LEARNING COMPRESSED IMAGE CLASSIFICATION FEATURES

Qiang Qiu and Guillermo Sapiro

Duke University,
Durham, NC 27708, USA

ABSTRACT

Learning a transformation-based dimension reduction, thereby compressive, technique for classification is here proposed. High-dimensional data often approximately lie in a union of low-dimensional subspaces. We propose to perform dimension reduction by learning a "fat" linear transformation matrix on subspaces using nuclear norm as the optimization criteria. The learned transformation enables dimension reduction, and, at the same time, restores a low-rank structure for data from the same class and maximizes the separation between different classes, thereby improving classification via learned low-dimensional features. Theoretical and experimental results support the proposed framework, which can be interpreted as learning compressing sensing matrices for classification.

1. INTRODUCTION

Dimensionality reduction plays a significant role in classification tasks. High-dimensional data often have a small intrinsic dimension. Thus, not all measured dimensions are important to represent the underlying models. For example, face images of a subject [1], [2], handwritten images of a digit [3], and trajectories of a moving object [4] can all be well-approximated by a low-dimensional subspace of the high-dimensional ambient space. Dimensionality reduction techniques transform high-dimensional data into a meaningful representation of reduced dimensionality [5].

Various methods have been widely used for dimensionality reduction, such as Principal Components Analysis (PCA) [6], Linear Discriminant Analysis (LDA), Multidimensional scaling (MDS) [7], Isomap [8], and Locally Linear Embedding (LLE) [9]. A survey on dimensionality reduction can be found in [5]. In general, these methods attempt to recover an intrinsic low-dimensional structure by preserving global or local characteristics of the original data in a low-dimensional representation. However, the low-dimensional intrinsic structures are often violated for real-world data. For example, under the assumption of Lambertian reflectance, [1] shows that face images of a subject obtained under a wide variety of lighting conditions can be accurately approximated with a 9-dimensional linear subspace. However, real-world face images are often captured under pose variations; in addition, faces are not perfectly Lambertian, and exhibit cast shadows and specularities [10]. This deviation from ideal low-dimensional models needs to be properly addressed by dimensionality reduction techniques. Moreover, popular dimensionality reduction techniques are task-agnostic, and as such, not necessarily optimal for a given challenge. Both these issues are addressed in this work.

When data from a low-dimensional subspace are arranged as columns of a single matrix, the matrix should be approximately low-rank. Thus, a promising way to handle corrupted underlying structures of realistic data, and as such, deviations from ideal subspaces, is to restore such low-rank structure. In this paper, we propose a dimensionality reduction technique for classification by learning a linear transformation on subspaces using matrix rank, via its nuclear norm convex surrogate, as the optimization criteria. The learned linear transformation not only reduces dimensionality, but also recovers a low-rank structure for data from the same class and increases separations between classes. In other words, the proposed method will transform the original data to low-dimensional subspaces, where intra-class variations are reduced and inter-class separations are increased, for improved classification. Intuitively, the proposed framework shares some of the attributes of LDA, but with a significantly different metric. Our proposed method outperforms LDA, as well as other popular dimensionality reduction methods.

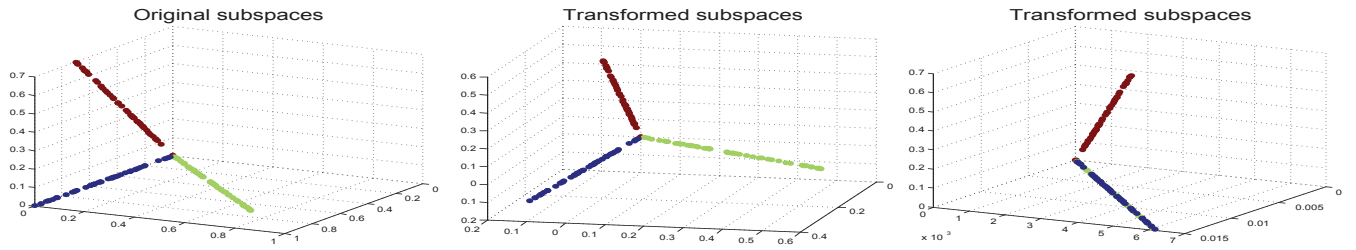
2. LOW-RANK TRANSFORMATIONS FOR DIMENSIONALITY REDUCTION

In this section, we first describe a transformation learning framework using nuclear norm as the optimization criteria. Then we propose to perform dimension reduction using such learned transformation for classification tasks. Section 3 provides further experimental analysis.

2.1. Transformation Learning using Nuclear Norm

Consider data points $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N \subseteq \mathbb{R}^d$, with each data point $\mathbf{y}_i$ in one of the possible low-dimensional subspaces of $\mathbb{R}^d$, and the data arranged as columns of $\mathbf{Y}$. We assume the class labels are known beforehand for training purposes. $\mathbf{Y}_c$ denotes the set of points in the c-th class, points again arranged as columns of the corresponding matrix.

Work partially supported by ONR, NSF, NGA, ARO, and AFOSR.



(a) $\big[\theta_{AB} = 0.79, \quad \theta_{AC} = 0.79, \quad \theta_{BC} = 1.05\big]$
$\mathbf{A}(green), \mathbf{B}(blue), \mathbf{C}(red)$.

(b) $\mathbf{T} = \begin{bmatrix} 0.36 & -0.14 & -0.14 \\ -0.11 & 0.87 & 0.08 \\ -0.11 & 0.08 & 0.87 \end{bmatrix}$;
$\big[\theta_{AB} = 1.57, \quad \theta_{AC} = 1.57, \quad \theta_{BC} = 1.57\big]$.

(c) $\mathbf{T} = \begin{bmatrix} 0.42 & 0.33 & -0.13 \\ 0.39 & 0.32 & -0.16 \\ -0.17 & -0.14 & 0.81 \end{bmatrix}$;
$\big[\theta_{AB} = 0.006, \quad \theta_{AC} = 1.53, \quad \theta_{BC} = 1.53\big]$.

Fig. 1: Learning transformation $\mathbf{T}$ using (1). We denote the angle between subspaces $\mathbf{A}$ and $\mathbf{B}$ as θ_{AB} (and analogous for the other pairs of subspaces). Using (1), we transform subspaces in (a) to (b) and (c) respectively. In (b), we assign three subspaces $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ to three different classes. In (c), we assign subspaces $\mathbf{A}$ and $\mathbf{B}$ to one class, and $\mathbf{C}$ to the other for a two-class scenario. We observe that the learned transformation $\mathbf{T}$ increases the inter-class subspace angle towards the maximum $\frac{\pi}{2}$, and reduces intra-class subspace angle towards the minimum 0.

As data points lie in low-dimensional subspaces, the matrix $\mathbf{Y}_c$ is expected to be *low-rank*. However, as discussed above, this low-rank structure is often violated for real data. We propose in [11] to learn a $d \times d$ linear transformation $\mathbf{T}$,

$$\arg\min_{\mathbf{T}} \sum_{c=1}^{C} ||\mathbf{T}\mathbf{Y}_c||_* - ||\mathbf{T}\mathbf{Y}||_*, \quad s.t. ||\mathbf{T}||_2 = 1. \qquad (1)$$

where $||\cdot||_*$ denotes the matrix nuclear norm, i.e., the sum of the singular values of a matrix. The nuclear norm is the convex envelop of the rank function over the unit ball of matrices [12]. As the nuclear norm can be optimized efficiently, it is often adopted as the best convex approximation of the rank function in the literature on rank optimization (see, e.g., [10] and [13]). The condition $||\mathbf{T}||_2 = 1$ prevents the trivial solution $\mathbf{T} = 0$. Please note our model is not about regularizing the transform T via the nuclear norm, which is common in the literature, but about regularizing the transformed feature space. We proved the following result in [11],

Theorem 1. *Let $\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_C$ be matrices of the same row dimensions, and $[\mathbf{Y_1}, \mathbf{Y_2}, \cdots, \mathbf{Y_C}]$ be the matrix concatenation, we have*

$$||[\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_C]||_* \leq \sum_{c=1}^{C} ||\mathbf{Y}_c||_*,$$

with equality obtained if the column spaces of every pair of matrices $\mathbf{Y}_i, \mathbf{Y}_j, i \neq j$, are orthogonal.

Based on Theorem 1, the proposed objective function (1) reaches the minimum 0 if the column spaces of every pair of matrices are orthogonal after applying the learned transformation $\mathbf{T}$; or equivalently, (1) reaches the minimum 0 when the separation between every pair of subspaces is maximized after transformation, i.e., the smallest principal angle ([14], [15]) between subspaces equals $\frac{\pi}{2}$. Therefore, the linear transformation learned using (1) restores a low-rank structure for data from the same class (due to the first term) and encourages a high-rank structure for data from different classes. In this way, we reduce the variation within the classes and introduce separations between the classes for improved classification.

As discussed in [11], such improved separation is not obtained if the rank function or other (popular) matrix norms, e.g., the induced 2-norm and the Frobenius norm, replace the nuclear norm in (1). Thus, we adopt the nuclear norm in (1) for two major reasons: (a) The nuclear norm is the best convex approximation of the rank function [12], which helps to reduce the variation within classes (first term in (1)); (b) The objective function (1) is in general optimized when the distance between subspaces of different classes is maximized after transformation, which helps to introduce separations between the classes.

We illustrate the properties of the above mentioned learned transformation $\mathbf{T}$ using synthetic examples in Fig. 1. We adopt a simple gradient descent optimization method discussed in [11] to search for the transformation matrix T that minimizes (1). As shown in Fig. 1, the learned transformation $\mathbf{T}$ via (1) increases the inter-class subspace angle towards the maximum $\frac{\pi}{2}$, and reduces intra-class subspace angle towards the minimum 0. Note that Fig. 1c illustrates transformation learning using (1) for a special two-class case, which is exploited later in a binary classification tree.

In [11], we apply learned transformations using (1) for subspace clustering. In clustering tasks, as the data labeling is not known beforehand in practice, the clustering algorithm iterates between two stages: In the first assignment stage, we obtain clusters using any subspace clustering methods, e.g., R-SSC [11]. In the second update stage, based on the current clustering result, we compute the optimal subspace transformation that minimizes (1). The algorithm is repeated until the clustering assignments stop changing. Extensive experiments shows this approach significantly outperforms state-of-the-art methods for subspace clustering.

2.2. Dimensionality Reduction in Classification

Given data $\mathbf{Y} \subseteq \mathbb{R}^d$, so far, we considered a square linear transformation $\mathbf{T}$ of size $d \times d$. If we devise a "fat" linear transformation $\mathbf{T}$ of size $r \times d$, where $r < d$, we enable dimensionality reduction along with discriminative transformation. In Section 3 we will study in detail how the size of reduced dimensionality, i.e., the size of the learned transformation matrix, affects classification. Here we first describe the classifier adopted in this paper, i.e., a classification tree using learned transformations.

A classification tree consists of hierarchically connected *split* (internal) nodes and *leaf* (terminal) nodes. Each split node corresponds to a weak learner, and evaluates each arriving data point and sends it to the left or right child based on the weak learner binary outputs (here we focus on binary trees). Each leaf node stores the statistics of the data points that arrived during training. During testing, a classification tree returns a class posterior probability for a test sample. As discussed in [16], the classification performance in general increases further by employing more trees, with the output defined as the average of tree posteriors. Note that the weak learner associated with each split node plays a crucial role in a classification tree. An analysis of the effect of various popular weak learner models can be found in [16], including decision stumps, general oriented hyperplane learner, and conic section learner. We will use the formulation in (1), with $r < d$, to define such weak classifiers.

We now consider two-class data points, and $\mathbf{Y}_+$ and $\mathbf{Y}_-$ denote the set of points in each of the two classes respectively, points arranged as columns of the corresponding matrix. During training, at the i-th split node, we denote the arriving training samples as $\mathbf{Y}_i^+$ and $\mathbf{Y}_i^-$ (given more than two classes, we randomly divide classes into two categories). We then learn a transformation matrix $\mathbf{T}_i$ using (1), and represent the subspaces of $\mathbf{T}_i\mathbf{Y}_i^+$ and $\mathbf{T}_i\mathbf{Y}_i^-$ as $\mathbf{D}_i^+$ and $\mathbf{D}_i^-$ respectively. The weak learner model at the i-th split node is now defined as $\theta_i = (\mathbf{T}_i, \mathbf{D}_i^+, \mathbf{D}_i^-)$. We denote this weak learner model as *compressing transformation learner* in the paper. During testing, at the i-th split node, each arriving sample $\mathbf{y}$ uses $\mathbf{T_i y}$ as the low-dimensional feature, and is assigned to $\mathbf{D}_i^+$ or $\mathbf{D}_i^-$ that gives the smallest reconstruction error.

Various techniques are available to implement the above compressing transformation learner. In our implementation, we obtain $\mathbf{D}_i^+$ and $\mathbf{D}_i^-$ from $\mathbf{T}_i\mathbf{Y}_i^+$ and $\mathbf{T}_i\mathbf{Y}_i^-$ using the K-SVD method [17], and denote a transformation learner as $\theta_i = (\mathbf{T}_i, \mathbf{D}_i^+(\mathbf{D}_i^+)^\dagger, \mathbf{D}_i^-(\mathbf{D}_i^-)^\dagger)$, where $\mathbf{D}^\dagger = (\mathbf{D^T D})^{-1}\mathbf{D^T}$. The split evaluation of a test sample $\mathbf{y}$, $|\mathbf{T_i y} - \mathbf{D}_i^+(\mathbf{D}_i^+)^\dagger \mathbf{T_i y}|$, only involves matrix multiplication, which is of low computational cost at the testing time.

3. EXPERIMENTAL ANALYSIS

This section presents experimental analysis on the proposed low-rank transformation-based dimensionality reduction method for classification, using the public Extended YaleB face dataset. The Extended YaleB face dataset contains 38 subjects with near frontal pose under 64 lighting conditions. We split the dataset into two halves by randomly selecting 32 lighting conditions for training, and the other half for testing. All the images are resized to 16×16.

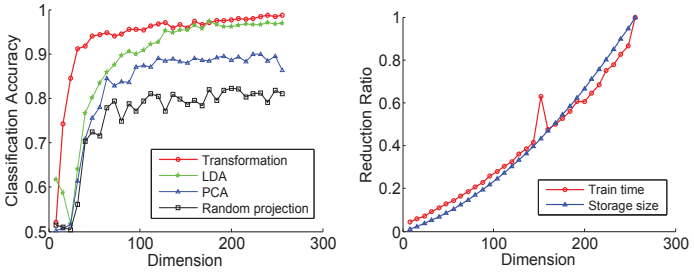
Table 1: Classification accuracies (%) and testing time for the Extended YaleB dataset using classification trees with different learners.

Method	Accuracy (%)	Testing time (s)
Non-tree based methods		
D-KSVD [18]	94.10	-
LC-KSVD [19]	96.70	-
SRC [2]	97.20	-
Classification tree(s)		
Decision stump (1 tree, depth 9)	28.37	0.09
Decision stump (100 trees, depth 9)	91.77	13.62
Conic section (1 tree, depth 9)	8.55	0.05
Conic section (100 trees, depth 9)	78.20	5.04
LDA (1 tree, depth 9)	38.32	0.12
LDA (100 tree, depth 9)	94.98	7.01
SVM (1 tree, depth 9)	95.23	1.62
Identity learner (1 tree, depth 7)	76.89	0.22
Identity learner (1 tree, depth 9)	84.95	0.29
Transformation learner (1 tree, depth 7)	97.70	0.11
Transformation learner (1 tree, depth 9)	**98.77**	0.15

We first demonstrate the capability of a classification tree using the suggested transformation learner (no compression for now) for classification. In Table 1, we first construct classification trees using different learners. For reference purpose, we also include the performance of several non-tree based subspace learning methods, which provide state-of-the-art classification accuracies on this dataset (often at considerably larger computational complexity). Using a single classification tree, the suggested transformation learner already significantly outperforms the popular weak learners *decision stump* and *conic section* [16], where 100 trees are used (30 tries are adopted here). The transformation learner requires comparable testing time as those popular weak learners. We observe that the proposed learner also outperforms more complex split functions such as SVM and LDA. The identity learner denotes a transformation learner but replacing the learned transformation with the identity matrix. Using a single tree with a depth of 9, the proposed approach outperforms state-of-the-art results reported on this dataset. As discussed in [16], the performance in general increases further by employing more trees.

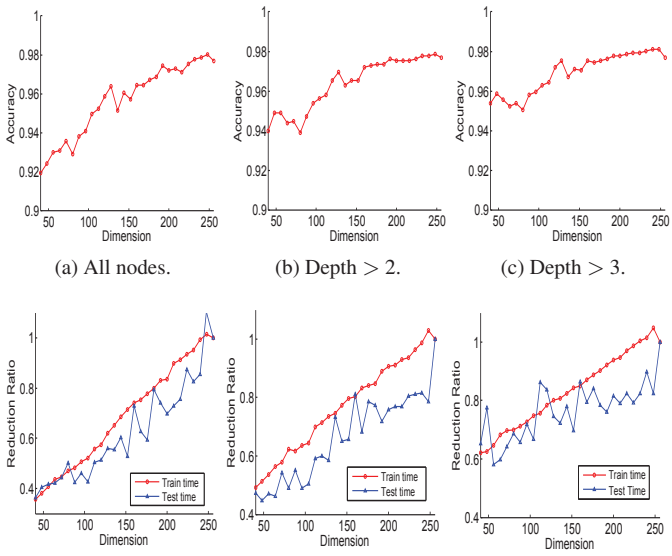
3.1. Compressing Transformation

As images are of the 16×16, without dimension reduction, the learned transformation matrix $\mathbf{T}$ is of size 256×256. We enable dimension reduction (compression) by learning from (1) a compressing transformation of size $r \times 256$, where $r < 256$. We now compare the proposed method with several popular



(a) Classification accuracy. (b) Training time and data size.

Fig. 2: Performance of the compressing transformation learner at the tree *root* node when gradually reducing the data dimensionality. In (a), we illustrate that the proposed technique (*red*) significantly outperforms several popular dimensionality reduction methods in classification accuracy. In (b), we show how the proposed technique enables (almost) linear reduction of the training time and the storage size required by a transformation learner.



(a) All nodes. (b) Depth > 2. (c) Depth > 3.

(d) All nodes. (e) Depth > 2. (f) Depth > 3 .

Fig. 3: Performance of a classification tree by gradually reducing the data dimensionality using the proposed technique. In the first column, dimension reduction is performed at all split nodes. In the second column, reduction is performed at split nodes with depth > 2, i.e., no reduction at the root node and its two child nodes. In the third column, reduction is performed at split nodes with depth > 3. The first row shows the classification accuracy at different reduced dimensions, and the second row shows the nearly linear reduction of the training and testing time for a classification tree using compressing transformation learners.

techniques such as LDA, PCA, and random projection, for dimensionality reduction in classification tasks. We construct as before a classification tree with a depth of 7 to classify the 38 subjects in the Extended YaleB dataset. Fig. 2 shows how dimension reduction affects the transformation learner at the *root* node of the tree (for now, all the other nodes are left full dimension). By gradually reducing the dimensionality r, we notice from Fig. 2a that the proposed compressing transformation-based approach significantly outperforms several popular methods for dimensionality reduction in classification. We observe only a small drop in the classification accuracy before the dimensionality reaches a small value, e.g., $r < 48$. Fig. 2b shows an almost linear reduction of the training time and storage required by a compressing transformation learner, when proportionally reducing dimensionality. At r about 50, we therefore obtain virtually the same classification performance at a fifth of the computational time and storage requirements.

Fig. 3 shows the overall performance of a classification tree using compressing transformation learners with the proposed dimension reduction. In the first column, dimension reduction is performed at all split nodes. In the second column, reduction is performed at split nodes with depth > 2, i.e., no reduction at the root node and its two child nodes. In the third column, reduction is performed at split nodes with depth $>$ 3. With more classes (subjects) present, higher dimensionality is required to represent the underlying structures. Thus, by preserving the full dimension at the root node and its two child nodes, we observe a better balance between reduction and accuracy. For example, as shown in Table 1, without dimension reduction, we obtain classification rate of 97.70% at a testing time of 0.11 sec. In the second column of Fig. 3, we obtain classification rate of 96.96% at a testing time of 0.06 sec, when we reduce data to half of the original dimension, i.e., $r = 128$. Comparing then to other popular weaker learners in Table 1, the proposed compressing transformation learner with dimension reduction performs split evaluations as fast, but at a significantly higher accuracy.

4. CONCLUSION

We introduced a dimensionality reduction technique for classification tasks by learning a "fat" transformation matrix using the nuclear norm as optimization criteria. The learned transformation matrix enables dimensionality reduction, and, at the same time, reduces variations within the classes and increases separations between the classes. We also demonstrated that, while maintaining classification accuracy, the proposed dimension reduction technique significantly reduces the training time, testing time, and storage size required by learners in a classification tree. The learned compressing transform can be seen as learning a compressing classification sensing matrix, and this, incorporating physical sensing constraints, is the subject of current studies.

5. REFERENCES

[1] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, vol. 25, no. 2, pp. 218–233, 2003.

[2] J. Wright, M. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, vol. 31, no. 2, pp. 210–227, 2009.

[3] T. Hastie and P. Y. Simard, "Metrics and models for handwritten character recognition," *Statistical Science*, vol. 13, no. 1, pp. 54–65, 1998.

[4] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *International Journal of Computer Vision*, vol. 9, pp. 137–154, 1992.

[5] L.J.P. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: A comparative review," 2008, Technical Report.

[6] I. Jolliffe, *Principal Component Analysis*, Springer-Verlag, 1986.

[7] T. Cox and M. Cox, *Multidimensional Scaling*, Chapman-Hall, 1994.

[8] J. B. Tenenbaum, V. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[9] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[10] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, June 2011.

[11] Q. Qiu and G. Sapiro, "Learning transformations for clustering and classification," *CoRR*, vol. abs/1309.2074, 2013.

[12] M. Fazel, "Matrix Rank Minimization with Applications," *PhD thesis, Stanford University*, 2002.

[13] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[14] J. Miao and A. Ben-Israel, "On principal angles between subspaces in R^n," *Linear Algebra and its Applications*, vol. 171, no. 0, pp. 81 – 98, 1992.

[15] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 2013, To appear.

[16] A. Criminisi and J. Shotton, *Decision Forests for Computer Vision and Medical Image Analysis*, Springer, 2013.

[17] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[18] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, San Francisco, CA, 2010.

[19] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.*, Colorado springs, CO, 2011.