MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Classification and Boosting with Multiple Collaborative Representations

Chi, Y.; Porikli, F.

TR2013-132 December 2013

Abstract

Recent advances have shown a great potential to explore collaborative representations of test samples in a dictionary composed of training samples from all classes in multi-class recognition including sparse representations. In this paper, we present two multi-class classification algorithms that make use of multiple collaborative representations in their formulations, and demonstrate performance gain of exploring this extra degree of freedom. We first present the Collaborative Representation Optimized Classifier (CROC), which strikes a balance between the nearest-subspace classifier, which assigns a test sample to the class that minimizes the distance between the sample and its principal projection in the selected class, and a Collaborative Representation based Classifier (CRC), which assigns a test sample to the class that minimizes the distance between the sample and its collaborative components. Several well-known classifiers become special cases of CROC under different regularization parameters. We show classification performance can be improved by optimally tuning the regularization parameter through cross validation. We then propose the Collaborative Representation based Boosting (CRBoosting) algorithm, which generalizes the CROC to incorporate multiple collaborative representations. Extensive numerical examples are provided with performance comparisons of different choices of collaborative representations, in particular when the test sample is available via compressive measurements.

IEEE Transactions on Pattern Analysis and Machine Intelligence

Classification and Boosting with Multiple Collaborative Representations

Yuejie Chi, *Member, IEEE,* and Fatih Porikli, *Senior Member, IEEE*

Abstract—Recent advances have shown a great potential to explore collaborative representations of test samples in a dictionary composed of training samples from all classes in multi-class recognition including sparse representations. In this paper, we present two multi-class classification algorithms that make use of multiple collaborative representations in their formulations, and demonstrate performance gain of exploring this extra degree of freedom. We first present the Collaborative Representation Optimized Classifier (CROC), which strikes a balance between the nearest-subspace classifier, which assigns a test sample to the class that minimizes the distance between the sample and its principal projection in the selected class, and a Collaborative Representation based Classifier (CRC), which assigns a test sample to the class that minimizes the distance between the sample and its collaborative components. Several well-known classifiers become special cases of CROC under different regularization parameters. We show classification performance can be improved by optimally tuning the regularization parameter through cross validation. We then propose the Collaborative Representation based Boosting (CRBoosting) algorithm, which generalizes the CROC to incorporate multiple collaborative representations. Extensive numerical examples are provided with performance comparisons of different choices of collaborative representations, in particular when the test sample is available via compressive measurements.

Index Terms—multi-class classification, sparsity, compressive sensing, collaborative representation, boosting.

1 Introduction

Multi-class classification, where the goal is to assign one of several class labels to a test sample, is an important task encountered in many applications and has attracted significant research interests in decades. It is widely used for protein function identification [2], text classification [3], face recognition [4], multi-user detection [5], etc.

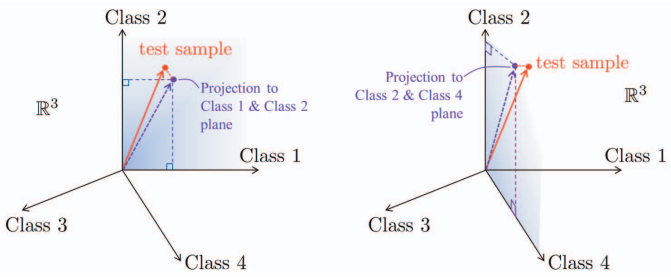
Recent advances in Compressive Sensing (CS) [6], [7] and Sparse Learning [8], [9] have reported significant success in the adoption of sparse representations in signal processing, machine learning and pattern recognition. If a signal can be represented by a few parameters, i.e. admits a sparse representation in certain domain, then it is possible to reconstruct the signal from a much smaller number of linear measurements than its ambient dimension, given that the measurement matrix satisfies certain properties such as restricted isometry properties [7]. Many real-world signals have been shown to possess such representations, for example, an image patch can be regarded as a sparse signal in the wavelet domain. On the other hand, given the compressive measurements of the test samples, it is shown in [10] that the Euclidean distance between samples from different classes are preserved in the compressive domain, enabling the performance of learning and inference tasks without first reconstructing the original samples [11]. This is especially desirable when full data are impossible to obtain due to either power constraints in sensing or unavailability of user information, which commonly arise in recommender systems.

There is also an increasing trend to explore sparsity in the feature domain, in particular for multi-class recognition such as face recognition [12], [13]. Assume that the test sample can be linearly represented by the training samples in the same class. It then admits a sparse representation in the *dictionary* spanned by all training samples from all classes, where most nonzero components are expected to be found in the correct class. By reconstructing the sparse representation using sparse recovery algorithms such as ℓ_1 minimization [14] or greedy pursuits [15], and feeding it into a Sparse Representation based Classifier (SRC) [12], Wright et al. showed that both accuracy and robustness can be improved for face recognition. However, one main drawback of this approach is the computational complexity of acquiring the sparse representations. The computational load of sparse recovery algorithms is still prohibitively high especially when the training set is large. Many works have been steered in this direction including the use of Gabor frame based sparse representations [16], learned dictionary of smaller size instead of the whole training set [17], random hashing [18], etc.

Despite the initial success, there has been a debate whether sparse representations are really necessary. In fact, a test sample has an infinite number of possible representations in the dictionary spanned by the training samples. Since all training samples *collaboratively* form the representation of the test sample, all of these possible

- *Y. Chi is with the Department of Electrical and Computer Engineering and Biomedical Informatics, The Ohio State University, Columbus, OH, 43210. Email: chi@ece.osu.edu.*
- *F. Porikli is with Mitsubishi Electric Research Laboratory, Cambridge, MA, 02189. Email: fatihporikli@ieee.org.*
- *This paper was presented in part at the IEEE Conference on Computer Vision and Pattern Recognition, Rhode Islands, June 2012 [1].*

Digital Object Indentifier 10.1109/TPAMI.2013.236 0162-8828/13/$31.00 © 2013 IEEE



(a) First CR choice (b) Second CR choice

Fig. 1. An illustration of why adopting multiple CRs helps in multi-class classification.

solutions are indiscriminately referred to as Collaborative Representations (CRs) [13], sometimes dictionary representations [19], in the literature. The sparse representation is only one such example. It is argued in [13], [20] that it is not the sparse representation but the adoption of collaborative representations in general play a more crucial role in the success of the SRC. For instance, using a different collaborative representation for the SRC, such as a regularized least-norm representation, similar performance can be achieved with much lower complexity [13]. However, a recent paper [21] suggested that sparse representations are more robust to occlusions and corruptions compared with their non-sparse counterparts.

Rather than trying to settle the debate by claiming which collaborative representation is optimal, we present multi-class classification algorithms that make use of multiple CRs in their formulations, and demonstrate performance gain by leveraging this extra degree of freedom through theoretical analysis and numerical experiments. Fig. 1 intuitively illustrates the concept, where a test sample from Class 2 is decomposed using two different CRs in (a) onto the space of Class 1 and 2, and in (b) onto the space of Class 2 and 4. In both cases the sample has a smaller projection residual to the correct Class 2, but in neither case it provides a large enough margin compared with the projection residual to the wrong class. By combining the two CRs cleverly, we can obtain a stronger confidence in claiming the test sample is indeed from Class 2. This performance gain indicates that it is possible to leverage several easy-to-compute CRs with weaker performance into a classifier with performance comparable to a CR with better performance but difficult to compute.

In this paper, we decompose the multi-class classification problem into two parts, first finding the CRs and then imposing them to a classifier that computes the residual towards each class in order to properly harness the CRs of the test sample. Using the CRs, the test sample is decomposed into a sum of components that each coming from a different subspace, possibly overlapping, spanned by a separate class. In the first half of this paper, we propose a multi-class classifier, called as Collaborative Representation Optimized Classifier (CROC), that achieves optimal combination of the Nearest Subspace Classifier (NSC) [22], which classifies a sample to the class with the minimal residual between the test sample and its principal projection to that class, and the Collaborative Representation based Classifier (CRC), which classifies a sample to the class with the minimal residual between the test sample and its CR components. Under our framework, the well-known SRC and NSC become special cases of CROC under different regularization parameters and particular choices of CRs. The regularization parameter can be optimally tuned via cross validation, which is done at little computational cost. We provide numerical examples to compare the classification performance for sparse and non-sparse CRs, and show in some cases the gain of using sparse representations can be achieved by using a non-sparse representation with an optimally tuned regularization parameter.

Furthermore, we show that the CROC applies a proper weighted combination in the residual domain of a particular CRC and NSC. In practice, it is often challenging to determine the proper rank of the subspace in NSC and which CR to use in CRC. While the success of CROC suggests potential benefits of using multiple CRs in a classifier, it is not straightforward to generalize to the scenario when there are more than two candidate CRCs and NSCs since running cross validation to find the corresponding weights becomes impractical as the number of classifiers gets large.

In the second half of this paper, we introduce the Collaborative Representation based Boosting (CRBoosting) algorithm, which finds a weighted sum of the CRCs and NSCs in the residual domain derived from a set of candidate CRs. The CRBoosting algorithm is inspired by AdaBoost [23], but the key difference is CRBoosting forms the weighted classifier in the residual domain before classification, while AdaBoost forms a weighted classifier in the decision domain. This allows CRBoosting to outperform classifiers using individual CRs. It can also optimize the performance even with only two candidate CRs, which is impossible for AdaBoost. In addition, the presented CRBoosting algorithm elegantly selects the best rank for the NSC and the best CRC to use. We provide performance bounds for the CRBoosting algorithm and present quantitative results to show the advantages of CRBoosting.

The rest of this paper is organized as follows. The multi-class classification problem is described in Section 2. The proposed CROC is presented in Section 3. The CRBoosting algorithm is then proposed in Section 4 to efficiently combine multiple CRCs and NSCs in classification. Numerical examples are given for digit classification and face recognition of proposed algorithms in Section 5. Finally we conclude the paper in Section 6.

A note on notation: we use boldface to denote matrices and vectors. For a matrix $\mathbf{A}$, $\mathbf{A}^T$ denotes its transpose, $\mathbf{A}^\dagger$ denotes its Penrose-Moore pseudo-inverse, $\mathbf{A}^{-1}$ denotes its inverse if exists. $\mathbf{I}_n$ denotes an identity matrix

of dimension n. We summarize the key acronyms and parameters used throughout the paper in Table 1.

TABLE 1
Acronyms and Key Parameters.

Notation	Meaning
CR	Collaborative Representation [13]
CRC	Collaborative Representation based Classifier
SRC	Sparse Representation based Classifier [12]
NSC	Nearest Subspace Classifier [22]
CROC	Collaborative Representation Optimized Classifier
CRBoosting	Collaborative Representation based Boosting
K	number of classes
$\mathbf{A} \in \mathbb{R}^{m\times n}$	training dictionary in the original space
$\mathbf{\Psi} \in \mathbb{R}^{d\times n}$	training dictionary in the feature space
$\mathbf{A}_i \in \mathbb{R}^{m\times n_i}$	training samples of ith class in the original space
$\mathbf{\Psi}_i \in \mathbb{R}^{d\times n_i}$	training samples of ith class in the feature space
$\mathbf{\Phi} \in \mathbb{R}^{d\times m}$	feature selection (measurement) matrix
$\mathbf{y} \in \mathbb{R}^m$	test sample in the original space
$\mathbf{y}_i \in \mathbb{R}^m$	ith CR component in the original space
$\mathbf{z} \in \mathbb{R}^d$	test sample in the feature space
$\mathbf{z}_i \in \mathbb{R}^m$	ith CR component in the feature space
$\mathbf{x}^{CR} \in \mathbb{R}^n$	CR in the original space
$\mathbf{s}^{CR} \in \mathbb{R}^d$	CR in the feature space

2 Multi-Class Classification

Assume there are K classes, where there are n_i training samples from the ith class stacked in a matrix as

$$\mathbf{A}_i = [\mathbf{a}_{i,1}, \cdots, \mathbf{a}_{i,n_i}] \in \mathbb{R}^{m\times n_i},$$

and $\mathbf{a}_{i,j} \in \mathbb{R}^m$ is the jth training sample of dimension m from the ith class. By concatenating all training samples we get the training dictionary

$$\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_K] \in \mathbb{R}^{m\times n},$$

where $n = \sum_{i=1}^{K} n_i$ is the total number of training samples. We are interested in classifying the test sample $\mathbf{y} \in \mathbb{R}^m$, given the labeled training samples in $\mathbf{A}$.

In this paper, the multi-class classification problem is explicitly decomposed into two parts, namely finding the CR of the test sample in the training dictionary, and inputting the CR to a classifier to estimate the label. We will discuss these two parts respectively below.

2.1 Collaborative Representations of Test Samples

We assume that samples within a class lie in the same low-dimensional linear subspace. For example, it is well-established that the face images of the same individual under various illuminations and expressions will approximately span a low-dimensional linear subspace in $\mathbb{R}^m$ [24], [25]. If the test sample $\mathbf{y}$ can be represented as a superposition of training samples in the dictionary $\mathbf{A}$, given in a linear model as

$$\mathbf{y} = \mathbf{A}\mathbf{x}^{CR}, \tag{1}$$

where $\mathbf{x}^{CR} \in \mathbb{R}^n$ is a CR of the test sample by exploring all training samples as a dictionary. When $\mathbf{A}$ is over-determined, i.e. the dimension of the samples is much larger than the number of training samples, the Least-Squares (LS) solution of (1) is given as

$$\mathbf{x}^{LS} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 = \mathbf{A}^{\dagger}\mathbf{y}, \tag{2}$$

where $\dagger$ denotes pseudo-inverse and $\mathbf{A}^{\dagger} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$.

In many cases the LS solution (2) might lead to over-fitting, therefore the test sample is mapped into a low-dimensional feature domain via dimensional reduction methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and random projections first. In face recognition, the features extracted using the above methods are referred to as Eigenface [26], Fisherface [24] and Randomface [12] respectively. Another important argument is motivated by the theory of CS, when it is impossible to acquire the full samples, only a partial observation is available via linear measurements and one is interested in classification using the incomplete information. This can be viewed equivalently as linear feature extraction. In this paper, we focus on linear features, i.e. the extracted features of both the test and the train samples can be written in terms of linear transformation:

$$\mathbf{z} = \mathbf{\Phi}\mathbf{y}; \qquad \mathbf{\Psi} = \mathbf{\Phi}\mathbf{A}, \tag{3}$$

where $\mathbf{\Phi} \in \mathbb{R}^{d\times m}$ is the measurement matrix or linear transformation, d is the feature dimension, $\mathbf{z}$ is the feature of the test sample, and $\mathbf{\Psi} = [\mathbf{\Psi}_1, \mathbf{\Psi}_2, \cdots, \mathbf{\Psi}_K]$ is the feature of the training dictionary. For face recognition, both Eigenface and Randomface are linear features while Fisherface is not. If $\mathbf{\Phi}$ is generated with random i.i.d. entries from Gaussian or uniform distributions, then the low-dimensional features effectively embed the high-dimensional samples by preserving their Euclidean distances up to a small perturbation via the Johnson-Lindenstrauss lemma [27].

Now the test sample in the feature domain can be represented as

$$\mathbf{z} = \mathbf{\Psi}\mathbf{s}^{CR}. \tag{4}$$

where $\mathbf{s}^{CR}$ denotes the CR computed using the extracted features. When the size of the training dictionary is greater than the feature dimension, there are an infinite number of possible representations, we solve the regularized problem below to find the CR,

$$\mathbf{s}^{CR} = \arg\min_{\mathbf{s}} \|\mathbf{z} - \mathbf{\Psi}\mathbf{s}\|_2^2 + \epsilon f(\mathbf{s}), \tag{5}$$

where ϵ is the regularization parameter and $f(\mathbf{s})$ is the regularization function.

The sparse representation is obtained by choosing $f(\mathbf{s}) = \|\mathbf{s}\|_1$, i.e.

$$\mathbf{s}^{L1}(\epsilon) = \arg\min_{\mathbf{s}} \|\mathbf{z} - \mathbf{\Psi}\mathbf{s}\|_2^2 + \epsilon\|\mathbf{s}\|_1, \tag{6}$$

The ℓ_1 constraint is imposed to approximate the ℓ_0 norm, aiming to use a minimal number of training samples to represent the test sample, as it is beneficial in some cases where most of the nonzero entries will come from the correct class, but the complexity is greatly increased.

It is also possible to consider a sparse representation with more structures such as group sparsity [28], where the ith group includes all samples from the ith class. In practice, the regularization parameter ϵ may depend on the noise variance and can be used to control the sparsity level of the CR.

The least-norm representation is obtained by choosing the regularizer as $f(\mathbf{s}) = \|\mathbf{s}\|_2$:

$$\mathbf{s}^{L2}(\epsilon) = \arg\min_{\mathbf{s}} \|\mathbf{z} - \mathbf{\Psi}\mathbf{s}\|_2^2 + \epsilon\|\mathbf{s}\|_2^2, \tag{7}$$

whose solution can be written explicitly as $\mathbf{s}_R^{L2}(\epsilon) = (\mathbf{\Psi}^T\mathbf{\Psi} + \epsilon\mathbf{I}_m)^{-1}\mathbf{\Psi}^T\mathbf{z}$.

The solutions for (6) and (7) can also be computed for the full test sample $\mathbf{y}$ in (1) without dimensionality reduction, denoting their solutions as $\mathbf{x}^{L1}(\epsilon)$ and $\mathbf{x}^{L2}(\epsilon)$.

The test image $\mathbf{y}$ or its reduced-dimension signal $\mathbf{z}$ is represented using all examples from all classes from (1) and (4). Since different classes "collaborate" in the process of forming the representation, $\mathbf{x}$ and $\mathbf{s}$ are considered as collaborative representations. If the CR of the test sample $\mathbf{y}$ is decomposed for each class as $\mathbf{x}^{CR} = [\mathbf{x}_1^{CR}, \cdots, \mathbf{x}_K^{CR}]$, where $\mathbf{x}_i^{CR}$ is the part of coefficients corresponding to the ith class in $\mathbf{x}^{CR}$. Then the test sample can be written as a sum of components from different classes, namely

$$\mathbf{y} = \sum_{i=1}^{K} \mathbf{y}_i, \tag{8}$$

where $\mathbf{y}_i = \mathbf{A}_i\mathbf{x}_i^{CR}$, $1 \le i \le K$ is defined as the ith collaborative representation component. Note this can also be defined for the feature $\mathbf{z}$, given

$$\mathbf{z} = \sum_{i=1}^{K} \mathbf{z}_i,$$

where $\mathbf{z}_i = \mathbf{\Psi}_i\mathbf{s}_i^{CR}$ is the ith collaborative representation component in the feature domain.

2.2 Nearest Subspace Classifier

The Nearest Subspace Classifier (NSC) [22] assigns the test sample $\mathbf{z}$ to the ith class if the projection residual r_i^{NS} from $\mathbf{z}$ to the subspace spanned by the ith training set $\mathbf{\Psi}_i$ is the smallest among all classes, i.e.

$$NSC(\mathbf{z}) = \arg\min_i \ r_i^{NS}. \tag{9}$$

When the number of training samples per class is small so that they do span a subspace, which for face recognition is usually the case, $\mathbf{\Psi}_i$'s are over-determined. Then r_i^{NS} is given as

$$\begin{aligned} r_i^{NS} &= \min_{\mathbf{s}_i} \|\mathbf{z} - \mathbf{\Psi}_i\mathbf{s}_i\|_2^2 && (10)\\ &= \left\|\mathbf{z} - \mathbf{\Psi}_i\mathbf{s}_i^{LS}\right\|_2^2 \\ &= \left\|(\mathbf{I} - \mathbf{\Psi}_i\mathbf{\Psi}_i^{\dagger})\mathbf{z}\right\|_2^2, \quad i = 1, \ldots, K. && (11)\end{aligned}$$

where $\mathbf{s}_i^{LS} = \mathbf{\Psi}_i^{\dagger}\mathbf{z}$, with $\mathbf{\Psi}_i^{\dagger} = (\mathbf{\Psi}_i^T\mathbf{\Psi}_i)^{-1}\mathbf{\Psi}_i^T$.

When the number of training samples is large, such as in digit recognition, $\mathbf{\Psi}_i$'s are under-determined, a principal subspace $\mathbf{B}_i \in \mathbb{R}^{d\times k}$ of rank k for each class is first extracted using PCA to avoid overfitting, then the projection residual $r_i^{NS(k)}$ is computed as

$$\begin{aligned} r_i^{NS(k)} &= \min_{\mathbf{s}_i} \|\mathbf{z} - \mathbf{B}_i\mathbf{s}_i\|_2^2, && (12)\\ &= \left\|\mathbf{z} - \mathbf{B}_i\mathbf{B}_i^T\mathbf{z}\right\|_2^2 \\ &= \left\|\mathbf{z} - \mathbf{\Psi}_i\mathbf{s}_i^{LS(k)}\right\|_2^2, \quad i = 1, \ldots, K. && (13)\end{aligned}$$

where (13) follows from the fact $\mathbf{B}_i\mathbf{\Psi}_i\mathbf{\Psi}_i^{\dagger} = \mathbf{B}_i$, and $\mathbf{\Psi}_i^{\dagger} = \mathbf{\Psi}_i^T(\mathbf{\Psi}_i\mathbf{\Psi}_i^T)^{-1}$, therefore $\mathbf{s}_i^{LS(k)} = \mathbf{\Psi}_i^{\dagger}\mathbf{B}_i\mathbf{B}_i^T\mathbf{z}$.

Strictly speaking, the NSC does not require collaboration of different classes to determine the label, and simply measures the similarity between the test sample and each class without considering the similarities between classes. In practice, the rank k has to be chosen via cross validation or other techniques in order to obtain good performance.

2.3 Collaborative Representation based Classifier

We define the Collaborative Representation based Classifier (CRC) which uses a choice of collaborative representation $\mathbf{s}^{CR} = [\mathbf{s}_1^{CR}, \mathbf{s}_2^{CR}, \cdots, \mathbf{s}_K^{CR}]$ of its feature $\mathbf{z}$ as an input[1], and identifies the test image with the ith class if the residual of the test sample using the i the collaborative representation component, i.e.

$$CRC(\mathbf{z}) = \arg\min_i \ r_i^{CR}, \tag{14}$$

where

$$\begin{aligned} r_i^{CR} &= \|\mathbf{z} - \mathbf{z}_i\|_2^2 \\ &= \left\|\mathbf{z} - \mathbf{\Psi}_i\mathbf{s}_i^{CR}\right\|_2^2, \ 1 \le i \le K \end{aligned} \tag{15}$$

is the smallest for the ith class.

The Sparse Representation based Classifier (SRC) was the first classifier proposed in the form of CRC [12] which uses the sparse representation as an input. In the supplementary material of [12] the authors discussed the benefits of the SRC from a sparse representation viewpoint. If the test image can be sparsely represented by all training images as $\mathbf{x}^{CR} = [0, \cdots, \mathbf{x}_i^{CR}, \cdots, 0]$, such that it can be represented by using only training samples within the correct class, given the abundance availability of training, then the residual for the correct class will be zero while the residual from other classes is the norm of the test image, resulting in maximal discriminative power for classification. In [13] the authors showed that the SRC checks not only the angle between the test image and the partial signal represented by the coefficient on the correct class (which should be small); but also the angle between the partial signal represented by the coefficient on the correct class and that on the rest classes (which should be large).

1. Also applies to the test sample $\mathbf{y}$ in the original space. For consistency of the presentation we adopt the feature space notation.

Although the name SRC indicates this method is designed for sparse representations, it was then quickly adopted in many follow-up works to incorporate other types of collaborative representations. Here, we recast the SRC as a special case of the CRC to avoid ambiguities and unify previous work under the same umbrella. In particular, the CRC using $\mathbf{x}^{LS}$ is adopted in [20], and the CRC using $\mathbf{s}^{L2}(\epsilon)$ is adopted in [13] respectively for face recognition. However, the computational cost of $\mathbf{x}^{LS}$ and $\mathbf{s}^{L2}(\epsilon)$ is much smaller than that of $\mathbf{s}^{L1}$.

3 Collaborative Representation Optimized Classifier (CROC)

In this section we propose a novel optimized classifier, which defines a regularized path of classifiers that connects the NSC and the CRC, where both the SRC and the NSC can be viewed as particular dots on the path.

3.1 Balancing Between NSC and CRC

Given the NSC and the CRC, which look at intra-class residual and inter-class residual respectively, we introduce the Collaborative Representation Optimized Classifier (CROC), which computes a regularized path to study the trade-off between these two classifiers, where

$$CROC(\mathbf{z}, \lambda) = \arg\min_i \; r_i(\lambda), \tag{16}$$

where the residual for each class is calculated as follows

$$r_i(\lambda) = (1-\lambda) r_i^{NS} + \lambda r_i^{CR}, \tag{17}$$

where $0 \leq \lambda \leq 1$. Note that the total weights of both classifiers sum up to 1, since the classification result doesn't change when the residual is scaled. The test sample is then assigned to the class that has the minimal weighted residual. When $\lambda = 0$, the CROC is equivalent to the NSC; and when $\lambda = 1$, the CROC is equivalent to the CRC. In practice, cross-validation may be used to determine the optimal λ, as we will further show in the numerical examples Section 5.1, and better regularization parameter exists to outperform the CRC regardless of the choice of collaborative representations for the test sample.

3.2 When Training Samples are Limited

If all $\mathbf{A}_i$'s are over-determined, the NSC is computed using (10). In this case, the residual of each class for the CRC can be rewritten in the following way:

$$\begin{aligned} r_i^{CR} &= \left\|\mathbf{z} - \mathbf{\Psi}_i \mathbf{s}_i^{CR}\right\|_2^2 \\ &= \left\|\mathbf{z} - \mathbf{\Psi}_i \mathbf{\Psi}_i^\dagger \mathbf{z} + \mathbf{\Psi}_i(\mathbf{\Psi}_i^\dagger \mathbf{z} - \mathbf{s}_i^{CR})\right\|_2^2 \\ &= \left\|(\mathbf{I} - \mathbf{\Psi}_i \mathbf{\Psi}_i^\dagger)\mathbf{z}\right\|_2^2 + \left\|\mathbf{\Psi}_i(\mathbf{\Psi}_i^\dagger \mathbf{z} - \mathbf{s}_i^{CR})\right\|_2^2 \qquad (18) \\ &= \left\|(\mathbf{I} - \mathbf{\Psi}_i \mathbf{\Psi}_i^\dagger)\mathbf{z}\right\|_2^2 + \left\|\mathbf{\Psi}_i(\mathbf{s}_i^{LS} - \mathbf{s}_i^{CR})\right\|_2^2 \\ &\triangleq r_i^{NS} + r_i^{DR}, \qquad (19) \end{aligned}$$

where (18) follows from

$$(\mathbf{I} - \mathbf{\Psi}_i \mathbf{\Psi}_i^\dagger)\mathbf{\Psi}_i = \mathbf{0}, \tag{20}$$

and (19) follows by letting

$$r_i^{DR} = \left\|\mathbf{\Psi}_i(\mathbf{s}_i^{LS} - \mathbf{s}_i^{CR})\right\|_2^2, \tag{21}$$

which measures the residual r_i^{DR} between the ith collaborative representation component of a test sample, and its orthogonal projection within that class. This can be seen as a measure of the difference between signal representations obtained from using only the intra-class information and the one using the inter-class information obtained from the collaborative representation.

Plugging (19) into the residual of the CROC (17), we get

$$r_i(\lambda) = r_i^{NS} + \lambda r_i^{DR}. \tag{22}$$

When the training samples are limited, i.e. $\mathbf{A}_i$'s are over-determined, we could also rewrite the residual error for the CROC by plugging (10) and (21) into (22), given as

$$\begin{aligned} r_i(\lambda) &= \left\|\mathbf{z} - \mathbf{\Psi}_i \mathbf{s}_i^{LS}\right\|_2^2 + \lambda \left\|\mathbf{\Psi}_i(\mathbf{s}_i^{LS} - \mathbf{s}_i^{CR})\right\|_2^2 \\ &= \left\|\mathbf{z} - \mathbf{\Psi}_i \mathbf{s}_i^{LS} + \sqrt{\lambda}\mathbf{\Psi}_i(\mathbf{s}_i^{LS} - \mathbf{s}_i^{CR})\right\|_2^2 \qquad (23) \\ &= \left\|\mathbf{z} - \mathbf{\Psi}_i\left[(1-\sqrt{\lambda})\mathbf{s}_i^{LS} + \sqrt{\lambda}\mathbf{s}_i^{CR}\right]\right\|_2^2 \\ &= \left\|\mathbf{z} - \mathbf{\Psi}_i \tilde{\mathbf{s}}_i\right\|_2^2, \qquad (24) \end{aligned}$$

where (23) follows again from (20), and

$$\tilde{\mathbf{s}}_i = (1-\sqrt{\lambda})\mathbf{s}_i^{LS} + \sqrt{\lambda}\mathbf{s}_i^{CR}.$$

Denote $\tilde{\mathbf{s}}$ as

$$\tilde{\mathbf{s}} = [\tilde{\mathbf{s}}_1, \cdots, \tilde{\mathbf{s}}_K] = (1-\sqrt{\lambda})\mathbf{s}^{LS} + \sqrt{\lambda}\mathbf{s}^{CR},$$

where $\mathbf{s}^{CR}$ is the input CR, $\mathbf{s}^{LS} = [\mathbf{s}_1^{LS}, \cdots, \mathbf{s}_K^{LS}]$ is a combined representation by the LS solution within each class, then $\tilde{\mathbf{s}}$ can be considered as another CR induced by $\mathbf{s}$.

4 Boosting with Multiple CRs

The CROC adopts a weighted combination of the CRC with a particular CR and the NSC with a specific rank, and suggests significant benefits in classification. For instance, in (10) and (12) of NSC, we can define $\mathbf{s}^{LS} = [\mathbf{s}_1^{LS}, \mathbf{s}_2^{LS}, \cdots, \mathbf{s}_K^{LS}]$ and

$$\mathbf{s}^{NS(k)} = [\mathbf{s}_1^{LS(k)}, \mathbf{s}_2^{LS(k)}, \cdots, \mathbf{s}_K^{LS(k)}]$$

as the corresponding LS representation with rank k.

In practice, it is challenging to determine a proper CR for the CRC as well as a proper rank for the NSC without running cross validation or prior knowledge. One can then ask the question that if it is possible to *combine* and *select from* multiple CRCs with different CRs and multiple NSCs with different ranks without running cross validation.

Here, we generalize the CROC approach to combine multiple CRCs and NSCs by formulating a weighted sum of residuals of each class using different CRs and LS representations, and classifying the test sample to the class with the smallest residual. This allows automatic selection of the most suitable representations for different problems. In addition, it provides an alternative way to determine the regularization parameters in (5), by including a set of collaborative representations with different regularization parameters into the candidate set. Below, we introduce an algorithm named Collaborative Representation based Boosting (CRBoosting) to determine the weights in the combined classifier, inspired by AdaBoost [23].

4.1 CRBoosting Algorithm

Consider a candidate set of T collaborative representations or least-squares representations. Let $\mathbf{s}^t$ be the tth representation of interest, $1 \leq t \leq T$. We are interested in finding a weighted classifier $H(\mathbf{z})$ that classifies a sample $\mathbf{z}$ with the class having the minimal weighted residual $r_i(\boldsymbol{\alpha})$, given as

$$H(\mathbf{z}) = \arg\min_i r_i(\boldsymbol{\alpha}),$$

where

$$r_i(\boldsymbol{\alpha}) = \sum_{t=1}^{T} \alpha_t \|\mathbf{z} - \boldsymbol{\Psi}_i \mathbf{s}_i^t\|_2^2, \tag{25}$$

$\alpha = [\alpha_1, \cdots, \alpha_T]$ is a positive vector, and without loss of generality, $\mathbf{1}^T\boldsymbol{\alpha} = \sum_{t=1}^{T} \alpha_t = 1$. In comparison, the AdaBoost algorithm finds a weighted classifier in the decision domain, given as

$$H^*(\mathbf{z}) = \arg \min_{1 \leq i \leq K} \sum_{t=1}^{T} \alpha_t h_i^t(\mathbf{z}), \tag{26}$$

where $h_i^t(\mathbf{z}) = 1$ if $\mathbf{z}$ is classified to the ith class by the tth CRC, and $h_i^t(\mathbf{z}) = 0$ otherwise.

With two candidate representations, the solution to AdaBoost is equal to the candidate with the larger weight. To spell out our advantage, the proposed CRBoosting classifier becomes the CROC and constructs an optimized classifier with better accuracy, as shown in Sec 5.3.

The CRBoosting weights α_t's are learned on a validation set via the proposed algorithm as summarized in Algorithm 1. It is straightforward to see $|\epsilon_t| < b_t$ for every t. The iterations stop if $\alpha_t = 0$ for certain t. In simulations, we run the algorithm with more iterations to allow refinement of weight estimation.

It is worth noting that $d_{\ell,t}$ measures the difference between the residual of the correct class and the minimal residual of the rest of the classes of the ℓth sample. Therefore $d_{\ell,t} < 0$ if it is labeled correctly by the ith CRC, and $d_{\ell,t} > 0$ otherwise. The CRBoosting algorithm thus update the distribution $D_{t+1}(\ell)$ by putting more weights on the tth classifier if it is incorrect.

Algorithm 1 CRBoosting

1: Input: training set $\mathbf{A}$, validation set $\mathbf{Y} = [\mathbf{y}_\ell] \in \mathbb{R}^{m \times L}$ and their labels c_ℓ, and the measurement matrix $\boldsymbol{\Phi}$;
2: Compute the measurements or features: $\mathbf{Z} = \boldsymbol{\Phi}\mathbf{Y}$, $\boldsymbol{\Psi} = \boldsymbol{\Phi}\mathbf{A}$;
3: Initialize the distribution $D_1(\ell) = 1/L$.
4: **for** $t = 1 \to N$ **do**
5: Find the representation on the distribution D_t to maximize $|\epsilon_t / b_t|$, where

$$\epsilon_t = \mathbb{E}_{D_t}[d_{\ell,t}] \quad \text{and } b_t = \max_\ell |d_{\ell,t}|, \tag{27}$$

and

$$d_{\ell,t} = \|\mathbf{z}_\ell - \boldsymbol{\Psi}_{c_\ell} \mathbf{s}_{c_\ell}^t\|_2^2 - \min_{i \neq c_\ell} \|\mathbf{z}_\ell - \boldsymbol{\Psi}_i \mathbf{s}_i^t\|_2^2.$$

6: Choose $\alpha_t \in \mathbb{R}$ as:

$$\alpha_t = \max\left\{ \frac{1}{2b_t} \log\left(\frac{b_t - \epsilon_t}{b_t + \epsilon_t} \right), 0 \right\};$$

7: Update D_{t+1}:

$$D_{t+1}(\ell) = \frac{D_t(\ell)}{Z_t} e^{\alpha_t d_{\ell,t}},$$

where Z_t is the normalization factor.
8: **end for**
9: Output: The weights $\{\alpha_t\}_{t=1}^T$ after normalization.

4.2 Classification Error on Validation Set

Similar to AdaBoost, the training error on the validation set is bounded by $\prod_{t=1}^{T} Z_t$ as the theorem below.

Theorem 4.1. *The validation error with respect to the initial distribution D_1 is bounded by*

$$\mathbb{P}_{D_1}(H) = \frac{1}{L} \sum_{\ell=1}^{L} 1\{H(\mathbf{z}_\ell) \neq c_\ell\} \leq \prod_{t=1}^{T} Z_t, \tag{31}$$

where $\mathbb{1}\{\mathcal{P}\}$ is the indicator function of an event $\mathcal{P}$.

Proof: The error on the validation set can be bounded by (28)-(30), where (29) follows from

$$\min_{i \neq c_\ell} \sum_{t=1}^{T} \alpha_t \|\mathbf{z}_\ell - \boldsymbol{\Psi}_i \mathbf{s}_i^t\|_2^2 \geq \sum_{t=1}^{T} \alpha_t \min_{i \neq c_\ell} \|\mathbf{z}_\ell - \boldsymbol{\Psi}_i \mathbf{s}_i^t\|_2^2,$$

and (30) follows from $\mathbb{1}\{a > b\} \leq \exp\{a - b\}$ for any a an b. Now we unwrap the distribution $D_T(\ell)$ as

$$D_T(\ell) = \frac{D_1(\ell)}{\prod_{t=1}^{T} Z_t} \cdot \exp\left\{ \sum_{t=1}^{T} \alpha_t d_{\ell,t} \right\},$$

and plug this and $D_1(\ell) = 1/L$ into (30), we get

$$\mathbb{P}_{D_1}(H(\mathbf{z}_\ell) \neq c_\ell) \leq \sum_{\ell=1}^{L} D_T(\ell) \cdot \prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} Z_t.$$

□

$$\mathbb{P}_{D_1}(H(\mathbf{z}_\ell) \neq c_\ell) = \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{1}\left\{ c_\ell \neq \arg\min_i \sum_{t=1}^{T} \alpha_t \|\mathbf{z}_\ell - \mathbf{\Psi}_i \mathbf{s}_i^t\|_2^2 \right\} \tag{28}$$

$$= \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{1}\left\{ \sum_{t=1}^{T} \alpha_t \|\mathbf{z}_\ell - \mathbf{\Psi}_{c_\ell} \mathbf{s}_{c_\ell}^t\|_2^2 \geq \min_{i\neq c_\ell} \sum_{t=1}^{T} \alpha_t \|\mathbf{z}_\ell - \mathbf{\Psi}_i \mathbf{s}_i^t\|_2^2 \right\}$$

$$\leq \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{1}\left\{ \sum_{t=1}^{T} \alpha_t \|\mathbf{z}_\ell - \mathbf{\Psi}_{c_\ell} \mathbf{s}_{c_\ell}^t\|_2^2 \geq \sum_{t=1}^{T} \alpha_t \min_{i\neq c_\ell} \|\mathbf{z}_\ell - \mathbf{\Psi}_i \mathbf{s}_i^t\|_2^2 \right\} \tag{29}$$

$$\leq \frac{1}{L}\sum_{\ell=1}^{L} \exp\left\{ \sum_{t=1}^{T} \alpha_t (\|\mathbf{z}_\ell - \mathbf{\Psi}_{c_\ell} \mathbf{s}_{c_\ell}^t\|_2^2 - \min_{i\neq c_\ell} \|\mathbf{z}_\ell - \mathbf{\Psi}_i \mathbf{s}_i^t\|_2^2) \right\} \tag{30}$$

4.3 Choosing CRC and α_t

From Theorem 4.1, we would like to select α_t's to minimize $\prod_{t=1}^{T} Z_t$. For the tth CRC, Z_t can be written as $Z_t = \mathbb{E}_{D_t}\left[e^{\alpha_t d_{\ell,t}}\right]$. Exact minimization of Z_t is difficult and we seek tractable approximations to minimize Z_t. Since the function $e^{\alpha r}$ is convex in r and any constant $\alpha \in \mathbb{R}$, if $r \in [-b, b]$, the following upper bound holds

$$e^{\alpha r} \leq e^{-b\alpha} \cdot \frac{b-r}{2b} + e^{b\alpha} \cdot \frac{r+b}{2b}, \tag{32}$$

let $b_t = \max_\ell |d_{\ell,t}|$, then Z_t is upper bounded by

$$Z_t \leq \frac{e^{-b_t\alpha_t} + e^{b\alpha_t}}{2} + \frac{e^{b_t\alpha_t} - e^{-b_t\alpha_t}}{2b_t}\epsilon_t, \tag{33}$$

where $\epsilon_t = \mathbb{E}_{D_t}[d_{\ell,t}]$. Then α_t can be chosen to minimize the upper bound of Z_t. By zero-forcing the derivative of the RHS of (33), and α_t is nonnegative, we get

$$\alpha_t = \max\left\{ \frac{1}{2b_t} \log\left(\frac{b_t - \epsilon_t}{b_t + \epsilon_t} \right), 0 \right\}, \tag{34}$$

which corresponds to $Z_t \leq \sqrt{1 - \epsilon_t^2/b_t^2} < 1$. From (33) it is straightforward to choose the tth CRC that minimize the upper bound of Z_t, i.e. maximize $|\epsilon_t/b_t|$. This choice is analogous to [29] for a probabilistic output in $[0, 1]$.

An alternative method is to consider the approximation of $e^{\alpha r}$ by the second-order Taylor expansion, as

$$e^{\alpha r} \approx 1 + \alpha r + \frac{1}{2}\alpha^2 r^2,$$

where $\alpha > 0$. Then Z_t is approximated by

$$Z_t \approx \mathbb{E}_{D_t}[1 + \alpha_t d_{\ell,t} + \frac{1}{2}\alpha_t^2 d_{\ell,t}^2] = 1 + \alpha_t\epsilon_t + \frac{1}{2}\alpha_t^2\beta_t, \tag{35}$$

where $\beta_t = \mathbb{E}_{D_t}[d_{\ell,t}^2]$. Then α_t can be chosen to minimize the RHS of (35), given

$$\alpha_t = \max\left\{ -\frac{\epsilon_t}{\beta_t}, 0 \right\},$$

and correspondingly $Z_t \approx 1 - \epsilon_t^2/\beta_t$. In this case the tth CRC should be chosen to minimize the approximation of Z_t in (35), i.e. to maximize ϵ_t^2/β_t. We refer to the CRBoosting algorithm associated with this update rule as CRBoosting-T.

Both updating rules (CRBoosting) and (CRBoosting-T) are heuristic and do not minimize exactly Z_t. In particular, the update rule of CRBoosting is based on the assumption that $d_{\ell,t}$ is bounded in an interval $[-b_t, b_t]$ and the bound can be calculated in the algorithm. If b_t is not very large, it is possible to find a good upper bound of $e^{\alpha r}$ and to choose α_t that minimizes this upper bound. On the other hand, if $d_{\ell,t}$ is not suitably bounded in a small interval $[-b_t, b_t]$, it is still possible to approximate $e^{\alpha r}$ by its second-order Taylor expansion, and to choose α_t that minimizes the Taylor expansion.

5 Numerical Results

We present numerical results on digit recognition and face recognition to show the classification accuracy gain by optimally choosing the regularization parameter. For digit recognition, the number of training images per class is very high, corresponding to the case $\mathbf{A}_i$ is under-determined; for face recognition, the number of training images per class is usually small, corresponding to the case $\mathbf{A}_i$ is over-determined. Finally, we provide performance of the CRBoosting algorithm. Throughout the session, the ℓ_1 minimization algorithm is implemented using the CVX toolbox [30].

Note that, the CROC and CRBoosting apply to any multi-class classification and object recognition problem that is formulated in a vector space. In the following examples our goal is not to report the best possible results, which may be obtained by selecting database specific features, using part-based representations, learning distance and alignment manifolds, etc., but to prove that a much better classification performance can be achieved by balancing the contributions of different intra-class and inter-class representations. Thus, we use *simple intensity features* to report the most objective comparative evaluations between the existing multi-class classification schemes and our methods.

The computational complexity of the CROC and CRBoosting algorithms depends on the candidate classifiers. However, if we assume a set of candidate CRCs are run a priori, the additional complexity of CROC and CRBoosting is very small.

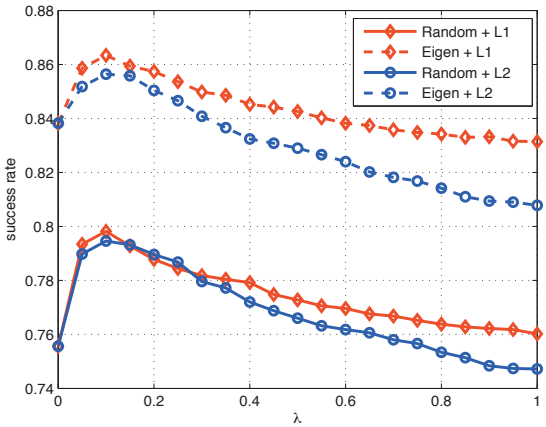


Fig. 2. Classification results of the CROC shown as a regularization path using partial measurements from random projection and eigenvector projections for the MNIST digits database.

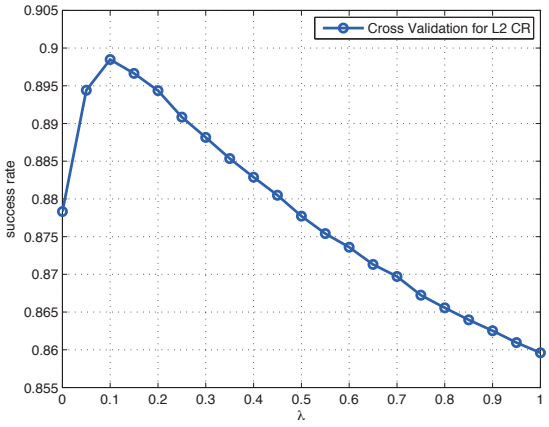


Fig. 3. Cross-validation for regularization parameter λ using least-norm CRC from eigenvector projections for the MNIST digits database.

5.1 Digit Recognition using CROC

The MNIST Handwritten Digits database [31] is used to test the proposed multi-class classification algorithm. There are about 6000 training examples and 1000 test examples of each class in the data set. Each image is an 8-bit gray-scale image of "0" through "9" of dimension $m = 28 \times 28$.

We consider a toy example where only $n_i = 50$ training examples is provided per class, and the number of test examples per class is $n_i = 500$. We make $d = 80$ measurements of each test sample, and the whole test image is assumed unknown. We test the CROC against different regularization parameters, with $\lambda \in [0, 1]$.

In the case where the full sample is not known, we could make partial observations using either random projections using compressive sensing or projection along the eigenvector directions. Figure 2 shows the classification accuracy for both scenarios using sparse (L1) and least-norm (L2) CRCs. Projections using eigenvectors achieve better result than random projections in terms of accuracy. When $\lambda = 1$, the SRC achieves slightly better result than the least-norm CRC using random projections, and this gain is even larger using eigenvector projections. However, a better classification can be achieved with λ around 0.1 for both CRCs who has very small performance gap between sparse and least-norm CRCs. Table 2 further summarizes the classification results for comparison. The optimal λ can be obtained by performing cross-validation on randomly selected training examples and testing examples for a few times, and compute the average classification accuracy for different λ and choose the optimal one. Figure 3 shows the average classification accuracy over 5 different partitions for the least-norm CRC using eigenvector projections, showing the optimal $\lambda = 0.1$ in this case.

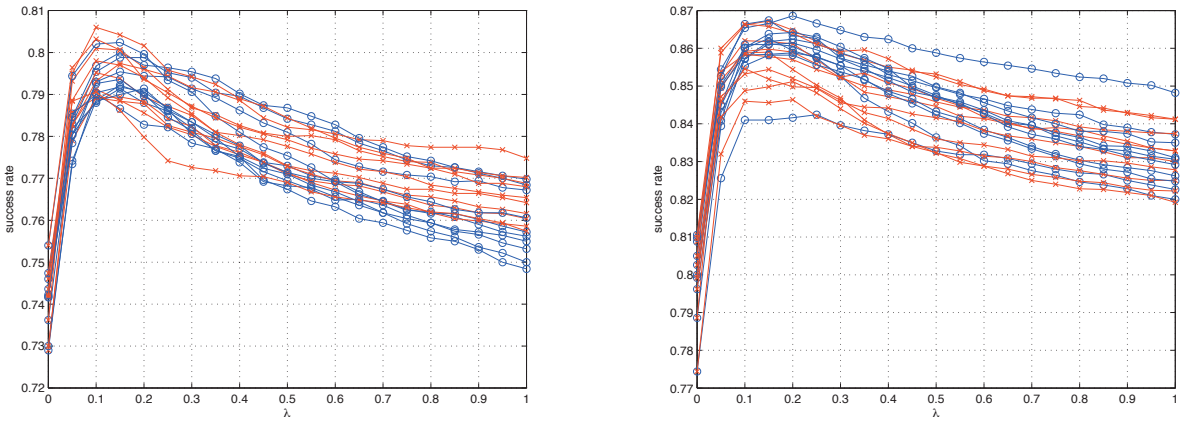
TABLE 2
Classification results of the NSC, CRC and CROC using partial measurements from random projections and eigenvector projections respectively.

Scenario	NSC	CRC	CROC ($\lambda = 0.1$)
Random+L1[%]	75.56	76.02	**79.82**
Random+L2[%]	75.56	74.72	**79.46**
Eigen+L1[%]	83.82	83.14	**86.34**
Eigen+L2[%]	83.82	80.78	**85.64**

We also examine the effect of different realizations of measurement matrices $\mathbf{\Phi}$ on the regulation path of CROC, where the entries of $\mathbf{\Phi}$ are all generated with standard Gaussian random variables. Figure 4 (a) shows the regularization paths of 10 realizations of $\mathbf{\Phi}$. We also resample the training and test samples with the same size 10 times, and Figure 4 (b) shows the regularization paths of each realizations. In both figures the circled lines correspond to least-norm representations, and the crossed lines correspond to sparse representations. The regularization paths all possess similar trends, where the optimal λ tends be the around 0.1. This indicates the validity of the proposed CROC is robust towards different choices of measurement matrices and training sets. On the other hand, the performance of CROC, and all other classifiers varied slightly between different $\mathbf{\Phi}$'s, and it is useful to optimize for a good measurement matrix for dimension reduction.

Figure 5 exemplifies how the CROC outperforms both the NSC and the CRC by using the least-norm CR. Each row shows the classifier residual using the NSC, the CRC and the CROC when $\lambda = 0.1$ respectively. For two test examples of digit "0": in (a) it is correctly classified by the NSC, but the CRC misclassifies it as digit "8"; while in (b) it is correctly classified by the CRC, but the NSC misclassifies it as digit "2". However, both can be correctly identified as "0" using a properly regularized CROC.

If we increase the number of training samples per class



(a) Different random measurements (b) Different training/test samples

Fig. 4. Classification results of the CROC shown as a regularization path using partial measurements from (a) 10 different realizations of random projections, and (b) 10 different resamplings of training and test samples, for the MNIST digits database. The circled lines use least-norm representations and the crossed lines use sparse representations.

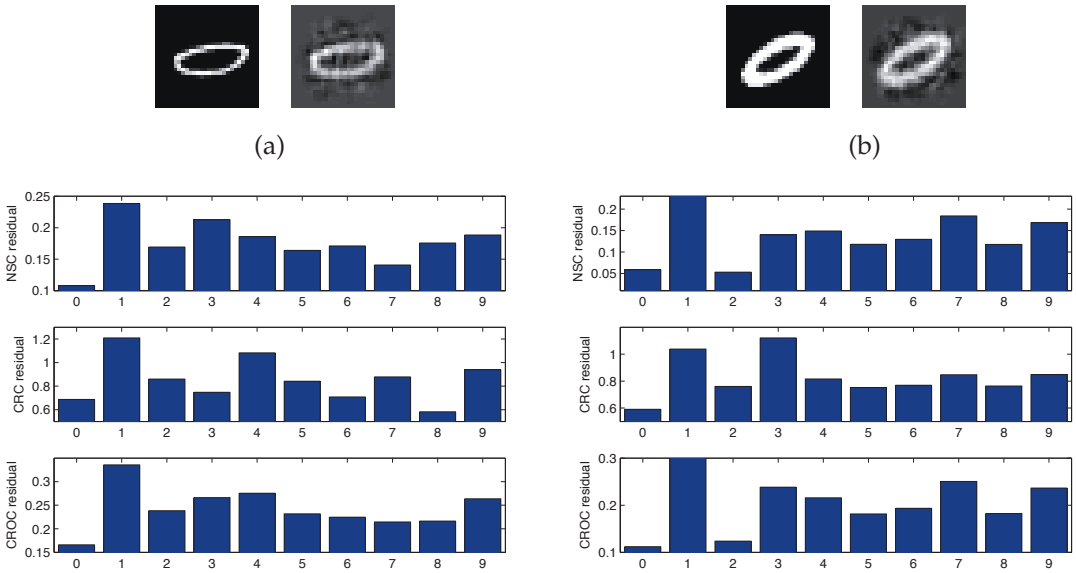


Fig. 5. Classifier residual for two examples of digit "0": (a) and (b) show the original digit and its reconstruction from $d = 80$ eigenvector projections; (c) shows the classifier residual for digit in (a), which is correctly classified by the NSC, but misclassified as "8" by the SRC; (d) shows the classifier residual for digit in (b), which is correctly classified by the SRC, but misclassified as "2" by the NSC. Both are correctly classified by the CROC with $\lambda = 0.1$.

to $n_i = 500$, the training dictionary per class is now over-complete and we will use a principal subspace $\mathbf{B}_i$ of dimension k for the NSC. We use the least-norm CRCs to re-do the experiment for both random projection and eigenvector projection when $k = 30$ and $k = 50$. As shown in Fig. 6, there is a jump in the performance when λ is around 0.05; and adopting the SCR does not give substantial gain compared with the computational-light method of optimizing the regularization parameter λ.

5.2 Face Recognition using CROC

We test the proposed CROC against the Extended Yale-B database [22], [32] and the AR database [33]. Since our main goal is to show the benefit of the extra freedom by considering the regularization path, we do not test the robustness of face recognition with disguise (sunglasses, scarves, etc.) in this work, yet such an extension is straightforward.

5.2.1 Extended Yale-B Database

The Extended Yale-B database contains 2414 frontal-face images of 38 individuals [32]. We use the cropped

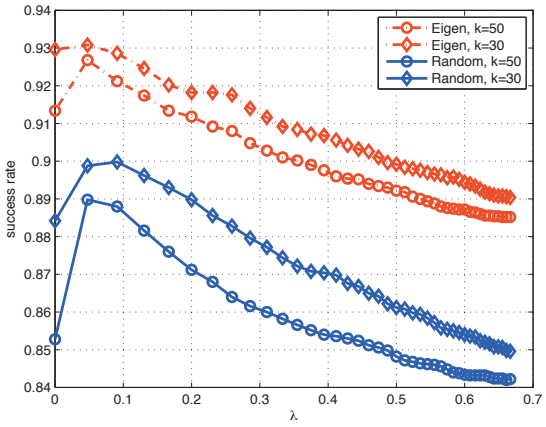


Fig. 6. Classification results for the regularization path for different methods using partial measurements for the MNIST digits database.

and unnormalized face images of size 192×168 which are captured under different illuminations [22] for our experiments. For each individual, we randomly select $n_i = 30$ training samples and the rest are for testing. We consider random features of dimensions $d = 100$ and 300 and test the variations below depending on if the full test image is available:

- With the full image: three CRCs corresponding to the LS representation $\mathbf{x}^{LS}$ from (2), the sparse representation $\mathbf{x}^{L1}$ from (6) and the least-norm representation $\mathbf{x}^{L2}$ from (7), are tested.
- Without the full image: two CRCs corresponding to the sparse representation $\mathbf{s}^{L1}$ from (6) and the L2 representation $\mathbf{s}^{L2}$ from (7) are tested.

The classification accuracy for the NSC, CRC and CROC with optimal λ are summarized in Table 3. It is obvious to see that when the full image is available, the $\mathbf{x}^{LS}$ representation achieves the best classification accuracy with low complexity. When the full image is not available, the SRC corresponds to $\lambda = 1$, and achieves better accuracy than the least-norm representation $\mathbf{s}^{L2}$ in terms of accuracy, in line with the previous work showing sparsity helps classification, in particular for smaller $d = 100$. However, this gain of using sparse representation [12] can be achieve by the least-norm representation with a properly tuned regularization parameter, around $\lambda = 0.1$, at much lower computational cost.

TABLE 3
Face recognition results for the NSC, CRC and CROC (with optimal λ): Full image with LS, L1 and L2 representations, partial images of various dimensions using Randomface with L1 and L2 representations for the Extended Yale-B database.

Scenario	Dim.	NSC	CRC	CROC (λ)
Full+LS	32256	97.46	**99.73**	**99.73** (0.8)
Full+L1	100	97.46	96.73	**97.82** (0.3)
	300	97.46	97.82	**98.28** (0.2)
Full+L2	100	97.46	91.20	**97.64** (0.2)
	300	97.46	97.82	**98.19** (0.2)
Reduced+L1	100	96.10	96.55	**97.19** (0.1)
	300	97.01	97.55	**98.19** (0.6)
Reduced+L2	100	96.10	89.11	**96.55** (0.1)
	300	97.01	97.19	**97.73** (0.2)

5.2.2 AR Database

Same as [12], we use a subset of 50 male subjects and 50 female subjects with only changes of illumination and expressions. For each subject, the seven images from Session 1 are used for training, and the other seven images from Session 2 are used for testing. The images are cropped to size 60×43.

Figure 7 shows the regularization path of face recognition results for CROC with different input for the CRC:

- With the full image: the CRC with LS representation $\mathbf{x}^{LS}$;
- Without the full image: the CRC with L2 representation $\mathbf{s}^{L2}$ using random projection, eigenvector projection and random pixel selection of the full image when $d = 100$ and $d = 300$.

In the full image case, we show that better accuracy can be achieved at $\lambda = 0.3$, about 1.5% improvement than at $\lambda = 1$, corresponding to the result in [20]. In almost all curves shown, some gain can be obtained by optimizing the regularization parameter λ. Figure 8 shows two face examples and corresponding random pixel selection features: (a) face "1" is correctly classified by the NSC, but misclassified as face "58" by the CRC; (b) face "2" is correctly classified by the CRC, but misclassified as face "25" by the NSC. Both are correctly classified by the CROC with $\lambda = 0.1$.

Figure 9 compares optimal classification result for the NSC, SRC and CROC with L2 representation using random pixel selection (partial), Randomface and Eigenface and LN CR with different feature dimensions $d = 30, 50, 100, 300$. The gain of the CROC with random pixel selection and Randomface is more significant than the gain with Eigenface.

5.3 Digit Recognition using CRBoosting

In this section, we test the proposed CRBoosting algorithm for digit recognition when the samples are compressively measured. We consider a set of candidate CRC that are commonly used in the literature, and a set of candidate NSC with different ranks. Again we make use of the MNIST Handwritten Digits database [31].

We use $n_{train} = 30$ or $n_{train} = 50$ samples per class for training, $n_{valid} = 100$ per class for the validation set to train CRBoosting, and $n_{test} = 500$ samples per class for testing. A random matrix of i.i.d. Gaussian entries is used to make $d = 80$ compressive measurements of each test sample. The candidates of CRC use $\mathbf{s}^{L2}$ and

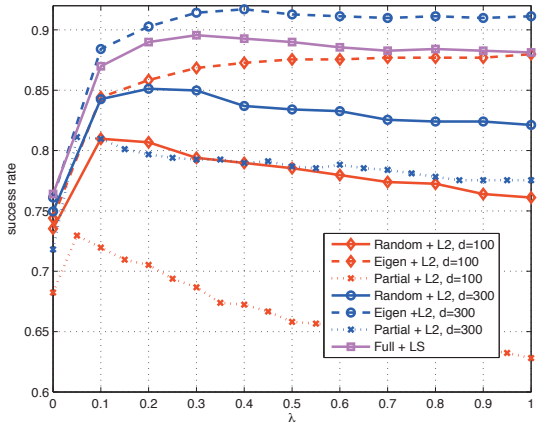
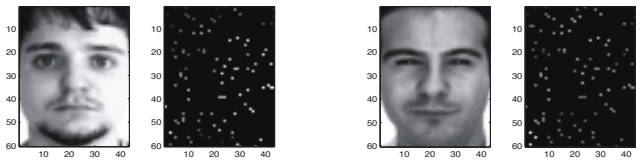


Fig. 7. Face recognition results on the regularization path for different projection and collaborative representations: Full image with L2 representation, random pixel selection (partial), random projection and eigenvector projection of full image with LS representations for the AR database.



(a) (b)

Fig. 8. Two face examples and corresponding random pixel selections: (a) face "1" is correctly classified by the NSC, but misclassified as face "58" by the CRC; (b) face "2" is correctly classified by the CRC, but misclassified as face "25" by the NSC. Both are correctly classified by the CROC with $\lambda = 0.1$.

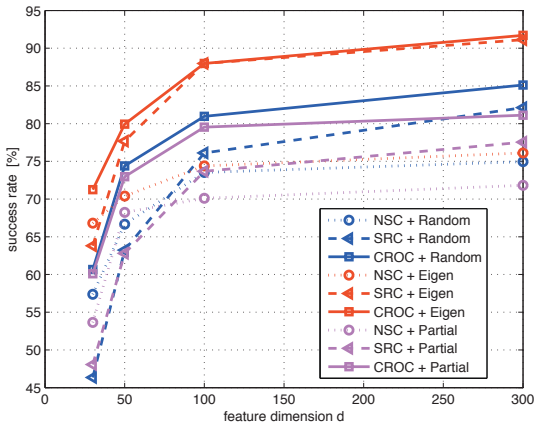


Fig. 9. Face recognition results for NSC, CRC and CROC using L2 representations versus different feature dimensions with random pixel selection (partial), Randomface and Eigenface for the AR database.

TABLE 4
Classification results of CRC(L_2), CRC(L_1), NSC(10), NSC(20), and CRBoosting on the validation and testing sets from $d = 80$ random measurements, with (a) $n_{train} = 30$ and (b) $n_{train} = 50$ training samples per class.

Random	Validation [%]	Testing [%]
CRC(L_2)	76.50	73.18
CRC(L_1)	77.30	73.48
NSC(10)	77.90	76.38
NSC(20)	79.50	76.94
CRBoosting	**80.20**	**78.46**

(a)

Random	Validation [%]	Testing [%]
CRC(L_2)	78.50	77.18
CRC(L_1)	80.90	78.24
NSC(10)	81.40	79.78
NSC(20)	82.60	81.04
CRBoosting	**83.70**	**81.64**

(b)

$\mathbf{s}^{L1}$ as input CRs, and we denote the CRC using $\mathbf{s}^{L_q}$ by CRC(L_q); the candidate NSC use rank $k = 10$ and $k = 20$, and we denote the NSC with rank k by NSC(k).

When $n_{train} = 30$, the weighting vector learned from CRBoosting is

$$\boldsymbol{\alpha} = [0, 0.1826, 0.3703, 0.4471],$$

where CRC(L_1), NSC(10) and NSC(20) are selected. When $n_{train} = 50$ the weighting vector learned from CRBoosting is

$$\boldsymbol{\alpha} = [0, 0.3406, 0, 0.6594],$$

where only CRC(L_1) and NSC(20) are selected. This shows that CRBoosting has the ability to select the most powerful representations to form the final classifier. In both cases the CRC(L_2) is not selected, which may be explained by its relative poor performance. Table 4 summarizes all the classification results, and CRBoosting performs best in both the validation and testing sets compared with all candidate CRCs.

We now use $n_{train} = 100$ samples per class for training and keep the size of validation and testing sets unchanged. We test both the CRBoosting and CRBoosting-T algorithms from a candidate set of CRC(L_2), CRC(L_1) and NSC(20) for $d = 80$. The weighting vector learned from CRBoosting is

$$\boldsymbol{\alpha} = [0, 0.3921, 0, 0.6079];$$

and the weighting vector learned from CRBoosting-T is

$$\boldsymbol{\alpha}_T = [0.0324, 0.1600, 0, 0.8076].$$

The classification results are summarized in Table 5, where both CRBoosting algorithms outperform all the

TABLE 5
Classification results of CRC(L_2), CRC(L_1), NSC(10), NSC (20) and CRBoosting on the validation and testing sets, with 100 training samples per class from $d = 80$ measurements for the MNIST dataset.

Random	Validation [%]	Testing [%]
CRC(L_2)	85.40	79.44
CRC(L_1)	85.90	82.82
NSC(10)	86.10	82.46
NSC(20)	86.60	84.30
CRBoosting	**88.10**	**85.84**
CRBoosting-T	**88.30**	**86.04**

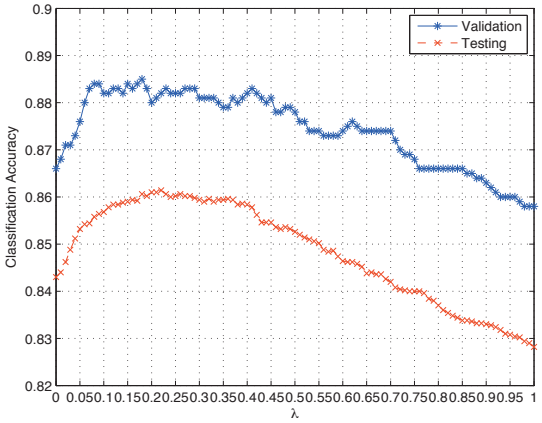


Fig. 10. The regularization path of CROC for NSC(20) and CRC(L_1) on both validation and testing sets for the MNIST dataset.

candidates, and CRBoosting-T achieves better result than CRBoosting.

Since CRC(L_2) and NSC(10) are not selected (having very small weights in the final classifier), we can compare this result with the regularization path of the CROC, where the residual is computed as a weighted sum of residuals from NSC(20) and CRC(L_1) as

$$r_i(\lambda) = (1-\lambda)\|\mathbf{z} - \mathbf{\Psi}_i \mathbf{s}_i^{NS(20)}\|_2^2 + \lambda\|\mathbf{z} - \mathbf{\Psi}_i \mathbf{s}_i^{L1}\|_2^2.$$

where $\lambda = 0$ coincides with the NSC, and $\lambda = 1$ coincides with CRC(L_1). We can see the weights learned from CRBoosting and CRBoosting-T are very consistent with the peak of the regularized path for both the validation and testing sets in Fig. 10.

5.4 Face Recognition using CRBoosting

We use the extended Yale-B database to test the performance of CRBoosting on face recognition. For each individual, we select the first $n_{train} = 20$ samples for training, the next $n_{valid} = 20$ samples for validation, and the rest are for testing. We use a random measurement matrix to take $d = 100$ compressive measurements of each sample. The candidates of CRC are CRC(L_1) and

TABLE 6
Classification results of CRC(L_2), CRC(L_1), NSC and CRBoosting on the validation and testing sets for the Extended Yale-B database from $d = 100$ measurements.

Random	Validation [%]	Testing [%]
CRC(L_2)	61.18	59.14
CRC(L_1)	70.66	65.65
NSC	72.50	72.85
CRBoosting	**74.21**	**75.07**
CRBoosting-T	**74.99**	**75.85**

CRC(L_2) as described earlier; and the candidate NSC use all training samples to form the subspace of rank 20. The classification results are summarized in Fig. 6, where the CRBoosting achieved a better performance than all candidates. The learned weighting vector from CRBoosting is

$$\boldsymbol{\alpha} = [0, 0.1677, 0.8323];$$

and the learned weighting vector from CRBoosting-T is

$$\boldsymbol{\alpha}_T = [0, 0.3741, 0.6259].$$

In both cases the CRC(L_2) is not selected due to its poor performance, and the learned weights between CRC(L_1) and NSC are comparable as the optimal value from CROC, indicating the effectiveness of the CRBoosting procedure. It is worth mentioning that although the CRBoosting procedure requires an additional validation set, it is possible to merge the training and validation set for testing after we learn the weights.

6 Conclusions

In this paper we explicitly decompose the multi-class classification problem into two steps, namely finding the collaborative representation and inputting it to the multi-class classifier. We explore different choices of collaborative representations and propose the Collaborative Representation Optimized Classifier (CROC) which provides a regularization path of classifiers where the NSC and the CRC are special cases on the whole regularization path. We show that classification performance can be further improved by optimally tuning the regularization parameter at no extra computational cost.

We further propose the Collaborative Representation based Boosting (CRBoosting) algorithm to efficiently combine multiple collaborative representations by classifying a test sample to the class with the minimal weighted sum of residuals from a set of candidate CRCs and NSCs, where the weights are found following an AdaBoosting based procedure. The ability to boost in the residual domain instead of in the decision domain allows CRBoosting to outperform the candidates even with only two candidates, which is not possible for AdaBoost. We also proved similar validation error bound for CRBoosting.

Our algorithms are validated through numerical results on digit recognition and face recognition in particular from compressively measured samples. We demonstrate the potential of exploring multiple collaborative representations over focusing on a particular choice in multi-class learning.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for helpful suggestions.

REFERENCES

[1] Y. Chi and F. Porikli, "Connecting the dots: From nearest subspace to collaborative representation," in *Computer Vision and Pattern Recognition (CVPR)*, 2012.
[2] C. H. Q. Ding and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural networks," *Bioinformatics*, vol. 17, pp. 349–358, 2001.
[3] K. Nigam, A. K. Mccallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," in *Machine Learning*, pp. 103–134, 1999.
[4] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, pp. 71–86, Jan. 1991.
[5] S. Verdu, *Multiuser detection*. Cambridge university press, 1998.
[6] D. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Comm. Pure and Applied Math.*, vol. 59, no. 6, pp. 797–829, 2006.
[7] E. Candeś, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure and Applied Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.
[8] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, pp. 4311–4322, 2006.
[9] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
[10] R. Calderbank, S. Jafarpour, and R. Schapire, "Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain," tech. rep., 2009.
[11] Y. Chi, Y. Xie, and R. Calderbank, "Compressive Demodulation of Mutually Interfering Signals," *IEEE Trans. on Information Theory*, Mar. 2013. submitted.
[12] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 31, no. 2, 2009.
[13] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: which helps face recognition?," in *International Conference on Computer Vision (ICCV)*, 2011.
[14] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *East*, vol. 43, no. 1, pp. 129–159, 2001.
[15] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
[16] M. Yang and L. Zhang, "Gabor feature based sparse representation for face recognition with Gabor occlusion dictionary," *European Conference on Computer Vision (ECCV)*, 2010.
[17] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.
[18] Q. Shi, H. Li, and C. Shen, "Rapid face recognition using hashing," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.
[19] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural computation*, vol. 15, no. 2, pp. 349–396, 2003.
[20] Q. Shi, A. Eriksson, A. Hengel, and C. Shen, "Is face recognition really a compressive sensing problem?," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.
[21] J. Wright, A. Ganesh, A. Yang, Z. Zhou, and Y. Ma, "Sparsity and robustness in face recognition," *Arxiv*, 2012.
[22] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.
[23] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory*, pp. 23–37, 1995.
[24] P. Belhumeur, J. Hespanda, and D. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 19, no. 7, pp. 711–720, 1997.
[25] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 25, no. 2, pp. 218–233, 2003.
[26] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
[27] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary mathematics*, vol. 26, no. 189-206, p. 1, 1984.
[28] L. Meier, S. V. D. Geer, P. Bhlmann, and E. T. H. Zrich, "The group lasso for logistic regression," *Journal of the Royal Statistical Society, Series B*, 2008.
[29] J. Bi, D. Wu, L. Lu, M. Liu, Y. Tao, and M. Wolf, "Adaboost on low-rank PSD matrices for metric learning," *Computer Vision and Pattern Recognition (CVPR)*, 2011.
[30] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," Apr. 2011.
[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[32] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 23, no. 6, pp. 643–660, 2001.
[33] A. Martinez and R. Benavente, "The AR face database," *CVC Technical Report 24*, 1998.

Yuejie Chi (S'09-M'12) received the Ph.D. degree in Electrical Engineering from Princeton University in 2012, and the B.E. (Hon.) degree in Electrical Engineering from Tsinghua University, Beijing, China, in 2007. Since September 2012, she has been an assistant professor with the department of Electrical and Computer Engineering and the department of Biomedical Informatics at the Ohio State University.

She has held visiting positions at Colorado State University, Stanford University and Duke University, and interned at Qualcomm Inc. and Mitsubishi Electric Research Lab. Her research interests include high-dimensional data analysis, statistical signal processing, machine learning and their applications in communications, networks, imaging and bioinformatics.

She received the best paper award from the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) in 2012. She received a Roberto Padovani scholarship from Qualcomm Inc. in 2010, and an Engineering Fellowship from Princeton University in 2007.

Fatih Porikli is a Distinguished Research Scientist at Mitsubishi Electric Research Labs (MERL). He received his PhD from NYU Poly, NY. Before joining MERL in 2000, he developed satellite imaging solutions at Hughes Research Labs and 3D display systems at AT&T Research Labs. His work covers areas including computer vision, machine learning, video surveillance, multimedia processing, structured and manifold based pattern recognition, biomedical vision, radar signal processing, and online learning with over 100 publications and 60 patents. He mentored more than 40 PhD students. He received R&D100 2006 Award in the Scientist of the Year category (select group of winners) in addition to 3 IEEE Best Paper Awards and 5 Professional Prizes. He serves as an Associate Editor of IEEE Signal Processing Magazine, SIAM Journal on Imaging Sciences, Springer Machine Vision Applications, Springer Real-time Image and Video Processing, and EURASIP Journal on Image and Video Processing. He served as the General Chair of IEEE Advanced Video and Signal based Surveillance Conference (AVSS) 2010 and participated in the organizing committee of many IEEE events.