworks. A solid strand of work provided generic frameworks [9, 24, 26] where many of the projection techniques find a common formulation, when combined with their individual parameter sets. Graph Embedding (GE) [24] is such a framework where the common part starts after providing the similarity matrix for the training data. The graph properties are the ones to be preserved and as such, methods for which the preserved property can be represented with a graph can be formulated in this framework.

Sparsity. Currently, sparse representation methods – also known as l_0 or l_1-norm minimization formulations – gain interest along with the maturization of the compressed sensing field [23]. One basic idea in compressed sensing is that most signals have a sparse representation as a linear combination of a reduced subset of signals from the same space. Naturally the signals tend to have a representation biased towards their own class, i.e. the sparse representation is mainly formed from samples from its own class. This is the starting point for Sparse Representation based Classification (SRC)[23], which proved to be a state-of-the-art method for face recognition.In the machine learning field the sparsity and compressed sensing idea brought new formulations such as Sparse PCA [28] or Sparse Regression Discriminant Analysis (SRDA)[5] which aim at having representations which are sparse over the space directions in the embeddings.

Proposed approach. The sparse representation property of the data in the original space is the one we propose to preserve in this paper. Such a sparse representation highlights the important correlations among the samples that one then wants to exploit to arrive at the final, effective low-dimensional embedding.

In particular, we use linear and non-linear projections with supervised and unsupervised learning to preserve the sparse representations. We adapt the LPP and LLE methods accordingly. We coin these adapted methods Sparse Representation based Linear Projections (SRLP) and Sparse Representation based Embedding (SRE). In the same vein, other methods than LPP and LLE could be adapted as well. Moreover, we compare the classification performance of SRLP and SRE against that achieved with the original LPP and LLE. Several benchmarks are used, incl. face, traffic sign, and handwritten digit recognition.

Recently, it came to our attention that the idea of preserving the sparse representation property has been independently developed already by Huang et al. [16]. They propose a Sparse Reconstruction Embedding method starting from LLE, which is basically identical to the unsupervised SRE method in our work. According to the figures given by these authors, their implementation is very slow even for small datasets, which is a major drawback in the face of the large datasets of our experiments. We propose a solution scheme to mitigate that problem. Moreover, we also present adapted LPP, as well as the supervised and unsupervised versions in both SRE and SRLP.

Structure of the paper. Section 2 formulates the embedding problem, reviews the sparse representation concepts and introduces our SRLP and SRE algorithms. Section 3 gives details on the datasets, classifiers and algorithms, as well as on the results for supervised and unsupervised settings. Conclusions are drawn in Section 4.

2 Sparse Representation Based Projections

In this section we describe the ideas behind sparse representation based projections and the way we modify the linear LPP and non-linear LLE for preserving this property. We use the formulation from [14], where the sparse representation based classifier is presented. First, the sparse problem we solve in this paper is defined in subsection 2.1, second, the sparse representation is introduced in subsection 2.2, third, we introduce Sparse Representation-based Linear Projections (SRLP) and, fourth, Sparse Representation-based Embedding (SRE) in subsections 2.3 and 2.4, respectively.

2.1 Problem Formulation

In an image-based recognition task we have a set of labeled training images $\{\mathbf{x}_i, l_i\}$ from C classes. We assume that the images are roughly aligned. $\{\mathbf{x}_i \in \mathbb{R}^M\}$ is the vectorial representation (either taking the raw pixel values in lexicographical order as in [14] or by extracting other features from the image – M pixel values), while $l_i \in \{1 \dots C\}$ gives the class of the i-th image. We are searching a D-dimensional space such that the corresponding points $\{\mathbf{y}_i \in \mathbb{R}^D\}$ preserve the sparse representation property as defined next. Let $\mathbf{X}_{N\times M} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]^T$, $\mathbf{Y}_{N\times D} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N]^T$, and N be the number of training samples.

2.2 Sparse Representation

In the training set, for each point $\mathbf{x}_i$ we are searching for its sparse representation given by:

$$\text{minimize} \quad \|\mathbf{w}_i\|_0 \;\; \text{subject to} \quad \mathbf{x}_i = \textstyle\sum_{j=1, j\neq i}^{N} \mathbf{w}_{ij}\mathbf{x}_j \tag{1}$$

where $\mathbf{w}_i \in \mathbb{R}^N$ is the sparse vector of weights, $\mathbf{w}_{ij}$ shows the contribution of the sample $\mathbf{x}_j$ to the sparse representation of $\mathbf{x}_i$. This is an l_0-minimization problem, which has been proven to be tractable by minimizing instead the l_1-norm of $\mathbf{w}_i$ if the solution is sparse enough[4]. Moreover, for practical purposes we incorporate the measurement noise $\varepsilon \in \mathbb{R}$ and relax to:

$$\text{minimize} \quad \|\mathbf{w}_i\|_1 \;\; \text{subject to} \quad \|\mathbf{x}_i - \textstyle\sum_{j=1, j\neq i}^{N} \mathbf{w}_{ij}\mathbf{x}_j\|_2 \leq \varepsilon \tag{2}$$

For Compressed Sensing, l_1-minimization proved to be efficient in recovering the sparsest solutions to underdetermined systems of linear equations [4]. Following our experiments and the study from [25], we are using the homotopy [1, 10] method for solving (2). It proved to yield the best trade-off between performance and running time. The reader is referred to [1, 10, 25] for more details. The sparse support is used as stopping criterion with a tolerance of 0.04, similar to the one reported by [23].

In a supervised sparse representation scenario based on labels, we restrict the problem (2) to work within the same class:

$$\text{minimize} \quad \|\mathbf{w}_i\|_1 \;\; \text{subject to} \quad \|\mathbf{x}_i - \textstyle\sum_{j=1, j\neq i, l_i = l_j}^{N} \mathbf{w}_{ij}\mathbf{x}_j\|_2 \leq \varepsilon \tag{3}$$

Also, for an unknown query sample $\mathbf{b} \in \mathbb{R}^N$, we can obtain the sparse representation over the training set:

$$\text{minimize} \quad \|\mathbf{v}\|_1 \;\; \text{subject to} \quad \|\mathbf{b} - \textstyle\sum_{j=1}^{N} \mathbf{v}\mathbf{x}_j\|_2 \leq \varepsilon \tag{4}$$

where $\mathbf{v} \in \mathbb{R}^N$ are the sparse representation weights and we assign the class label l_b from the set of labels L to be:

$$l_b = \arg\max_{c\in L} \sum_{i=1, l_i = c} \|\mathbf{v}_i\|_1 \tag{5}$$

This is the Sparse Representation Classifier[23] decision as used also in our experiments.

2.3 Sparse Representation-based Linear Projections

Locality Preserving Projections (LPP) [15] is a linear approximation of the non-linear Laplacian Eigenmap[2] method, aiming at preserving the local distances to the neighbors. The algorithmic procedure has three steps: constructing the adjacency graph, choosing the weights, and computing the eigenmaps. The result of the first two steps is a symmetric matrix of weights, $\mathbf{W}_{N\times N}$, representing the graph to be embedded in the projection.

Our proposed Sparse Representation-based Linear Projections (SRLP) replaces this graph representation (matrix) with a matrix of weights coming from the sparse representations. The sparse representations $\mathbf{w}_i$ are computed for each training sample $\mathbf{x}_i$, using (2) for the unsupervised case or using (3) for the supervised. The SRLP weighted adjacency matrix is

$$\mathbf{W}_{N\times N} = \max\{[|\mathbf{w}_1|,|\mathbf{w}_2|,\cdots,|\mathbf{w}_N|],[|\mathbf{w}_1|,|\mathbf{w}_2|,\cdots,|\mathbf{w}_N|]^T\} \tag{6}$$

We take absolute values to indicate the importance of the samples in the sparse representations. $\mathbf{W}$ is made symmetric by picking $\max\{\mathbf{w}_{ij},\mathbf{w}_{ji}\}$ for any samples i and j.

The eigenmap step is common to the original LPP and SRLP and consists in computing the eigenvectors and eigenvalues for the generalized eigenvector problem:

$$\mathbf{XLX}^T\mathbf{a} = \lambda\mathbf{XDX}^T\mathbf{a} \tag{7}$$

where $\mathbf{D}$ is a diagonal matrix whose entries are column (or row, since $\mathbf{W}$ is symmetric) sums of $\mathbf{W}$, $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ji}$. $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix. The i^{th} column of matrix $\mathbf{X}$ is $\mathbf{x}_i$.

Let the column vectors $\mathbf{a}_1,\cdots,\mathbf{a}_P$ be the solutions of equation (7), ordered according to their eigenvalues, $\lambda_1 < \cdots < \lambda_P$. Thus, the embedding is as follows:

$$\mathbf{x}_i \rightarrow \mathbf{y}_i = \mathbf{R}^T\mathbf{x}_i, \mathbf{R} = [\mathbf{a}_1,\mathbf{a}_2,\cdots,\mathbf{a}_P], \mathbf{R}\in\mathbb{R}^{M\times P} \tag{8}$$

2.4 Sparse Representation-based Embedding

Locally Linear Embedding (LLE)[19] is representative for non-linear embedding algorithms which preserve the neighborhood reconstruction property. The algorithmic procedure has three steps: finding the neighbors, solving for the reconstruction weights, and computing the embedding. The result of the first two steps is a sparse matrix of weights, $\mathbf{W}_{N\times N}$.

Our Sparse Representation-based Embedding (SRE) replaces this graph representation (matrix) with a matrix of weights coming from the sparse representations. The sparse representations $\mathbf{w}_i$ are computed for each training sample $\mathbf{x}_i$, using (2) for the unsupervised case or using (3) for the supervised. In LLE the row weights $\mathbf{w}_i^T$ are normalized by dividing by the sum of their elements, and so we do for SRE, $\mathbf{w}'_i = \mathbf{w}_i/(\sum_{j=1}^N \mathbf{w}_{ij})$. The SRE weighted adjacency matrix is

$$\mathbf{W}_{N\times N} = [\mathbf{w}'_1,\mathbf{w}'_2,\cdots,\mathbf{w}'_N] \tag{9}$$

The embedding step is common to the original LLE and SRE and consists in minimizing the following cost function in the D-dimensional embedding space:

$$\Phi(\mathbf{Y}) = \sum_{i=1}^{N} \|\mathbf{y}'_i - \sum_{j=1,j\neq i}^{N} \mathbf{w}'_{ij}\mathbf{y}_j\|^2 \tag{10}$$

The solution is obtained by solving the eigenvector problem of a sparse matrix $\mathbf{M} = (\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})$, where $\mathbf{I}$ is the identity matrix of rank N. The embedding coordinates found by LLE and SRE are given by the smallest nonzero eigenvectors. The reader is referred to the original LLE work [19] for more details.

Table 1: Datasets

dataset	total samples	train samples	test samples	dimensionality	# of classes
PIE	11554	8000	3554	784	68
BTSC	7125	4591	2534	784	62
GTSRB	39209	26640	12569	784	43
MNIST	70000	60000	10000	784	10
SwissRoll	2000	2000	2000	3	-

2.5 Relations to Unifying Frameworks

SRLP and SRE could also be formulated in terms of unifying frameworks like Graph Embedding [24] or Patch Alignment (PA) [26]. For SRLP, the graph weights and neighborhood of the LPP derivation should be replaced with the absolute values of the sparse representation coefficients and the nonzero weighted neighbors, resp. In the case of SRE, the neighborhood is given by the sparse representation over the training data instead of the K nearest neighbors as used in LLE. The weights are the sparse representation coefficients. While not explored here, the Manifold Elastic Net (MEN) framework [27] can be seen as an extension of the PA framework with desirable properties such as classification error minimization or over-fitting reduction through an elastic net penalty. SRLP and SRE can be further extended to MEN by using the PA derivations and the extensions from the MEN objective functions.

3 Experimental Results

In this section, we evaluate the performance achieved by preserving the sparse representation in comparison to the original formulations which preserve other properties.

3.1 Data Sets

We use as benchmarking data sets one for face, two for traffic sign and one for handwritten digit recognition, as well as the basic swiss roll for visually assessing the manifold embedding of the methods. Table 1 summarizes their characteristics.

The CMU PIE face database[1] contains 41,368 images for 68 individuals. We use the subset[2] and the training/testing split from [6], containing near frontal poses (C05, C07, C09, C27, C29) under different illuminations and expressions, totaling up to 170 images per subject. The Belgium Traffic Sign Dataset (BelgiumTS)[3][22] contains multiple, calibrated images of streets in Belgium. We pick a subset for classification (BTSC) containing 62 classes as in [22], and follow the training/testing split from the original dataset. The German Traffic Sign Recognition Benchmark (GTSRB)[4] Challenge is held at IJCNN 2011 [20]. This is a multi-class classification challenge, with single images as input. The dataset has 43 traffic sign classes and a total of about $50,000$ images recorded on German streets. The MNIST handwritten digit dataset[5] contains $60,000$ handwritten digits in the training set and $10,000$ handwritten digits in the test set. The images are 28×28 pixel grayscale images where the

[1] http://www.ri.cmu.edu/projects/project_418.html
[2] http://www.zjucadcg.cn/dengcai/Data/FaceData.html
[3] http://homes.esat.kuleuven.be/~rtimofte/
[4] http://benchmark.ini.rub.de/
[5] http://yann.lecun.com/exdb/mnist/

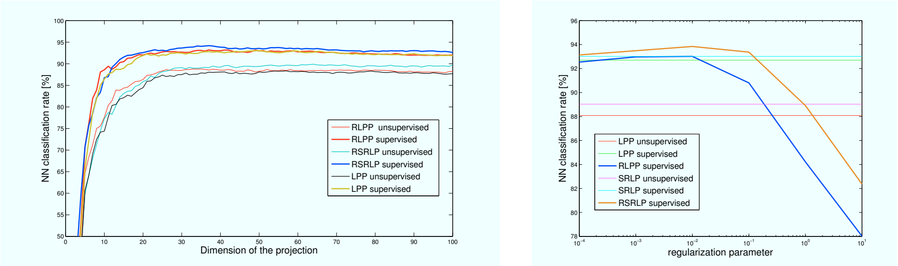

Figure 1: Performance vs. dimensionality (*left*) and regularization parameter (*right*) for different linear projections on BTSC dataset.

center corresponds to the center of mass of the pixels. Also for the other datasets, the images are cropped, resized to 28×28, and the feature vectors are the intensities l_2 normalized. The Swissroll dataset[6] is traditionally used for visually inspecting the 2D embeddings derived for a swissroll shaped 3D set of points.

3.2 Classifiers

We use six classifiers to evaluate the impact of different projection methods. Before using any of those, the projected features are l_2-normalized.

The Nearest Neighbor (NN) classifier assigns the the class to the one of the nearest training sample in the Euclidean sense. The Sparse Representation based Classifier (SRC)[23] uses the sparse representation for getting the weights for the training samples which contribute to the recovery of the unknown sample. The class is the one which sums up the largest contribution in the sparse representation of the sample (see subsection 2.2). The reader is referred to [14] for more details. Support Vector Machines (SVM) classifiers[8] belong to maximum margin linear classifiers and aim at simultaneously minimizing the empirical classification error and maximizing the geometric margin between the classes. This leads to low generalization errors. We use SVM in combination with four standard kernels, as in [17]: $k_{lin}(\mathbf{x},\mathbf{y}) = \mathbf{x}\cdot\mathbf{y}$, $k_{int}(\mathbf{x},\mathbf{y}) = \min(\mathbf{x},\mathbf{y})$, $k_{poly}(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y}+1)^5$, $k_{rbf}(\mathbf{x},\mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|^2)$ and we refer to these SVMs as Linear Kernel SVM (LSVM), Intersection Kernel SVM (IKSVM), Polynomial Kernel SVM (POLYSVM), and Radial Basis Function SVM (RBFSVM), resp. We train one-vs-all classifiers using LIBSVM [7] (with parameters $C = 10$) and LIBLINEAR [11] (with parameters $C = 10$, $B = 10$). The test example is associated with the class with the highest posterior probability estimated from the margin of the test example.

3.3 Algorithm Comparisons

We compare the proposed sparse representation based SRLP (subsection 2.3) and SRE (subsection 2.3), to the original methods, LPP [15] and LLE [19]. We also add PCA and LDA for unsupervised and supervised experiments, resp., and ISOMap and LE for visual inspection (see Fig.2). Moreover, we show the results on the regularized linear versions (namely RLDA,RLPP, and RSRLP), since the regularization improves the performance for the LDA [13] and LPP techniques for appropriate regularization parameters (see Fig. 1).

[6] http://www.cs.nyu.edu/~roweis/lle/code/swissroll.m

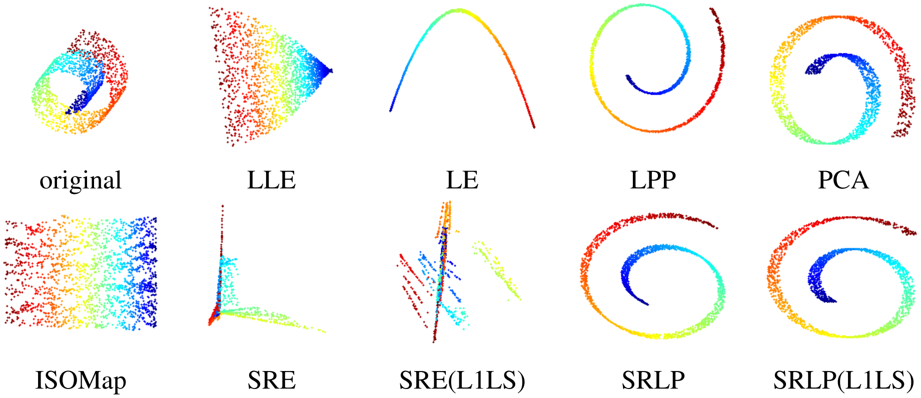

Figure 2: Swiss Roll projections into 2D starting from $N = 2000$ original 3D points.

For all the regularized techniques the weighting parameter is fixed to 10^{-2}. The number of nearest neighbors for the LLE algorithm is set to 12, which was the best in the tests we ran.

We depict the results obtained for RLPP, RSRLP, LLE, SRE, PCA by running each one for a range of subspace dimensions. Some experimental results are not available/computed. RLPP and RSRLP are constrained to projections with dimensionality less than 200 for BTSC, less than 100 for PIE, less than 80 for GTSRB, and less than 50 for MNIST, for reasons of speed. LLE and SRE are constrained to embeddings in smaller than 200-dimensional subspaces.

In LLE, for each new sample the nearest neighbors and the reconstruction weights are found in the original space, and the projection is computed by summing up the projected nearest neighbors multiplied by the reconstruction weights. Similarly, for SRE, the new samples are sparsely represented over the training material and the weights are used in the projection for reconstructing the embedded samples from the training samples' embeddings.

For the supervised SRE we are learning the sparse representation of the training sample from its own class (see subsection 2.4). Similarly, for LLE, the nearest neighbors for each training sample are taken from its own class. Obviously, there is no valuable connection between the classes for both the LLE and SRE methods in the supervised learning case. This is also the case for LPP in the supervised case, where the local neighborhood is taken within the class of each sample. In the supervised case we preserve the properties at the level of each class and ignore the inter-class relations.

3.4 Unsupervised learning results

For unsupervised learning we consider RLPP, LLE, our proposed counterparts RSRLP and SRE, and standard PCA. The number of nearest neighbors for RLPP is empirically fixed to 5 and the weights are given by the heat kernel with $t = 5$ [15].

The projections of the artificial swissroll dataset (see Fig. 2) show no big differences between LPP and its SRLP counterpart. We are getting closer to PCA when SRLP uses the L1LS method[7] instead of a homotopy[8] for solving the l_1-norm minimizations. The L1LS method usually yields solutions with more non-zero elements than the homotopy method,

[7] http://stanford.edu/~boyd/l1_ls/

[8] http://www.eecs.berkeley.edu/~yang/software/l1benchmark/

Table 2: Classification accuracy - unsupervised case (best dimension in brackets)

	classifier	RLPP	RSRLP	LLE	SRE	PCA
BTSC dataset	NN	88.79[33]	89.90[64]	73.64[120]	80.86[140]	78.06[103]
	SRC	91.99[61]	92.90[63]	75.30[200]	82.20[160]	85.64[181]
	LSVM	91.04[191]	91.24[55]	79.44[200]	84.88[200]	90.41[187]
	IKSVM	91.55[96]	92.38[75]	77.56[180]	84.57[160]	89.23[121]
	POLYSVM	90.84[48]	91.16[91]	76.95[200]	82.08[200]	90.02[197]
	RBFSVM	91.36[42]	91.87[98]	78.77[200]	83.35[200]	90.29[160]
PIE dataset	NN	95.75[190]	96.51[101]	89.76[200]	93.53[190]	93.98[198]
	SRC	96.37[99]	97.21[79]	90.57[180]	95.16[200]	96.34[196]
	LSVM	91.39[99]	94.37[97]	87.09[190]	93.33[200]	95.53[194]
	IKSVM	95.53[99]	96.46[99]	89.93[180]	94.63[200]	97.61[196]
	POLYSVM	96.79[99]	97.13[99]	89.59[200]	94.12[200]	97.92[162]
	RBFSVM	95.16[59]	97.19[99]	90.26[200]	94.43[200]	97.92[150]
GTSRB dataset	NN	80.37[70]	84.39[74]	60.95[200]	66.50[170]	64.82[180]
	SRC	85.57[73]	88.89[78]	62.61[200]	68.92[180]	72.66[99]
	LSVM	84.89[78]	85.61[99]	63.34[200]	70.28[190]	84.37[195]
	IKSVM	86.33[63]	87.33[78]	63.48[200]	71.02[200]	82.55[100]
	POLYSVM	90.46[58]	93.09[78]	62.39[200]	70.61[180]	86.85[115]
	RBFSVM	89.42[73]	93.01[78]	63.86[200]	71.33[180]	86.77[100]
MNIST	NN	96.62[42]	96.67[45]	94.39[190]	96.11[130]	97.61[50]
	LSVM	91.26[50]	91.39[50]	95.13[200]	96.82[190]	92.37[186]
	POLYSVM	98.06[50]	98.21[50]	96.01[200]	96.94[150]	98.78[85]

and thus yields more neighbors for the SRE and SRLP embeddings. For the non-linear algorithms, we see adequate projections using ISOMap, LLE, and LE, while SRE has rather unintuitive projections, as expected since it does not preserve local distances. SRE(L1LS) has a more pronounced clustering effect on the projection, caused by the increased number of nonzeros in the sparse representation.

Table 2 shows the best classification accuracies for the unsupervised learning settings, and the corresponding dimensionalities. Unsupervised RSRLP gives a strong improvement for the face and traffic sign datasets over RLPP, for all considered classifiers, while on handwritten digits it is on par with RLPP. SRE gives an even stronger improvement over LLE. Yet, the non-linear SRE and LLE are outperformed by the linear algorithms, even by PCA with sufficient dimensions. Their poor performance needs to be investigated further. Tuning the number of neighbors of LLE for each setting (dimensionality of the projection, dataset, and classifier) could improve performance, but is inefficient and cumbersome. SRE (and SRLP) on the other hand has no such tuning parameter. The running time of SRE is a few seconds for $N = 2000$ on Swissroll, while taking hours for GTSRB.

3.5 Supervised learning results

For the supervised learning experiments we consider the supervised versions of RLPP and LLE, and our RSRLP and SRE, as well as RLDA. Table 3 gives the best achieved classification accuracies along with the dimensionality of the projections. As expected, the supervised algorithms outperform their unsupervised counterparts (see Table 2). Supervised RSRLP im-

Table 3: Classification accuracy[%] - supervised case (best dimension in brackets)

	classifier	RLPP	RSRLP	LLE	SRE	RLDA
BTSC dataset	NN	93.21[37]	94.20[37]	77.94[90]	83.19[100]	92.50[61]
	SRC	95.19[43]	95.54[99]	78.26[120]	83.19[100]	93.17[61]
	LSVM	92.11[29]	92.23[40]	80.82[140]	84.98[100]	90.41[42]
	IKSVM	93.21[76]	93.13[62]	79.16[130]	84.33[100]	90.37[45]
	POLYSVM	93.37[40]	93.25[40]	78.81[110]	83.58[100]	91.20[37]
	RBFSVM	93.61[40]	93.96[40]	79.83[100]	83.74[80]	91.75[45]
PIE dataset	NN	97.72[55]	97.86[37]	92.83[180]	95.98[170]	97.55[51]
	SRC	98.06[53]	98.26[84]	92.74[180]	95.98[180]	97.97[61]
	LSVM	96.03[99]	97.19[95]	91.84[180]	96.12[180]	97.36[61]
	IKSVM	97.41[94]	97.69[63]	92.57[180]	94.54[160]	97.36[65]
	POLYSVM	97.66[79]	98.06[94]	92.88[180]	96.00[160]	97.61[67]
	RBFSVM	97.69[79]	98.01[55]	92.80[200]	95.98[160]	97.61[67]
GTSRB dataset	NN	87.07[48]	91.85[28]	67.13[170]	72.33[50]	92.73[40]
	SRC	90.06[40]	93.64[53]	67.46[200]	71.13[100]	93.56[42]
	LSVM	85.66[82]	87.87[57]	69.25[200]	72.81[170]	87.95[32]
	IKSVM	87.72[78]	89.51[78]	67.99[200]	69.46[200]	87.37[42]
	POLYSVM	92.08[34]	94.79[78]	67.44[200]	72.06[80]	92.63[32]
	RBFSVM	93.51[79]	94.64[78]	68.39[200]	72.11[80]	92.90[42]
MNIST	NN	96.72[48]	96.77[41]	95.48[50]	96.11[130]	90.23[9]
	LSVM	91.25[50]	91.39[50]	96.48[170]	96.76[190]	88.72[9]
	POLYSVM	98.06[50]	98.25[40]	96.76[200]	97.07[160]	92.07[9]

proves over RLPP. SRE improves over the original LLE. The improvements are smaller when compared to the unsupervised case. Again, the non-linear SRE and LLE are outperformed by the linear algorithms.

For the out-of-sample estimation we compute the sparse representation over the training samples. Thus, an SRC decision can be taken. For instance, in the BTSC case, we achieve 85% with SRC in the original space, while after non-linear SRE embedding and out-of-sample estimation, SRC barely reaches 83% with a 200-dimensional embedding, while SRC+PCA at 200 achieves 85.60%. This shows that the non-linear projections (like LLE and SRE) warrant additional research. Sometimes linear algorithms are better, even PCA.

It is easier to improve on GTSRB with sparse representations than on BTSC. This could be due to the fact that BTSC has on average 3 annotations for each physically distinct traffic sign and 62 classes, while GTSRB has 30 annotations and 43 classes. Again, the improvement of sparse representation methods over the locality preserving methods is small for the MNIST dataset that has a large number of training samples and just 10 classes. From these experiments it seems that sparse representations are most effective when the original sampling is neither very sparse nor very dense. With too few original samples, the correlations in the data are difficult to pick up, whereas with many, other methods seem capable of capturing the gist just as well.

4 Conclusions

In this paper, we investigated the idea of preserving the sparse representation of the data in linear and non-linear projections. We start from the graph embedding viewpoint for standard projection techniques and change this graph based on the sparse representation of the signals. The main drawback still is the computational time required for computing the sparse representations for the training data. This can be a few orders of magnitude higher than for other state-of-the-art techniques. Extensive experimental results show that the proposed methods – SRLP and SRE, the modified versions of LPP and LLE, respectively – are on par with or consistently outperform the original formulations in supervised and unsupervised learning settings. The sparse representation property shows great potential and all approaches that admit a graph embedding formulation are amenable to their adaptation.

Acknowledgments. This work has been partly supported by the European Commission FP7-231888-EUROPA project.

References

[1] Muhammad Salman Asif. Primal dual pursuit: A homotopy based algorithm for the dantzig selector. *M.S. Thesis. Georgia Institute of Technology*, 2008.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, 2001.

[3] I. Borg and P. Groenen. *Modern Multidimensional Scaling: theory and applications (2nd ed.)*. Springer-Verlag, 2005.

[4] A.M. Bruckstein, D.L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, February 2009.

[5] Deng Cai. *Spectral Regression: A Regression Framework for Efficient Regularized Subspace Learning*. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, May 2009.

[6] Deng Cai, Xiaofei He, and Jiawei Han. Efficient kernel discriminant analysis via spectral regression. In *Proc. Int. Conf. on Data Mining (ICDM'07)*, 2007.

[7] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[8] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[9] Fernando De la Torre. A least-squares framework for component analysis. *under review in PAMI*, 2011.

[10] David L. Donoho and Yaakov Tsaig. Fast solution of l_1-norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54(11): 4789–4812, 2008.

[11] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008.

[12] R.A. Fisher. The statistical utilization of multiple measurements. *Annals of Eugenics*, 8:376–386, 1938.

[13] J.H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84:165–175, 1989.

[14] Arvind Ganesh, Andrew Wagner, Zihan Zhou, Allen Yang, Yi Ma, and John Wright. Chapter 1. face recognition by sparse representation (accepted). *Cambridge University Press*, 2011.

[15] Xiaofei He and Partha Niyogi. Locality preserving projections. In *NIPS*, 2003.

[16] Shaoli Huang, Cheng Cai, and Yang Zhang. Dimensionality reduction by using sparse reconstruction embedding. In *PCM (2)*, pages 167–178, 2010.

[17] Subhransu Maji and Jitendra Malik. Fast and accurate digit classification. Technical Report UCB/EECS-2009-159, EECS Department, University of California, Berkeley, Nov 2009. URL http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-159.html.

[18] A.M. Martinez and A.C. Kak. PCA versus LDA. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[19] Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. In *IEEE International Conference on Computer Vision*, volume 290, pages 2323–2326, 2001.

[20] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *submitted to International Joint Conference on Neural Networks*, 2011.

[21] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, december 2000.

[22] Radu Timofte, Karel Zimmermann, and Luc Van Gool. Multi-view traffic sign detection, recognition, and 3d localisation. In *Proceedings of the IEEE Computer Society's Workshop on Applications of Computer Vision (WACV)*, Snowbird, Utah, USA, December 2009. IEEE Computer Society.

[23] John Wright, Allen Y. Yang, Arvind Ganesh, Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Learning*, 31(2), February 2009.

[24] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):40 –51, january 2007. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.250598.

[25] A. Yang, A. Ganesh, Z. Zhou, S. Sastry, and Y. Ma. Fast l_1-minimization algorithms and an application in robust face recognition: a review. Technical Report UCB/EECS-2010-13, University of California, Berkeley, 2010.

[26] Tianhao Zhang, Dacheng Tao, Xuelong Li, and Jie Yang. Patch alignment for dimensionality reduction. *IEEE Trans. Knowl. Data Eng.*, 21(9):1299–1313, 2009.

[27] Tianyi Zhou, Dacheng Tao, and Xindong Wu. Manifold elastic net: a unified framework for sparse dimension reduction. *Data Min. Knowl. Discov.*, 22(3):340–371, 2011.

[28] Hui Zou, Trevor Hastie, and Rob Tibshirani. Sparse principal component analysis. *JCGS*, 15:262–286, 2006.