Max Residual Classifier

Hien V. Nguyen
University of Maryland, College Park
hien@umiacs.umd.edu

Vishal M. Patel
University of Maryland, College Park
pvishalm@umiacs.umd.edu

Abstract

We introduce a novel classifier, called max residual classifier (MRC), for learning a sparse representation jointly with a discriminative decision function. MRC seeks to maximize the differences between the residual errors of the wrong classes and the right one. This effectively leads to a more discriminative sparse representation and better classification accuracy. The optimization procedure is simple and efficient. Its objective function is closely related to the decision function of the residual classification strategy. Unlike existing methods for learning discriminative sparse representation that are restricted to a linear model, our approach is able to work with a non-linear model via the use of Mercer kernel. Experimental results show that MRC is able to capture meaningful and compact structures of data. Its performances compare favourably with the current state of the art on challenging benchmarks including rotated MNIST, Caltech-101, Caltech-256, and SHREC'11 non-rigid 3D shapes.

1. Introduction

Sparseness has proven to be a valuable property in statistical signal processing. A random variable is sparse if it is active more rarely compared to Gaussian random variable of the same variance. Common measures of sparseness are convex functions such as kurtosis and ℓ_1-norm. This comes from the property that expectation of convex function is large if data is concentrated in the extremes, which are near zero and very far from zero. The ℓ_0-norm is another effective measure of sparseness. Optimization using ℓ_0-norm is proven equivalent to using ℓ_1-norm under certain assumptions [6]. Interestingly, maximization of sparseness is also equivalent to maximization of independence if data have a super Gaussian distribution, which is often the case for natural images [14].

It has been conjectured that biological systems make use of *sparse coding* strategy for representing visual and auditorial inputs [26, 2]. These arguments are supported by several theoretical, computational, and experimental studies which suggest that brains encode sensory information using a small number of active neurons at any given point of time [9, 37]. Sparse coding can confer several advantages including higher storage efficiency and better energy consumption. Although the connection between sparseness and the underlying mechanism of biological systems is still being investigated, sparse coding has been found to work well in practice. It leads to the emergence of new theories like compressive sensing. Also, it significantly outperforms the traditional approaches in many practical applications ranging from compression, denoising, recognition, etc.

These merits of sparseness have result in recent explosion of activities in modelling a signal using sparse representation. This approach is further corroborated by the observation that most signals encountered in practical applications are compressible. In other words, their sorted magnitudes in some basis obey power law decay. As a result, a signal can be well approximated by linear combinations of a few atoms taken from an appropriate basis or a dictionary $\mathbf{D}$. The choice of basis depends on the nature of data and the task at hand. For example, predefined basis such as wavelets or Fourier basis are the most common among the traditional choices for signal compression.

In recent years, numerous papers have shown the benefits of learning the basis directly from the data [7, 32, 34, 40]. This approach was first introduced in [26] which shows that biologically plausible features similar to Gabor wavelet can be learned from natural images. In its most basic form, a basis is learned by minimizing the reconstruction errors [1, 23]. This gives rise to a simple classification scheme called the *residual classifier*. In particular, a novel signal is assigned to the class with the smallest residual error resulting from a sparse approximation. The residual classifier works surprisingly well and, in many cases [35, 41], outperforms sophisticated classifiers like support vector machine (SVM) [38]. One of the main reasons behind the successes is the highly compact and robust data representation.

Margin classifiers and residual classifier represent two different approaches to classification. The first one focuses on learning a good decision function that maximizes the discrimination, while the second one seeks a compact basis

to represent the data. Over-fitting often happens to margin classifiers when heavy noise and high redundancies are present within the input signals. The residual classifier is more robust, however, is not optimal for classification since no discriminative information is taken into account during the training process. There have been several attempts to enhance the discriminant power of the residual classifier [25, 45]. However, they are mostly restricted to a linear model with rather involved optimization procedures.

Our paper makes the following contributions:

- Proposes a novel framework for learning the sparse representation jointly with discriminative classifier.
- Develops a simple and efficient optimization procedure; Investigates the connection to margin-classifier.
- Provides an extension of our approach to the non-linear case via the Mercer kernel.
- Presents numerous experimental results and discussions to evaluate the proposed algorithm.

The rest of our paper is organized as follows. Section 1.1 summarizes the related efforts. Section 1.2 defines notations used through out the paper. Section 2 briefly explains the necessary background on residual classification. Section 3 introduces MRC formulation. Section 4 elaborates on the optimization procedures. Section 5 interprets the MRC formulation from the perspective of margin classifiers. Section 6 presents an extension of MRC for handling non-linear data. Section 7 discusses experimental results on challenging datasets such as rotated MNIST and Shrec 3D Non-rigid shapes. Finally, section 8 concludes the paper.

1.1. Related Work

Unsupervised dictionary learning has been used for the classification of audio signals [13, 31], human faces [41], and general images [30, 17, 43]. Shortly after, various discriminative dictionary learning methods have been proposed in the literature [24, 29, 47, 31, 45, 35]. In [24], a multiclass version of the logistic function on the residual errors is used to control the trade-off between reconstruction and discrimination. In contrast, [29] learns sparse representation jointly with a linear classifier by maximizing the discrimination of the sparse coefficients. In a similar vein, [47] directly incorporates the labels in the dictionary updating stage to enhance the discrimination of sparse coefficients in the context of a linear classifier. Inspired by the fact that the accuracy of sparse coding stage depends on the incoherence between the dictionary atoms, [31] adds a term that promotes incoherence between dictionaries of different classes. [45] has proposed combining both residual errors and sparse coefficients for classification.

1.2. Notations

Vectors are denoted by bold lower case letters and matrices by bold upper case letters. The ℓ_0-pseudo-norm $\|.\|_0$ is defined as the number of non-zero elements in a vector. The ℓ_1-norm of an n dimensional vector $\mathbf{x} = [x_1, \cdots, x_n]^T$ is defined as $\|\mathbf{x}\|_1 = \sum_i |x_i|$. The Frobenius norm of a matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$ is defined as $\|\mathbf{X}\|_F = (\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{X}(i,j)^2)^{1/2}$. The total number of classes is C. The dimension of input signal is denoted by n, output signals by d, dictionary size by $\{K_c\}_{c=1}^{C}$. The pair $(\mathbf{y}, \ell)$ denotes a training sample drawn from $\mathcal{X}$, where $\mathbf{y} \in \mathbb{R}^n$ is an input signal and $\ell \in \{1, \ldots, C\}$ is the corresponding class label. $S_c \in \mathcal{X}$ denote the set of training samples from c-th class (i.e. $\ell = c$). N_c is the number of samples in S_c. $\mathbf{Y}_c \in \mathbb{R}^{n\times N_c}$ is the matrix formed by horizontal concatenation of column signals in S_c. $\mathbf{Y} = [\mathbf{Y}_1 \ldots \mathbf{Y}_C] \in \mathbb{R}^{n\times N}$ is formed by the concatenation of all signals, where $N = \sum_{c=1}^{C} N_c$ is the total number of training from all classes.

2. Classification Using Residual From Sparse Representation

Residual classifier comprises of two main stages. First, class-specific dictionaries $\mathbf{D}_c \in \mathbb{R}^{n\times K_c}$ are learned by minimizing the reconstruction errors:

$$\{\mathbf{D}_c^*, \mathbf{X}_c^*\} = \underset{\mathbf{D}_c, \mathbf{X}_c}{\operatorname{argmin}} \|\mathbf{Y}_c - \mathbf{D}_c\mathbf{X}_c\|_F^2 \tag{1}$$

$$\text{subject to: } \|\mathbf{X}_c(:,i)\|_0 \leq T_0,\ \forall i \in \{1, \ldots, N_c\}, \tag{2}$$

where $\mathbf{X}_c(:,i)$ denote the i-th column of $\mathbf{X}_c \in \mathbb{R}^{K_c\times N_c}$. Note that the ℓ_0-norm constraint can be replaced with an ℓ_1-norm constraint. This replacement does not affect the development of our framework.

Second, given a novel sample $(\mathbf{y}, \ell)$, sparse codes are obtained for each class-specific dictionary using pursuit algorithms like orthogonal matching pursuit (OMP) [27]:

$$\mathbf{x}_c = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{D}_c\mathbf{x}\|_F^2 \ \text{ s.t. } \ \|\mathbf{x}\|_0 \leq T_0. \tag{3}$$

Alternatively, one can also perform sparse coding in a collaborative manner in which the class-specific dictionaries are concatenated together:

$$\mathbf{x}_{con} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{D}_{con}\mathbf{x}\|_F^2 \text{ s.t. } \|\mathbf{x}\|_0 \leq T_0, \tag{4}$$

where $\mathbf{x}_{con} = [\mathbf{x}_1^T \ldots \mathbf{x}_C^T]^T$, $\mathbf{D}_{con} = [\mathbf{D}_1 \ldots \mathbf{D}_C]$, and T_0 is the chosen sparsity constant. The residual vector resulting from the sparse approximation of $\mathbf{y}$ using $\mathbf{D}_c$ is simply

$$\mathbf{r}(\mathbf{y}, c) = \mathbf{y} - \mathbf{D}_c\mathbf{x}_c. \tag{5}$$

In both cases, classification is done by assigning the novel sample to the class of smallest residual errors measured by the magnitude of the residual vector:

$$\hat{\ell} = \underset{c}{\operatorname{argmin}} \|\mathbf{r}(\mathbf{y}, c)\|^2. \tag{6}$$

3. Max Residual Classifier (MRC)

The residual classification strategy yields state-of-the-art performances for many practical computer vision tasks [13, 31, 41, 30, 17, 43]. However, it is not optimal for classification since discriminative information is not considered during training. It is also unable to transform data to a latent space to deal with intra-class variation and non-linearity of data like in SVM. In this section, we develop an alternative learning framework to mitigate these drawbacks.

An important property of the residual classifier is its sole dependence on the total differences of residual errors defined as:

$$\Delta(\mathbf{y}, c) = \sum_{i=1}^{C} \left(\|\mathbf{r}(\mathbf{y}, c)\|^2 - \|\mathbf{r}(\mathbf{y}, i)\|^2 \right). \quad (7)$$

Then the classification rule in (6) is equivalent to:

$$\hat{\ell} = \underset{c}{\operatorname{argmin}}\ \Delta(\mathbf{y}, c). \quad (8)$$

This property constitutes the main intuition behind our approach. It suggests that the discriminative power of residual classifier can be improved simply by minimizing:

$$\mathbb{E}[\Delta(\mathbf{y}, \ell)] = \int_{(\mathbf{y},\ell)\in\mathcal{X}} \Delta(\mathbf{y}, \ell)\ dP(\mathbf{y}, \ell), \quad (9)$$

where $P(\mathbf{y}, \ell)$ is the joint probability distribution of input signals and their corresponding labels. The minimization of (9) promotes the reduction of residual errors for the right classes, and the amplification of it otherwise. The objective function can not be minimized directly since we often do not have prior knowledge about $P(\mathbf{y}, \ell)$. A common practice is minimizing the empirical estimation of the objective function given a set of training samples:

$$\mathbb{E}_{emp}[\Delta(\mathbf{y}, \ell)] = \sum_{c=1}^{C} \sum_{(\mathbf{y},\ell)\in S_c} \Delta(\mathbf{y}, \ell). \quad (10)$$

Signals are usually assumed to lie on a low-dimensional manifold embedded in a high dimensional space. Dealing with the high-dimension is not practical for both learning and inference tasks. To this end, we allow the transformation of signals to a latent space where representations are more compact and the residual errors are more separated. For notational simplicity, we first consider the case of linear transformation. The extension to non-linear case will be presented in later section. Let $\mathbf{W} \in \mathbb{R}^{d\times n}$ denote the desirable linear operator, whose rows are orthogonal and normalized to unit-norm to prevent degenerated solutions. The complete description of MRC is as follows:

$$\{\mathbf{W}^*, \mathbf{D}_c^*, \mathbf{X}_{con}^{c*}\} = \underset{\mathbf{W},\mathbf{D}_c,\mathbf{X}_{con}^c}{\operatorname{argmin}}\ \mathbb{E}_{emp}[\Delta(\mathbf{W}\mathbf{y}, \ell)] \quad \text{s.t} \quad (11)$$

$$\mathbf{W}\mathbf{W}^T = \mathbf{I} \ \text{ and } \ \|\mathbf{X}_{con}^c(:, j)\|_0 \le T_0, \quad \forall c = 1, \ldots, C.$$

Here, we enforce sparseness in a collaborative manner. $\mathbf{X}_{con}^c$ is the sparse coefficients associated with the c-th class, resulting from approximating $\mathbf{W}\mathbf{Y}_c$ using $\mathbf{D}_{con} = [\mathbf{D}_1 \ldots \mathbf{D}_C]$. $\mathbf{X}^c(:, j)$ is the j-th column of $\mathbf{X}_{con}^c$. The joint sparsity constraint allows inter-class competition among dictionary atoms and practically leading to better performances. More explanation is provided in the next section.

4. Optimization Procedure

First, we expand the objective function in (11) to a more useful form for optimization:

$$\mathbb{E}_{emp}[\Delta(\mathbf{W}\mathbf{y}, \ell)] = \sum_{c=1}^{C} \sum_{(\mathbf{y},\ell)\in S_c} \Delta(\mathbf{W}\mathbf{y}, \ell) =$$

$$\sum_{c=1}^{C}\sum_{i=1}^{C} \left(\|\mathbf{W}\mathbf{Y}_c - \mathbf{D}_c\mathbf{X}_c^c\|_F^2 - \|\mathbf{W}\mathbf{Y}_c - \mathbf{D}_i\mathbf{X}_i^c\|_F^2 \right), \quad (12)$$

where $\mathbf{D}_c \in \mathbb{R}^{d\times K_c}$ is now the dictionary in the latent space, and $\mathbf{X}_i^c$ is the sparse codes of $\mathbf{W}\mathbf{Y}_c$ over $\mathbf{D}_i$. In connection to (11), the joint sparse codes can be expressed as $\mathbf{X}_{con}^c = [\mathbf{X}_1^{c\,T} \ldots \mathbf{X}_C^{c\,T}]^T$. The first term and the second term of (12) are residual errors of the true classes and the wrong ones, respectively. In order to solve (11), we restrict the solution of $\mathbf{D}_c$ to the linear subspace spanned by the input signals. It can be shown that most, if not all, dictionary learning algorithms satisfy this condition. They include MOD [8], KSVD [1], and their variants [23]. Under this condition, we introduce a proposition to facilitate the development of the optimization algorithm.

Proposition 1. *There exists an optimal solution* $\mathbf{W}^*$ *and* $\mathbf{D}_c^*$ *to (11) that has the following form:*

$$\mathbf{W}^* = (\mathbf{Y}\mathbf{A})^T, \quad \mathbf{D}_c^* = \mathbf{A}^T\mathbf{K}\mathbf{B}_c \quad (13)$$

for some $\mathbf{A} \in \mathbb{R}^{N\times d}$, *some* $\mathbf{B}_c \in \mathbb{R}^{N\times K_c}$, *and* $\mathbf{K} = \mathbf{Y}^T\mathbf{Y}$.

As a corollary of the above proposition, $\mathbf{W}$ and $\mathbf{D}_c$ can be found by optimizing $\mathbf{A}$ and $\mathbf{B}_c$. There are several advantages for doing this. First, we can jointly updating both the transformation and the dictionaries via $\mathbf{A}$. Second, operating on $\mathbf{A}$ and $\mathbf{B}_c$ permits an easy extension to the non-linear case via Mercer kernels, which will be explained in details later. The above proposition elucidates the effect of transformation $\mathbf{W}$ on dictionaries. In particular, columns of $\mathbf{W}$ define the subspace that dictionary atoms live in.

Solving for A: First, we fix $\mathbf{B}_c$ and the associated sparse coefficients $\mathbf{X}_{con}^c$ in order to solve for $\mathbf{A}$. Substituting (13) into (12) together with some simple algebraic manipula-

tions, we arrive at an elegant form of the objective function:

$$\mathbb{E}_{emp}[\Delta(\mathbf{W}\mathbf{y},\ell)] = \text{tr}\left(\mathbf{A}^T(\mathbf{R}_1 - \mathbf{R}_2)\mathbf{A}\right), \quad (14)$$

$$\mathbf{R}_1 = (C-1)\sum_{c=1}^{C}(\mathbf{K}_c - \mathbf{K}\mathbf{B}_c\mathbf{X}_c^c)(\mathbf{K}_c - \mathbf{K}\mathbf{B}_c\mathbf{X}_c^c)^T, \quad (15)$$

$$\mathbf{R}_2 = \sum_{c=1}^{C}\sum_{i=1,i\neq c}^{C}(\mathbf{K}_c - \mathbf{K}\mathbf{B}_i\mathbf{X}_i^c)(\mathbf{K}_c - \mathbf{K}\mathbf{B}_i\mathbf{X}_i^c)^T. \quad (16)$$

where $\mathbf{K}_c = \mathbf{Y}^T\mathbf{Y}_c \in \mathbb{R}^{N\times N_c}$. $\mathbf{R}_1$ and $\mathbf{R}_2$ indicate how well samples are approximated using dictionaries from their own classes and from the other classes, respectively. The solution of $\mathbf{A}$ is given by an eigendecomposition. In particular, columns of $\mathbf{A}$ are the eigenvectors corresponding to the smallest eigenvalues of $(\mathbf{R}_1 - \mathbf{R}_2)$.

Solving for $(\mathbf{B}_c, \mathbf{X}_{con}^c)$: We fix $\mathbf{A}$ in order to solve for $\mathbf{B}_c$. This can be done by first solving for $\mathbf{D}_c$. Then from (13) $\mathbf{B}_c$ is obtained by taking the pseudo-inverse:

$$\mathbf{B}_c = (\mathbf{A}^T\mathbf{K})^\dagger\mathbf{D}_c. \quad (17)$$

where the $(.)^\dagger$ denotes the Moore-Penrose pseudo-inverse. Note that this operation is exact since we can show that $\mathbf{D}_c$ is in the row subspace of $\mathbf{A}$. Unfortunately, the direct optimization of (11) over $\mathbf{D}_c$ is difficult. We instead minimize its upper bound given by the following proposition.

Proposition 2. *The objective function in (11) is upper bounded by the following function:*

$$\mathcal{L} = (C-1)C\sum_{c=1}^{C}\left(\|\mathbf{A}^T\mathbf{K}_c - \mathbf{D}_{con}\mathbf{X}^c\|_F^2 \quad (18)$$

$$+\alpha\sum_{i=1,i\neq c}^{C}\|\mathbf{D}_i\mathbf{X}_i^c\|_F^2 - \beta\|\mathbf{A}^T\mathbf{K}_c\|_F^2\Big), \quad (19)$$

where $\mathbf{D}_{con} = [\mathbf{D}_1,\ldots,\mathbf{D}_c]$, $\mathbf{K}_c = \mathbf{Y}^T\mathbf{Y}_c$, $\alpha = \frac{C}{C-1}$, *and* $\beta = \frac{1}{2C}$.

The last term on the right hand side of (19) is independent of $\mathbf{D}_c$, therefore, can be discarded. The minimization of (19) requires the joint dictionary to represent data from all classes well (first term of $\mathcal{L}$), and the representational power from the wrong class to be small (second term of $\mathcal{L}$). This implicitly requires the residual magnitudes from the wrong classes to be as large as possible. The upper bound turns out to have an interesting connection to FDDL [45]. In particular, the first two terms of $\mathcal{L}$ is similar to the first term in the objective function of FDDL up to a scaling constant. However, [45] does not provide any explicit connection between their objective function and the residual classifier. In addition, FDDL does not allow the transformation of data to a latent space in order to better deal with intra-class variations and non-linearities of data.

The upper-bound $\mathcal{L}$ is a convex function with respect to $\mathbf{D}_c$ when $\mathbf{W}$ and $\mathbf{X}_c^k$ are fixed, which can be reduced to:

$$\alpha\sum_{k=1,k\neq c}^{C}\|\mathbf{D}_c\mathbf{X}_c^k\|_F^2 + \sum_{k=1}^{C}\|\mathbf{E}_c^k - \mathbf{D}_c\mathbf{X}_c^k\|_F^2, \quad (20)$$

$$\text{where} \quad \mathbf{E}_c^k = \mathbf{A}^T\mathbf{K}_k - \sum_{j=1,j\neq c}^{C}\mathbf{D}_j\mathbf{X}_j^k. \quad (21)$$

$\mathbf{E}_c^k$ is the approximation error when using all dictionaries except $\mathbf{D}_c$ to approximate signals from class k. The optimal $\mathbf{D}_c$ is obtained simply by:

$$\mathbf{D}_c = \mathbf{M}_c\mathbf{V}_c^\dagger, \text{ where } \mathbf{M}_c = \sum_{k=1}^{C}\mathbf{E}_c^k{\mathbf{X}_c^k}^T, \quad (22)$$

$$\mathbf{V}_c = (1+\alpha)\sum_{k=1,k\neq c}^{C}\mathbf{X}_c^k{\mathbf{X}_c^k}^T + \mathbf{X}_c^c\mathbf{X}_c^{cT}. \quad (23)$$

The minimization of the upper bound $\mathcal{L}$ over $\mathbf{X}_{con}^c$ reduces to minimizing:

$$\|\mathbf{A}^T\mathbf{K}_c - \mathbf{D}_{con}\mathbf{X}_{con}^c\|_F^2 + \alpha\sum_{k=1,k\neq c}^{C}\|\mathbf{D}_k\mathbf{X}_k^c\|_F^2 \quad (24)$$

$$\text{subject to: } \|\mathbf{X}_{con}^c(:,j)\|_0 \leq T_0$$

$\mathbf{X}^c$ can be solved using any greedy algorithms. In this paper, we use the orthogonal matching pursuit (OMP) algorithm due to its high efficiency. The objective function in (24) can be rewritten in a more OMP-friendly form:

$$\|\mathbf{Z}_c - \mathbf{D}_{exp}^c\mathbf{X}_{con}^c\|_F^2, \quad (25)$$

where $\mathbf{Z}_c$ and $\mathbf{D}_{exp}^c$ are defined as follows:

$$\mathbf{Z}_c = \begin{pmatrix}(\mathbf{A}^T\mathbf{K}_c)^T & \mathbf{0} & \ldots & \mathbf{0}\end{pmatrix}^T, \quad (26)$$

$$\mathbf{D}_{exp}^c = \begin{pmatrix}\mathbf{D}_1 & \mathbf{D}_2 & \ldots & \mathbf{D}_C \\ \gamma_1\mathbf{D}_1 & 0 & \ldots & 0 \\ 0 & \gamma_2\mathbf{D}_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \gamma_C\mathbf{D}_C\end{pmatrix}, \quad (27)$$

$$\gamma_i = \begin{cases}\sqrt{\alpha}, & \text{if } i \neq c \\ 0 & \text{if } i = c\end{cases}, \quad \forall i \in \{1,\ldots,C\}. \quad (28)$$

The computation of OMP performed on the expanded dictionary $\mathbf{D}_{exp}^c$ is only twice the computation of the OMP performed on the original $\mathbf{D}_{con}$. This is because each column of $\mathbf{D}_{exp}^c$ contains at most $2d$ non-zero coefficients. When the ℓ_1-norm is used for measuring sparseness instead of ℓ_0-norm, the updating of $\mathbf{X}_{con}^c$ can be done by algorithms such as FISTA [3] and Iterative Projective Method [33].

5. Interpretation

For simplicity of notations, we consider a two-class problem, i.e. $C = 2$. The generalization to $C > 2$ is straight forward. Given a novel test signal $\mathbf{z} \in \mathbb{R}^n$ with an unknown label, the classification is done by considering the expression in (29). In particular, the classifier predicts class 1 if $\Delta(\mathbf{Wz}, 1) \leq 0$ and class 2 otherwise, where:

$$\Delta(\mathbf{Wz}, 1) = \|\mathbf{r}(\mathbf{Wz}, 1)\|^2 - \|\mathbf{r}(\mathbf{Wz}, 2)\|^2 = \sum_{i=1}^{d} \left\langle \text{vec}(\mathbf{w}_i^T \mathbf{w}_i), \text{vec}\left(\boldsymbol{\delta}_1 \boldsymbol{\delta}_1^T - \boldsymbol{\delta}_2 \boldsymbol{\delta}_2^T\right) \right\rangle \quad (29)$$
$$\text{where} \quad \boldsymbol{\delta}_i = \mathbf{z} - \mathbf{Y}\mathbf{B}_i\mathbf{x}_i, \quad \forall i = [1, 2].$$

Here, vec(.) is the vectorization of a matrix, $\langle ., . \rangle$ denote a dot product between two vectors, and $\mathbf{w}_i$ is the i-th row of $\mathbf{W}$. One can think of $\boldsymbol{\delta}_i$ as the residual vectors in the input space ($\mathbb{R}^n$).

This is indeed a margin-classifier whose separating hyperplane specified by the normal vector $\sum_{i=1}^{d} \text{vec}(\mathbf{w}_i^T \mathbf{w}_i)$, and input feature by vec $\left(\boldsymbol{\delta}_1 \boldsymbol{\delta}_1^T - \boldsymbol{\delta}_2 \boldsymbol{\delta}_2^T\right)$. First, note that the normal vector is a sum of d vectors. Each of which constructs a simple classifier whose separating hyperplane has only n degree of freedom instead of n^2. The overall decision function is equivalent to average pooling of d simple classifiers together for making a decision. Second, note that the input feature contains second-order interactions between elements of residual vectors $\boldsymbol{\delta}_i$. This is similar to classification using the residual vectors $\boldsymbol{\delta}_i$ with a second-degree polynomial kernel. The decision function in (29) inherits the discriminative nature from margin-classifier as well as the robustness from sparse representation.

6. Kernel Max Residual Classifier

Non-linearities arise in many practical applications of computer vision. For example, popular descriptors such as spatial pyramid and region of covariance both have non-linear distance measures. Non-linear structure in data can be exploited by transforming the data into a high-dimensional feature space where they might exist as a simple Euclidean geometry. In order to avoid the computational issues related to high-dimensional mapping, Mercer kernels are often used to carry out the mapping implicitly. We adopt Mercer kernels for extending MRC to the non-linear case.

Let $\Phi : \mathbb{R}^d \rightarrow \mathcal{H}$ be a mapping from the input space to the reproducing kernel Hilbert space $\mathcal{H}$. Let $\kappa : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ be the kernel function associated with Φ. The mapping $\mathcal{M}$ from the input space to the latent space is no longer linear. It, however, can be characterized by a compact, linear operator $\boldsymbol{\mathcal{W}} : \mathcal{H} \rightarrow \mathbb{R}^d$ that maps every input signal $\mathbf{z} \in \mathbb{R}^n$ to $\boldsymbol{\mathcal{W}}\Phi(\mathbf{z}) \in \mathbb{R}^d$. Following a similar spirit with the proposition 1, by letting $\boldsymbol{\mathcal{K}} = \langle \Phi(\mathbf{Y}), \Phi(\mathbf{Y}) \rangle_{\mathcal{H}} = [\kappa(\mathbf{y}_i, \mathbf{y}_j)]_{i,j=1}^N$, we can show that:

$$\boldsymbol{\mathcal{W}}^* = \mathbf{A}^T \Phi(\mathbf{Y})^T; \; \mathbf{D}_c^* = \mathbf{A}^T \boldsymbol{\mathcal{K}} \mathbf{B}_c. \quad (30)$$

Using (30), we can re-write the mapping $\mathcal{M}$ explicitly as:

$$\mathcal{M} : \mathbf{z} \in \mathbb{R}^n \rightarrow \boldsymbol{\mathcal{W}}\Phi(\mathbf{z}) = \mathbf{A}^T \langle \Phi(\mathbf{Y}), \Phi(\mathbf{z}) \rangle_{\mathcal{H}} = \mathbf{A}^T [\kappa(\mathbf{y}_1, \mathbf{z}), \ldots, \kappa(\mathbf{y}_N, \mathbf{z})]^T \quad (31)$$

Let $\boldsymbol{\mathcal{K}}_c = \langle \Phi(\mathbf{Y}), \Phi(\mathbf{Y}_c) \rangle_{\mathcal{H}}$. The non-linear MRC is equivalent to minimization of the following objective function:

$$\sum_{c=1}^{C} \sum_{i=1}^{C} \left(\|\mathbf{A}^T \boldsymbol{\mathcal{K}}_c - \mathbf{D}_c \mathbf{X}_c^c\|_F^2 - \|\mathbf{A}^T \boldsymbol{\mathcal{K}}_c - \mathbf{D}_i \mathbf{X}_i^c\|_F^2 \right) \quad (32)$$

We notice that the objective function does not explicitly depend on $\Phi(.)$, but the kernel matrices $\boldsymbol{\mathcal{K}}_c$. The optimal $\mathbf{A}$ and $\mathbf{B}_c$ can be solved in exactly the same way as in the linear case with $\mathbf{K}_c$ replaced by $\boldsymbol{\mathcal{K}}_c$. Note that in the non-linear

Input: Kernel matrix $\{\boldsymbol{\mathcal{K}}_c\}_{c=1}^C$, sparse setting T_0, dictionary size $\{K_c\}_{c=1}^C$, output dimension d.
Task: Find $\mathbf{A}^*$ and $\{\mathbf{B}_c^*\}_{c=1}^C$ by solving (11).
Initialize:
- Set iteration $J = 1$. Set columns of $\mathbf{A}$ to d dominant eigenvectors of $\boldsymbol{\mathcal{K}}$. Set $\{\mathbf{B}_c\}_{c=1}^C$ to random matrices.
Stage 1: *Sparse Coding*
- Solve for $\mathbf{X}_{con}^c$ as in (25) using the OMP algorithm, $\forall c = \{1, \ldots, C\}$
Stage 2: *Dictionary Update*
- Solve for $\mathbf{D}_c$ as in (22), $\forall c \in \{1, \ldots, C\}$
- Update $\mathbf{B}_c = (\mathbf{A}^T \mathbf{K})^\dagger \mathbf{D}_c$, $\forall c \in \{1, \ldots, C\}$
Stage 3: *Transformation Update*
- Compute $\mathbf{R}_1$ as in (15), and $\mathbf{R}_2$ as in (16)
- Perform eigendecomposition of $(\mathbf{R}_1 - \mathbf{R}_2) = \mathbf{U}\Lambda\mathbf{U}^T$
- Set $\mathbf{A} = \mathbf{U}(:, \mathcal{I}_J)$, where $\mathcal{I}_J$ is the index set of d smallest eigenvalues of $(\mathbf{R}_1 - \mathbf{R}_2)$
- Increment $J = J + 1$. Repeat from stage 1 until stopping conditions reached.
Output: $\mathbf{A}$, $\{\mathbf{B}_c\}_{c=1}^C$ and $\{\mathbf{X}_{con}^c\}_{c=1}^C$.

Figure 1. The MRC algorithm for both linear and non-linear cases.

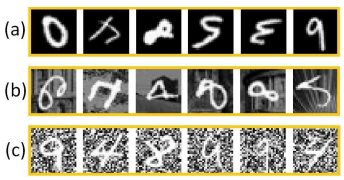

Figure 2. Sample digits from the rotated MNIST dataset. (a) Digits with random rotations, (b) Digits with random rotations and image backgrounds, (c) Digits with random backgrounds.

case, the dimension of the latent space can be higher than the dimension of the input space and is only upper bounded by the number of training samples. Figure 1 summarizes the MRC algorithm for both linear and kernel cases.

7. Experimental Results

In this section, we evaluate our proposed algorithm on several challenging datasets including rotated MNIST digits [19], Caltech-101 objects [28], Caltech-256 [12], SHREC'11 non-rigid 3D shapes [22]. Given a novel test sample $\mathbf{z} \in \mathbb{R}^n$, we transform it to the latent space using (31). The classification is done as explained in section 2. In particular, sparse coding is performed on the transformed signal separately using class-specific dictionaries $\{\mathbf{D}_c\}_{c=1}^{C}$. The test sample is classified to the class with the smallest residual error. We also analyse and compare our method with the state-of-the-art. For all the experiments in this section, the maximum number of iteration in Figure 1 is set to 80. Parameters selection is done by a 5-fold cross-validation.

7.1. Rotated MNIST

The rotated MNIST benchmark [19] contains gray scale images of hand-written digits of size 28×28 pixels. The images were originally taken from the MNIST dataset introduced in [21], and transformed in several ways to create more challenging classification problems. In the first dataset, called the *mnist-rot*, digits are rotated by random angles generated uniformly between 0 and 2π radians. The second dataset, called the *mnist-rot-back-image*, is created by inserting random backgrounds into *mnist-rot* dataset. The *mnist-back-rand* dataset is created by inserting random backgrounds in the original MNIST digit images. For all 3 datasets, there are 10000, 2000, and 50000 images for training, validation, and testing, respectively. Figure 2 shows sample images from the above datasets.

We learn MRC using all the training images and validation images. The parameters setting is as follows: $d = 200$, $T_0 = 15$, $K_c = 500\ \forall c \in \{1, \ldots, C\}$. Figure 3 displays the transformations learned by MRC on the *mnist-rot* dataset using a linear kernel. Each subplot of Figure 3 corresponds to a row of the matrix $\mathbf{W} = \mathbf{A}^T\mathbf{Y}^T$. They have a strong similarity to circular harmonic functions, thus, can capture more rotational invariant features. These transformations make a good sense given that the dataset consists a lot of variation along the circular direction. A polynomial kernel of degree 4 is used for classification since it usually gives about 2% improvement over the accuracy of the linear MRC. Classification performances and comparison are shown in table 1.

In all cases, MRC performances compare favourably to the state-of-the-art. MRC performs significantly better than all other methods for the last two datasets. While SRC does reasonably well for the first dataset, it performs poorly on the last two datasets which contain backgrounds. This is because it operates on the original data which are heavily corrupted by the insertion of backgrounds. In contrast, MRC works better because it can captures the discriminative and compact structure of data that are more robust against noise.

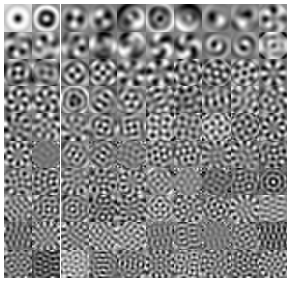

Figure 3. Example of transformations learned by the max residual classifier from the rotated MNIST dataset.

Dataset	SVM [38]	SRC[41]	KSVD [1]	FDDL [45]	MRC
(a)	88.89	85.05	86.74	89.42	88.94
(b)	44.82	29.52	42.62	44.18	45.25
(c)	85.42	72.41	82.94	85.89	87.25

Table 1. Comparison of recognition accuracy (%) on the rotated MNIST datasets: (a) *mnist-rot*, (b) *mnist-rot-back-image*, (c) *mnist-back-rand*.

7.2. Caltech-101

We perform the second set of experiments on the Caltech-101 dataset [28]. This dataset consists of 101 object classes, and 1 background class collected from Internet. The database contains a diverse and challenging set of images from buildings, musical instruments, animals and natural scenes, etc. A combination of 39 descriptors as in [10] is used to represent an image. We follow the suggested protocol in [20, 15], namely, we train on m images, where $m \in \{5, 10, 15, 20, 25, 30\}$, and test on the rest. The corresponding parameters settings of MRC are: $T_0 = \{3, 4, 5, 7, 8, 9\}$, $d = \{5, 10, 15, 20, 25, 30\}$, $K_c = d\ \forall c = \{1, \ldots, C\}$. To compensate for the variation of the class size, we normalize the recognition results by the number of test images to get per-class accuracies. The final recognition accuracy is then obtained by averaging per-class accuracies across 102 categories. Table 2 shows our classification accuracy in comparison with the state-of-the-art.

This dataset is challenging because the number of training samples is small, making it more difficult for learning, especially when signals are high-dimensional. This is because, with a fixed number of training sample, the generalization of learning reduces as the dimension increases. In contrast to other discriminative sparse repre-

# train samples	5	10	15	20	25	30
Irani [5]	-	-	65.0	-	-	70.4
Griffin [12]	44.2	54.5	59.0	63.3	65.8	67.6
Lazebnik [20]	-	-	56.4	-	-	64.6
Malik [46]	46.6	55.8	59.1	62.0	-	66.2
Yang [44]	-	-	67.0	-	-	73.2
Wang [39]	51.15	59.77	65.43	67.74	70.16	73.44
SRC [41]	48.8	60.1	64.9	67.7	69.2	70.7
KSVD [1]	49.8	59.8	65.2	68.7	71.0	73.2
D-KSVD [47]	49.6	59.5	65.1	68.6	71.1	73.0
LC-KSVD [16]	54.0	63.1	67.7	70.5	72.3	73.6
MRC	54.5	64.1	69.3	73	75.8	77.5

Table 2. Comparison of recognition accuracy (%) on Caltech-101.

sentation methods, MRC allows transformation of data to a low-dimensional latent space to deal with this problem. In addition, MRC is able to integrate multiple descriptors with non-linear distance measures. These are the reasons for why MRC outperforms other sparse approaches, including SRC [41] and discriminative KSVD [47], by a significant margin. We note that better results on this dataset was reported in [42]. Their method uses multiple kernel learning which can be combined with MRC to improve the results further.

7.3. Caltech-256

We also repeated the same experiment on Caltech-256 dataset. Table 3 shows our classification results in comparison with the state of the art. MRC's performances compare favourably with other discriminative learning approaches for all training configurations. Better results on Caltech-256 were recently reported in [4]. However, we do not compare with these results since their method focuses on learning rich features from image intensities using hierarchical networks, while our goal is to design a better classifier.

# train samples	15	30
Griffin [12]	28.3	34.1
Gemert [36]	-	27.2
Yang [18]	34.4	41.2
Gehler [11]	34.6	45.8
MRC	36.2	47

Table 3. Comparison of recognition results on Caltech-256 dataset.

7.4. SHREC'11 Non-Rigid 3D Shapes

The availability of new modelling, digitizing, and visualizing 3D shapes has led to an explosion of interest in 3D shape recognition in recent years. The problem is challenging due to various factors including 3D articulation, non-rigid deformation, and occlusion. The common choices of 3D shape matching and retrieval algorithms are nearest neighbor and bipartite graph matching. On one hand, the performance of nearest neighbor is heavily dependent on the quality of shape descriptors, which are often corrupted under noise and occlusion. On the other hand, graph matching is often computationally expensive. To this end, we propose the use of MRC as an efficient method for classifying 3D shapes. We evaluate the algorithm on the SHREC'11 dataset. This is a large-scale 3D dataset consisting of 600 non-rigidly deformed shapes derived from 30 different objects. Figure 4 displays sample shapes with different articulations containing in this database.

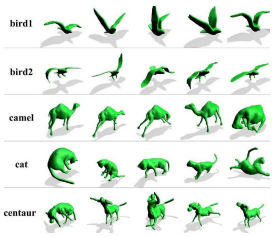

Figure 4. Sample shapes from the SHREC'11 dataset.

Method	Accuracy (%)
SVM [38]	93.7
SRC [41]	90.2
KSVD [1]	94.9
FDDL [45]	96.3
MRC	98.8

Table 4. Comparison of recognition accuracy on SHREC'11.

We choose to represent a 3D shape using a bag of graph distance histograms (GDH) [22]. Each GDH is computed at a random point on the 3D shape. The histogram is formed by quantizing graph distances from the selected point to a set of anchor points into 100 bins. For more details on the computation of GDH, please refer to [22]. Each shape in SHREC'11 dataset is represented with 1000 GDHs. 3 randomly selected shapes (i.e. 3000 GDHs) of each class are used for training, and the rest for testing. The parameters setting for learning MRC is as follows: $T_0 = 10$, $d = 400$, $K_c = 500\ \forall c = \{1, \ldots, C\}$, polynomial kernel of degree 4. Given a test sample, we first use MRC to classify 1000 GDHs into 30 classes. Each GDH constitutes a vote for one class. The final label is taken as the label of the class corresponding to the majority of 1000 votes. We repeat the same experiment with SVM, SRC, KSVD, and FDDL. Table 4 shows the performance of MRC in comparison with these classification methods.

8. Conclusion

In this paper, we introduced a simple yet efficient approach for learning discriminative sparse representation. Our approach allows discriminative decision function and sparse representation to be jointly optimized. Our classifier inherits the discriminative nature of margin-classifier and the robustness from sparse representation. Extensive experimental results on both 2D and 3D datasets have demonstrated the effectiveness of our method. Our classifier outperforms the current state-of-the-art despite its simplicity.

References

[1] M. Aharon, M. Elad, and A. M. Bruckstein. The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse represen-

tation. *IEEE Trans. Signal Process.*, 2006. 1, 3, 6, 7

[2] H. Barlow et al. Single units and sensation: a neuron doctrine for perceptual psychology. *Perception*, 1(4):371–394, 1972. 1

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. 4

[4] L. Bo, X. Ren, and D. Fox. Multipath sparse coding using hierarchical matching pursuit. In *CVPR*, June 2013. 7

[5] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, June 2008. 7

[6] D. L. Donoho. Neighborly polytopes and sparse solutions of underdetermined linear equations. Technical report, 2005. 1

[7] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Processing*, 15(12):3736–3745, 2006. 1

[8] K. Engan, S. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999. Proceedings.*, volume 5, pages 2443–2446. IEEE, 1999. 3

[9] D. Field et al. Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Am. A*, 4(12):2379–2394, 1987. 1

[10] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221 –228, 29 2009-oct. 2 2009. 6

[11] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221–228, 2009. 7

[12] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. *Technical Report*, 2007. 6, 7

[13] H. K. R. Grosse and A. Y. Ng. Shift-invariant sparse coding for audio classification. In *Proceedings of the Twenty-third Conference on Uncertainty in Artificial Intelligence*, 2007. 2, 3

[14] A. Hyvärinen, J. Hurri, and P. Hoyer. *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision.*, volume 39. Springer, 2009. 1

[15] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In *CVPR*, pages 1 –8, June 2008. 6

[16] Z. Jiang, Z. Lin, and L. Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *CVPR*, pages 1697 –1704, June 2011. 7

[17] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. Le-Cun. Learning invariant features through topographic filter maps. In *CVPR*, pages 1605–1612. IEEE, 2009. 2, 3

[18] N. Kulkarni and B. Li. Discriminative affine sparse codes for image classification. In *CVPR*, pages 1609 –1616, June 2011. 7

[19] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pages 473–480. ACM, 2007. 6

[20] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, volume 2, pages 2169 – 2178, 2006. 6, 7

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6

[22] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H. Van Nguyen, R. Ohbuchi, et al. Shrec'11 track: Shape retrieval on non-rigid 3d watertight meshes. *3DOR*, 11:79–88, 2011. 6, 7

[23] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010. 1, 3

[24] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *CVPR*, pages 1–8. IEEE, 2008. 2

[25] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. *CoRR*, abs/0809.3083, 2008. 2

[26] B. Olshausen et al. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. 1

[27] Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pages 40–44. IEEE, 1993. 2

[28] P. Perona, R. Fergus, and F. F. Li. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Workshop on Generative Model Based Vision*, page 178, 2004. 6

[29] D. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition. In *CVPR*, pages 1–8. IEEE, 2008. 2

[30] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine learning*, pages 759–766. ACM, 2007. 2, 3

[31] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, pages 3501–3508. IEEE, 2010. 2, 3

[32] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun. Efficient learning of sparse representations with an energy-based model. *NIPS*, 2006. 1

[33] L. Rosasco, A. Verri, M. Santoro, S. Mosci, and S. Villa. Iterative projection methods for structured sparsity regularization. *MIT Technical Reports, MIT-CSAIL-TR-2009-050, CBCL-282*, 2009. 4

[34] S. Roth and M. Black. Fields of experts: A framework for learning image priors. In *CVPR*, volume 2, pages 860–867. IEEE, 2005. 1

[35] A. Shrivastava, H. Nguyen, V. Patel, and R. Chellappa. Design of non-linear discriminative dictionaries for image classification. 1, 2

[36] J. C. van Gemert, J. M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. In *ECCV*, pages III: 696–709, 2008. 7

[37] J. Van Hateren. A theory of maximizing sensory information. *Biological cybernetics*, 68(1):23–29, 1992. 1

[38] V. Vapnik. *The nature of statistical learning theory*. springer, 1999. 1, 6, 7

[39] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, pages 3360 –3367, June 2010. 7

[40] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *ICCV*, volume 2, pages 1800–1807. IEEE, 2005. 1

[41] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *TPAMI*, 31(2):210–227, 2009. 1, 2, 3, 6, 7

[42] J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao. Group-sensitive multiple kernel learning for object categorization. In *CVPR*, pages 436–443. IEEE, 2009. 7

[43] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801. IEEE, 2009. 2, 3

[44] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794 –1801, June 2009. 7

[45] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *ICCV*, pages 543–550. IEEE, 2011. 2, 4, 6, 7

[46] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*. 7

[47] Q. Zhang and B. Li. Discriminative K-SVD for dictionary learning in face recognition. In *CVPR*, pages 2691 –2698, June 2010. 2, 7