Sparse Embedding: A Framework For Sparsity Promoting Dimensionality Reduction

Hien V. Nguyen[1], Vishal M. Patel[1], Nasser M. Nasrabadi[2], Rama Chellappa[1]

[1] University of Maryland, College Park, MD
[2] U.S. Army Research Laboratory, Adelphi, MD

Abstract. We introduce a novel framework, called *sparse embedding* (SE), for simultaneous dimensionality reduction and dictionary learning. We formulate an optimization problem for learning a transformation from the original signal domain to a lower-dimensional one in a way that preserves the sparse structure of data. We propose an efficient optimization algorithm and present its non-linear extension based on the kernel methods. One of the key features of our method is that it is computationally efficient as the learning is done in the lower-dimensional space and it discards the irrelevant part of the signal that derails the dictionary learning process. Various experiments show that our method is able to capture the meaningful structure of data and can perform significantly better than many competitive algorithms on signal recovery and object classification tasks.

1 Introduction

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. As an example of the effect of dimension on learning, Stone [1] showed that, under certain regularity assumption including that samples are identically independent distributed, the optimal rate of convergence for non-parametric regression decreases exponentially with the dimension of the data. As the dimension increases, the Euclidean distances between feature vectors become closer to each other making the inference task harder. This is known as the concentration phenomenon [2]. To address these issues, various linear and non-linear dimensionality reduction (DR) techniques have been developed (see [3] and references therein). In general, these techniques map data to a lower-dimensional space such that non-informative or irrelevant information in the data are discarded.

Recently, there has been an explosion of activities in modeling a signal using appropriate sparse representations (see [4] and references therein). This approach is motivated by the observation that most signals encountered in practical applications are compressible. In other words, their sorted coefficient magnitudes in some basis obey power law decay. For this reason, signals can be well-approximated by linear combinations of a few columns of some appropriate basis or dictionary $\mathbf{D}$. Although predefined basis such as wavelets or Fourier basis give rather good performances in signal compression, it has been shown that dictionaries learned directly from data can be more compact leading to better performances in many important tasks such as reconstruction and classification [5–7].

However, existing algorithms for finding a good dictionary have some drawbacks. The learning of $\mathbf{D}$ is challenging due to the high dimensional nature of the training data, as well as the lack of training samples. Therefore, DR seems to be a natural solution. Unfortunately, the current DR techniques are not designed to respect and promote underlying sparse structures of data. Therefore, they cannot help the process of learning the dictionary $\mathbf{D}$. Note that the recently developed DR technique [8, 9] based on the sparse linear model is also not suitable for the purpose of sparse learning since it assumes that the dictionary is given. Ideally, we want an algorithm that can discard non-informative part of the signal and yet does not destroy the useful sparse structures present in the data.

The second disadvantage of the existing sparse learning framework is its inability to handle sparse signals within non-linear models. Linear models used for learning $\mathbf{D}$ are often inadequate to capture the non-linear relationships within the data that naturally arise in many important applications. For example, in [10–12] it has been shown that by taking into account non-linearity, one can do better in reconstruction and classification. In addition, spatial pyramid [13], a popular descriptor for object and scene classification, and region of covariance [14], a popular descriptor for object detection and tracking, both have non-linear distance measures thus making the current sparse representation inappropriate.

In this paper, we propose a novel framework, called *sparse embedding* (SE), that brings the strength of both dimensionality reduction and sparse learning together. In this framework, the dimension of signals is reduced in a way such that the sparse structures of signals are promoted. The algorithm simultaneously learns a dictionary in the reduced space, yet, allows the recovery of the dictionary in the original domain. This empowers the algorithm with two important advantages: 1) Ability to remove the distracting part of the signal that negatively interferes with the learning process, and 2) Learning in the reduced space with smaller computational complexity. In addition, our framework is able to handle sparsity in non-linear models through the use of Mercer kernels.

1.1 Notations

Vectors are denoted by bold lower case letters and matrices by bold upper case letters. The ℓ_0-pseudo-norm, denoted by $\|\|_0$, is defined as the number of nonzero elements in a vector. The Frobenius norm of a matrix $\mathbf{X}$ in $\mathbb{R}^{n\times m}$ is defined as $\|\mathbf{X}\|_F^2 = (\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{X}(i,j)^2)^{1/2}$. We denote the dimension of input signal by n, output signals by d, dictionary size by K. *Input space* (in $\mathbb{R}^n$) refers to the ambient space of the original input signal. *Feature space* (in $\mathbb{R}^{\tilde{n}}$) indicates the high dimensional space of the signal after being transformed through some mapping Φ. *Reduced space* and *output space* (in $\mathbb{R}^d$) are used interchangeably to refer to the space of output signals after dimensionality reduction.

2 Sparse Embedding Framework

The classical approach to learn sparse representations [15] is by minimizing the reconstruction error over a finite set of signals subject to some sparsity constraint. Let $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N] \in \mathbb{R}^{n \times N}$ denotes the matrix of N input signals, where $\mathbf{y}_i \in \mathbb{R}^n$. A popular formulation is:

$$\{\mathbf{D}^*, \mathbf{X}^*\} = \underset{\mathbf{D},\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{DX}\|_F^2, \quad \text{subject to: } \|\mathbf{x}_i\|_0 \leq T_0, \forall i \tag{1}$$

where $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K] \in \mathbb{R}^{n \times K}$ is called the dictionary that we seek to learn, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{K \times N}$ is the sparse representation of $\mathbf{Y}$ over $\mathbf{D}$, and T_0 is the sparsity level. The cost function in (1) promotes a dictionary $\mathbf{D}$ that can best represent $\mathbf{Y}$ by linearly combining only a few of its columns. This type of optimization can be done efficiently using the current methods [12, 15].

Different from the classical approaches, we develop an algorithm that embeds input signals into a low-dimensional space, and simultaneously learns an optimized dictionary. Let $\mathcal{M}$ denote the mapping that transforms input signals into the output space. In general, $\mathcal{M}$ can be non-linear. However, for simplicity of notations, we temporarily restrict our discussions to linear transformations. The extension to the non-linear case will be presented in section 4. As a result, the mapping $\mathcal{M}$ is characterized using a matrix $\mathbf{P} \in \mathbb{R}^{d \times n}$. We can learn the mapping together with the dictionary through minimizing some appropriate cost function $\mathcal{C}_{\mathbf{Y}}$:

$$\{\mathbf{P}^*, \mathbf{D}^*, \mathbf{X}^*\} = \underset{\mathbf{P},\mathbf{D},\mathbf{X}}{\operatorname{argmin}} \, \mathcal{C}_{\mathbf{Y}}(\mathbf{P}, \mathbf{D}, \mathbf{X}) \tag{2}$$

This cost function $\mathcal{C}_{\mathbf{Y}}$ needs to have several desirable properties. First, it has to promote sparsity within the reduced space. At the same time, the transformation $\mathbf{P}$ resulting from optimizing $\mathcal{C}_{\mathbf{Y}}$ must preserve the useful information present in original signals. The second criterion is needed in order to prevent the pathological case of mapping into the origin, which obtains the sparsest solution but is obviously of no interest. Towards this end, we propose the following optimization:

$$\{\mathbf{P}^*, \mathbf{D}^*, \mathbf{X}^*\} = \underset{\mathbf{P},\mathbf{D},\mathbf{X}}{\operatorname{argmin}} \left(\|\mathbf{PY} - \mathbf{DX}\|_F^2 + \lambda \|\mathbf{Y} - \mathbf{P}^T\mathbf{PY}\|_F^2\right) \quad \text{subject to: } \mathbf{PP}^T = \mathbf{I}, \text{ and } \|\mathbf{x}_i\|_0 \leq T_0, \, \forall i \tag{3}$$

where $\mathbf{I} \in \mathbb{R}^{d \times d}$ is the identity matrix, λ is a non-negative constant, and the dictionary is now in the reduced space, i.e., $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K] \in \mathbb{R}^{\mathbf{d} \times K}$. The first term of the cost function promotes sparsity of signals in the reduced space. The second term is the amount of energy discarded by the transformation $\mathbf{P}$, or the difference between low-dimensional approximations and the original signals. In fact, the second term is closely related to PCA as by removing the first term in (3), it can be shown that the solution of $\mathbf{P}$ coincides with the principal components of the largest eigenvalues, when the data are centered.

In addition, we also require rows of $\mathbf{P}$ to be orthogonal and normalized to unit norm. $\mathbf{P}$ plays the role of selecting the right subspace, or equivalently the right features, in which the useful sparse structures within data are revealed. There are several compelling reasons to keep the orthogonality constraint. First, this constraint leads to a computationally efficient scheme for optimization and classification. Second, it allows the extension of SE to the non-linear case. Note that the columns of dictionary $\mathbf{D}$ still form a non-orthogonal basis in the output space despite the orthogonality constraint of $\mathbf{P}$.

3 Optimization Procedure

Proposition 1 *There exists an optimal solution* $\mathbf{P}^*$ *and* $\mathbf{D}^*$ *to (3), for sufficiently large* λ*, that has the following form:*

$$\mathbf{P}^* = (\mathbf{YA})^T; \quad \mathbf{D}^* = \mathbf{A}^T\mathbf{Y}^T\mathbf{YB} \tag{4}$$

for some $\mathbf{A} \in \mathbb{R}^{N\times d}$*, and some* $\mathbf{B} \in \mathbb{R}^{N\times K}$*. Moreover,* $\mathbf{P}^*$ *and* $\mathbf{D}^*$ *have minimum Frobenius norm among all optimal solutions of* $\mathbf{P}$ *and* $\mathbf{D}$*, respectively.*

Proof. See the appendix in the attachment.

As a corollary of the proposition 1, it is sufficient to seek an optimal solution for the optimization in Eq. (3) through $\mathbf{A}$ and $\mathbf{B}$. By substituting Eq. (4) into Eq. (3), we have:

$$\mathcal{C}_{\mathbf{Y}}(\mathbf{P},\mathbf{D},\mathbf{X}) = \|\mathbf{A}^T\mathbf{K}(\mathbf{I}-\mathbf{BX})\|_F^2 + \lambda\|\mathbf{Y}(\mathbf{I}-\mathbf{AA}^T\mathbf{K})\|_F^2 \tag{5}$$

where $\mathbf{K} = \mathbf{Y}^T\mathbf{Y}$ and λ is a regularization parameter. The equality constraint becomes

$$\mathbf{PP}^T = \mathbf{A}^T\mathbf{KA} = \mathbf{I} \tag{6}$$

The solution can be derived as

$$\{\mathbf{A}^*,\mathbf{B}^*,\mathbf{X}^*\} = \underset{\mathbf{A},\mathbf{B},\mathbf{X}}{\operatorname{argmin}} \left(\|\mathbf{A}^T\mathbf{K}(\mathbf{I}-\mathbf{BX})\|_F^2 + \lambda\|\mathbf{Y}(\mathbf{I}-\mathbf{AA}^T\mathbf{K})\|_F^2\right) \tag{7}$$
$$\text{subject to: } \mathbf{A}^T\mathbf{KA} = \mathbf{I}, \text{ and } \|\mathbf{x}_i\|_0 \le T_0$$

The advantage of this formulation will become clear later. Basically, this formulation allows a joint update of $\mathbf{P}$ and $\mathbf{D}$ via $\mathbf{A}$. As we shall see later in section 4, because the objective function is not explicitly represented in terms of $\mathbf{Y}$, it is then possible to use the kernel method to make the algorithm non-linear. Despite (7) being non-convex, our experiments show that effective solutions can be found through iterative minimization.

3.1 Solving for A

In this stage, we assume that $(\mathbf{B}, \mathbf{X})$ are fixed. As a result, we can remove the sparsity constraint of (7). The following proposition shows that $\mathbf{A}$ can be solved efficiently after some algebraic manipulation:

Proposition 2 *The optimal solution of (7) when* $\mathbf{B}$ *and* $\mathbf{X}$ *are fixed is:*

$$\mathbf{A}^* = \mathbf{V}\mathbf{S}^{-\frac{1}{2}}\mathbf{G}^* \tag{8}$$

where $\mathbf{V}$ *and* $\mathbf{S}$ *come from the eigendecomposition of* $\mathbf{K} = \mathbf{V}\mathbf{S}\mathbf{V}^T$*, and* $\mathbf{G}^* \in \mathbb{R}^{N \times d}$ *is the optimal solution of the following minimum eigenvalues problem:*

$$\{\mathbf{G}^*\} = \underset{\mathbf{G}}{\mathit{argmin}} \quad \mathbf{tr}\left[\mathbf{G}^T\mathbf{H}\mathbf{G}\right] \tag{9}$$
$$\textit{subject to: } \mathbf{G}^T\mathbf{G} = \mathbf{I}$$

where $\mathbf{H} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^T((\mathbf{I} - \mathbf{B}\mathbf{X})(\mathbf{I} - \mathbf{B}\mathbf{X})^T - \lambda\mathbf{I})\mathbf{V}\mathbf{S}^{\frac{1}{2}} \in \mathbb{R}^{N \times N}$.

Proof. The cost function can be expanded as follows:

$$\mathcal{C}_{\mathbf{Y}}(\mathbf{P}, \mathbf{D}, \mathbf{X}) = \|\mathbf{A}^T\mathbf{K}(\mathbf{I} - \mathbf{B}\mathbf{X})\|_F^2 + \lambda\|\mathbf{Y}(\mathbf{I} - \mathbf{A}\mathbf{A}^T\mathbf{K})\|_F^2 \tag{10}$$
$$= \mathbf{tr}\left[(\mathbf{I} - \mathbf{B}\mathbf{X})(\mathbf{I} - \mathbf{B}\mathbf{X})^T\mathbf{K}^T\mathbf{Q}^T\mathbf{K} + \lambda(\mathbf{K} - 2\mathbf{K}^T\mathbf{Q}^T\mathbf{K} + \mathbf{K}^T\mathbf{Q}^T\mathbf{K}\mathbf{Q}\mathbf{K})\right] \tag{11}$$

where $\mathbf{Q} = \mathbf{A}\mathbf{A}^T \in \mathbb{R}^{N \times N}$. The constraint $\mathbf{A}^T\mathbf{K}\mathbf{A} = \mathbf{I}$ leads to the new constraint $\mathbf{A}\mathbf{A}^T\mathbf{K}\mathbf{A}\mathbf{A}^T = \mathbf{A}\mathbf{A}^T$ or $\mathbf{Q}\mathbf{K}\mathbf{Q}^T = \mathbf{Q}$. Using this equality constraint, and also notice that $\mathbf{tr}(\mathbf{K})$ is a constant, the objective function in (11) can be simplified to a more elegant form:

$$\tilde{\mathcal{C}}_{\mathbf{Y}}(\mathbf{P}, \mathbf{D}, \mathbf{X}) = \mathbf{tr}\left[((\mathbf{I} - \mathbf{B}\mathbf{X})(\mathbf{I} - \mathbf{B}\mathbf{X})^T - \lambda I)\mathbf{K}^T\mathbf{Q}^T\mathbf{K}\right] \tag{12}$$

With a simple change of variable $\mathbf{G} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^T\mathbf{A}$, and noting that $\mathbf{Q} = \mathbf{A}\mathbf{A}^T$, the cost function can be further simplified as:

$$\tilde{\mathcal{C}}_{\mathbf{Y}}(\mathbf{P}, \mathbf{D}, \mathbf{X}) = \mathbf{tr}\left[((\mathbf{I} - \mathbf{B}\mathbf{X})(\mathbf{I} - \mathbf{B}\mathbf{X})^T - \lambda I)\mathbf{V}\mathbf{S}^{\frac{1}{2}}(\mathbf{S}^{\frac{1}{2}}\mathbf{V}^T\mathbf{A})(\mathbf{A}^T\mathbf{V}\mathbf{S}^{\frac{1}{2}})\mathbf{S}^{\frac{1}{2}}\mathbf{V}^T\right]$$
$$= \mathbf{tr}\left[\mathbf{G}^T\mathbf{S}^{\frac{1}{2}}\mathbf{V}^T((\mathbf{I} - \mathbf{B}\mathbf{X})(\mathbf{I} - \mathbf{B}\mathbf{X})^T - \lambda I)\mathbf{V}\mathbf{S}^{\frac{1}{2}}\mathbf{G}\right] = \mathbf{tr}\left[\mathbf{G}^T\mathbf{H}\mathbf{G}\right] \tag{13}$$

On the other hand, the equality constraint can also be simplified as:

$$\mathbf{A}^T\mathbf{K}\mathbf{A} = \mathbf{A}^T\mathbf{V}\mathbf{S}^{\frac{1}{2}}\mathbf{S}^{\frac{1}{2}}\mathbf{V}^T\mathbf{A} = \mathbf{G}^T\mathbf{G} = \mathbf{I} \tag{14}$$

Eqs. (13) and (14) show that the original optimization in (7) is equivalent to (9), and the optimal solution $\mathbf{A}^*$ can be recovered as in (8), i.e., $\mathbf{A}^* = \mathbf{V}\mathbf{S}^{-\frac{1}{2}}\mathbf{G}^*$. Note that since $\mathbf{K}$ is a positive semidefinite matrix, the diagonal matrix $\mathbf{S}$ has non-negative entries. $\mathbf{S}^{-\frac{1}{2}}$ is obtained by setting non-zero entries along the diagonal of $\mathbf{S}$ to the inverse of their square root and keeping zero elements the same. This completes the proof.

3.2 Solving for B and X

In order to solve for $\mathbf{B}$, we keep $\mathbf{A}$ and $\mathbf{X}$ fixed. The second term in (7) disappears, and the objective function reduces to:

$$\|\mathbf{A}^T\mathbf{K}(\mathbf{I}-\mathbf{B}\mathbf{X})\|_F^2 = \mathbf{tr}\,\left(\mathbf{K}^T\mathbf{Q}\mathbf{K} - 2\mathbf{X}^T\mathbf{B}^T\mathbf{K}^T\mathbf{Q}\mathbf{K} + \mathbf{X}^T\mathbf{B}^T\mathbf{K}^T\mathbf{Q}\mathbf{K}\mathbf{B}\mathbf{X}\right) \quad (15)$$

A possible way of solving for $\mathbf{B}$ is by taking the derivative of the objective function with respect to $\mathbf{B}$ and setting it to zero, which yields:

$$-2(\mathbf{K}^T\mathbf{Q}\mathbf{K})\mathbf{X}^T + 2(\mathbf{K}^T\mathbf{Q}\mathbf{K})\mathbf{B}(\mathbf{X}\mathbf{X}^T) = 0 \quad (16)$$

$$\mathbf{B} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^\dagger \quad (17)$$

This is similar to the method of optimal direction (MOD) [16] updating step except that $\mathbf{B}$ is not the dictionary but its representation coefficients over the training data $\mathbf{Y}$.
It is also possible to update $\mathbf{B}$ using the KSVD algorithm [15]. First, let us rewrite the objective function (15) to a more KSVD-friendly form:

$$\|\mathbf{A}^T\mathbf{K}(\mathbf{I}-\mathbf{B}\mathbf{X})\|_F^2 = \|\mathbf{Z}-\mathbf{D}\mathbf{X}\|_F^2 \quad (18)$$

where $\mathbf{Z} = \mathbf{A}^T\mathbf{K} \in \mathbb{R}^{d\times N}$ is the set of output signals. We can solve for $\mathbf{B}$ in two steps. First, we apply the KSVD algorithm to learn a dictionary $\mathbf{D}_{ksvd}$ from $\mathbf{Z}$. Second, we try to recover $\mathbf{B}$ from $\mathbf{D}_{ksvd}$. From Proposition 1, it follows that the optimal dictionary has to be in the columns subspace of the input signals $\mathbf{Z}$. We can observe from the KSVD algorithm that its output dictionary also obeys this property. As a result, we can recover $\mathbf{B}$ exactly by simply taking the pseudo-inverse:

$$\mathbf{B} = (\mathbf{Z})^\dagger\mathbf{D}_{ksvd} \quad (19)$$

In this paper, we choose a KSVD-like updating strategy because it is more computationally efficient. Our experiments show that both updating strategies produce similar performances for applications like object classification.

Sparse coding that solves for $\mathbf{X}$ can be done by any off-the-shelves pursuit algorithms. We use the orthogonal matching pursuit (OMP) [17] due to its high efficiency and effectiveness. Note that sparse coding is the most expensive step in many dictionary learning algorithms. Kernel KSVD [11] is an extreme example where sparse coding has to be done in the high-dimensional feature space. Our algorithm performs sparse coding in the reduced space leading to a huge computational advantage, yet, is capable of taking into account the non-linearity as we shall demonstrate in the next section.

4 Non-linear Extension of Sparse Embedding

There are many important applications of computer vision that deal with nonlinear data [13, 14]. Non-linear structures in data can be exploited by transforming the data into a high-dimensional feature space where they may exist as a

```
Input: A kernel matrix $\boldsymbol{\mathcal{K}}$, sparse setting $T_0$, dictionary size $K$, dimension $d$, and $\lambda$.
Task: Find $\mathbf{A}^*$ and $\mathbf{B}^*$ by solving Eq. (4).
Initialize:
- Set iteration $J = 1$. Perform eigendecomposition $\boldsymbol{\mathcal{K}} = \mathbf{VSV}^T$
- Set $\mathbf{A} = \mathbf{V}(:, \mathcal{I}_0)$, where $\mathcal{I}_0$ is the index set of $d$ largest eigenvalues of $\boldsymbol{\mathcal{K}}$
Stage 1: Dictionary Update
- Learn a dictionary $\mathbf{D}$ and $\mathbf{X}$ from the reduced signals $\mathbf{Z} = \mathbf{A}^T\boldsymbol{\mathcal{K}}$ using KSVD
- Update $\mathbf{B} = (\mathbf{Z})^{\dagger}\mathbf{D}$
Stage 2: Embedding update
- Compute $\mathbf{H} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^T((\mathbf{I} - \mathbf{BX})(\mathbf{I} - \mathbf{BX})^T - \lambda\mathbf{I})\mathbf{VS}^{\frac{1}{2}}$
- Perform eigendecomposition of $\mathbf{H} = \mathbf{U}\Lambda\mathbf{U}^T$
- Set $\mathbf{G} = \mathbf{U}(:, \mathcal{I}_J)$, where $\mathcal{I}_J$ is the index set of $d$ smallest eigenvalues of $\mathbf{H}$
- Update $\mathbf{A} = \mathbf{VS}^{-\frac{1}{2}}\mathbf{G}$
- Increment $J = J + 1$. Repeat from stage 1 until stopping conditions reached.
Output: $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{X}$.
```

Fig. 1. The SE algorithm for both linear and non-linear cases.

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 the use of Mercer kernels to extend our analysis to the non-linear case.

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

$$\boldsymbol{\mathcal{P}}^* = \mathbf{A}^T\Phi(\mathbf{Y})^T; \ \mathbf{D}^* = \mathbf{A}^T\boldsymbol{\mathcal{K}}\mathbf{B}. \tag{20}$$

Using Eq. (20), we can write the mapping $\mathcal{M}$ in an explicit form:

$$\mathcal{M} : \mathbf{s} \in \mathbb{R}^n \rightarrow \boldsymbol{\mathcal{P}}\Phi(\mathbf{s}) = \mathbf{A}^T\langle \Phi(\mathbf{Y}), \Phi(\mathbf{s}) \rangle_{\mathcal{H}} = \mathbf{A}^T[k(\mathbf{y}_1, \mathbf{s}), \ldots, k(\mathbf{y}_N, \mathbf{s})]^T \tag{21}$$

Similar to the linear case, the non-linear SE gives rise to the following cost function:

$$\|\boldsymbol{\mathcal{P}}\Phi(\mathbf{Y}) - \mathbf{DX}\|_F^2 + \lambda\|\Phi(\mathbf{Y}) - \boldsymbol{\mathcal{P}}^T\boldsymbol{\mathcal{P}}\Phi(\mathbf{Y})\|_{\mathcal{H}}^2 \tag{22}$$

which can be expressed in terms of $\mathbf{A}$ and $\mathbf{B}$ using Eq. (20), yielding an equivalent optimization:

$$\begin{aligned}\{\mathbf{A}^*, \mathbf{B}^*, \mathbf{X}^*\} = \underset{\mathbf{A},\mathbf{B},\mathbf{X}}{\operatorname{argmin}} \ \|\mathbf{A}^T\boldsymbol{\mathcal{K}}(\mathbf{I} - \mathbf{BX})\|_F^2 + \lambda\mathbf{tr}\left[(\mathbf{I} - \mathbf{AA}^T\boldsymbol{\mathcal{K}})^T\boldsymbol{\mathcal{K}}(\mathbf{I} - \mathbf{AA}^T\boldsymbol{\mathcal{K}})\right] \\ \text{subject to: } \mathbf{A}^T\boldsymbol{\mathcal{K}}\mathbf{A} = \mathbf{I}, \text{ and } \|\mathbf{x}_i\|_0 \leq T_0, \forall i \end{aligned} \tag{23}$$

The resulting optimization can be solved in the same way as in the linear case. Fig. 1 summarizes the SE algorithm. Note that in the non-linear case, the dimension of the output space can be higher than the dimension of the input space, and is only upper bounded by the number of training samples.

5 Experiments

In this section, we evaluate our algorithm on both synthetic and real datasets. In addition, we propose a novel classification scheme that leads to competitive performances on 3 different datasets: USPS, Caltech-101, and Caltech-256. We also analyse and compare our method with the state of the art. For all the experiments in this section, we set the maximum number of iteration J for our SE algorithm shown in Fig. 1 and that of the KSVD algorithm to 5 and 80, respectively.

5.1 Recovery of Dictionary Atoms

Similar to the previous works [15, 17], we first run our algorithm on a set of synthetic signals. The goal is to verify if our algorithm is able to learn the sparse patterns from a set of training data that comes with distortions.

Generation of Training Signals: Let $\mathbf{D} \in \mathbb{R}^{80\times 50}$ be a *generating dictionary.* The first 30 elements in each column of $\mathbf{D}$ are generated randomly, and the last 50 elements are set to zero. Each column of $\mathbf{D}$, which we will call a *dictionary atom*, is normalized to unit norm. 2000 training signals are generated by linearly combining 3 random atoms from this dictionary and superimposed with distortion signals:

$$\mathbf{Y} = \mathbf{D}\mathbf{X} + \alpha\mathbf{E} \tag{24}$$

where α is the distortion level; $\mathbf{X} \in \mathbb{R}^{50\times 2000}$ is a matrix where each of its column has at most 3 non-zero elements at independent locations with random values; $\mathbf{E} \in \mathbb{R}^{80\times 2000}$ is a matrix where each of its column is called a *distortion signal* and generated as follows: The first 30 elements in each column of $\mathbf{E}$ are set to zero, and the last 50 elements are generated randomly and independently under Gaussian distribution. Each column of $\mathbf{E}$ is also normalized to unit norm.

Our task is to recover $\mathbf{D}$ from $\mathbf{Y}$. We will first use SE to simultaneously reduce the dimension via $\mathbf{A}$ and learn a dictionary via $\mathbf{B}$. The original dictionary can be estimated as $\hat{\mathbf{D}} = \mathbf{Y}\mathbf{B}$. We compare our results with KSVD. To demonstrate the benefit of our proposed joint dimensionality reduction and dictionary learning, we also compare our results with the approach when the dimensionality reduction is done using PCA before learning the dictionary using KSVD.

Let $\mathbf{P}_{pca}$ represent the PCA transformation. We learn a dictionary, denoted $\mathbf{D}_{pca}$, using KSVD on the set of reduced signals $\mathbf{P}_{pca}\mathbf{Y}$. The original dictionary is recovered by first computing the coefficient matrix $\mathbf{B}_{pca} = (\mathbf{P}_{pca}\mathbf{Y})^{\dagger}\mathbf{D}_{pca}$, and then $\hat{\mathbf{D}} = \mathbf{Y}\mathbf{B}_{pca}$. Note that since the columns of $\mathbf{D}_{pca}$ are in the column subspace of $\mathbf{P}_{pca}\mathbf{Y}$, the computation of the coefficient matrix $\mathbf{B}_{pca}$ is exact.

Verification of Recovered Dictionaries: For all methods, the computed dictionary was compared against the generating dictionary. This comparison is done by sweeping through the columns of the generating dictionary to find the closest column (in ℓ_2 distance). The distance is measured by $\left(1 - |\mathbf{d}_i^T\hat{\mathbf{d}}_i|\right)$, where $\hat{\mathbf{d}}_i$ is the i-th estimated dictionary atom, and $\mathbf{d}_i$ is its closest atom in the original dictionary. A distance of less than 0.01 is considered a success. We learn dictionaries with sparsity level $T_0 = 3$, the dictionary size $K = 50$, and $\lambda = 1.1$.

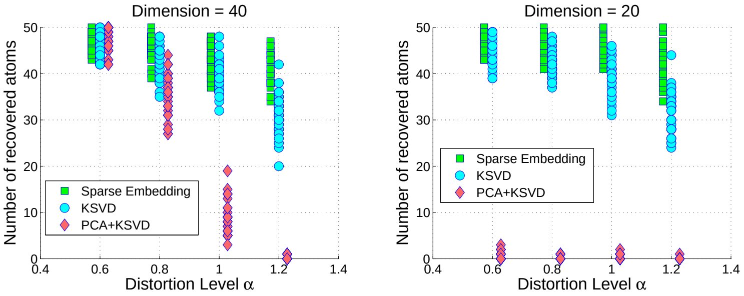


Fig. 2. Comparison of number of recovered dictionary atoms over 40 trials.

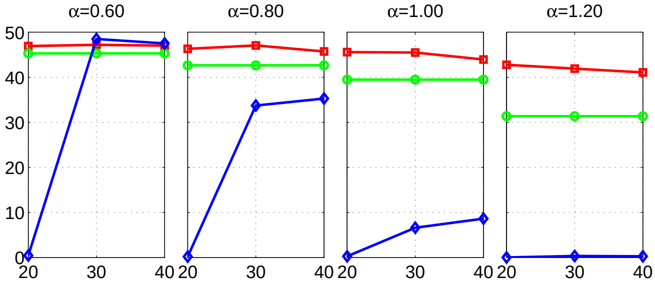


Fig. 3. Average number of successfully recovered dictionary atoms versus the dimension of the reduced space for different distortion level α. Blue color line corresponds to results for PCA+KSVD scheme, green color line for KSVD, and red color line for SE.

Fig. 2 compares the number of successes over 40 trials. Fig. 3 plots the average number of success versus the dimension of the output space for different distortion levels. SE consistently outperforms both KSVD and PCA+KSVD with larger and larger performance gaps as the distortion level increases. Fig. 3 shows that the performance of PCA+KSVD decreases drastically not only when the level of distortion level increases, but also when the dimension gets smaller than 30, which is the true dimension of our sparse signals. In contrast, SE outperforms KSVD even when the dimension goes below 30. Interestingly, for the high distortion level like $\alpha = 1.2$, it is beneficial to reduce the dimension to even below the true dimension of sparse signals (see the right most chart of Fig. 3).

In order to understand the reason behind the good results of SE, we visually inspect the transformation $\mathbf{P}$. Fig. 4 shows the images of $\mathbf{P}$ and the PCA mapping. The first 30 rows of the $\mathbf{P}$ weight heavily on the first 30 dimensions of $\mathbf{Y}$.

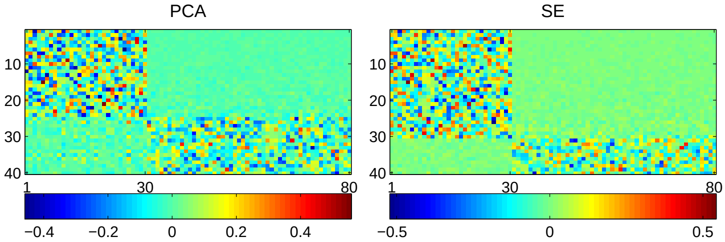


Fig. 4. Comparison of PCA mapping (left) and the transformation $\mathbf{P}$ learned from SE (right). Distortion level $\alpha = 1$, dimension of the reduced space is 40.

In other words, SE efficiently preserves sparse structures of signals and discards non-sparse distortions. In contrast, PCA does not preserve the sparse patterns since it is attempting to capture more of the signal variation. Only around 24 rows of PCA focus on the first 30 dimensions and the rest put more emphasis on non-sparse structures. Being able to get rid of distortions while preserving the sparse structures enables SE to achieve a higher recovery rate.

5.2 Latent Sparse Embedding Residual Classifier (LASERC)

Classification is an important component in many computer vision applications. In this section, we propose a novel classification scheme motivated by the SE framework. For generality, we will consider the non-linear setting. Let there be C different classes of signals $\{\mathbf{Y}_i = [\mathbf{y}^i_j]_{j=1}^{N_i} \in \mathbb{R}^{n \times N_i}\}_{i=1}^C$. We use the SE algorithm in Fig. 1 to learn $\{\mathbf{A}_i, \mathbf{B}_i\}_{i=1}^C$, which implicitly provides $\{\boldsymbol{\mathcal{P}}_i, \mathbf{D}_i\}_{i=1}^C$. Given a new test sample $\mathbf{s}_t$, the classification is done in three steps:
I) We compute output signals $\mathbf{z}_i$ by mapping $\mathbf{s}_t$ via $\mathcal{M}_i$,

$$\mathcal{M}_i : \ \mathbf{s}_t \rightarrow \mathbf{z}_i = \boldsymbol{\mathcal{P}}_i \Phi(\mathbf{s}_t) = \mathbf{A}_i^T \langle \Phi(\mathbf{Y}_i), \Phi(\mathbf{s}_t) \rangle_{\mathcal{H}} = \mathbf{A}_i^T \mathbf{k}_{i,t} \tag{25}$$

$$\text{where: } \mathbf{k}_{i,t} = [k(\mathbf{y}^i_1, \mathbf{s}_t), \ldots, k(\mathbf{y}^i_{N_i}, \mathbf{s}_t)]^T \in \mathbb{R}^{N_i} \tag{26}$$

II) We obtain the sparse code $\mathbf{x}_i$ for $\mathbf{z}_i$ over the dictionary $\mathbf{D}_i = \mathbf{A}_i^T \boldsymbol{\mathcal{K}}_i \mathbf{B}_i$ using the OMP algorithm, where

$$\boldsymbol{\mathcal{K}}_i = \langle \Phi(\mathbf{Y}_i), \Phi(\mathbf{Y}_i) \rangle_{\mathcal{H}} = [k(\mathbf{y}^i_j, \mathbf{y}^i_k)]_{j,k=1}^{N_i} \in \mathbb{R}^{N_i \times N_i} \tag{27}$$

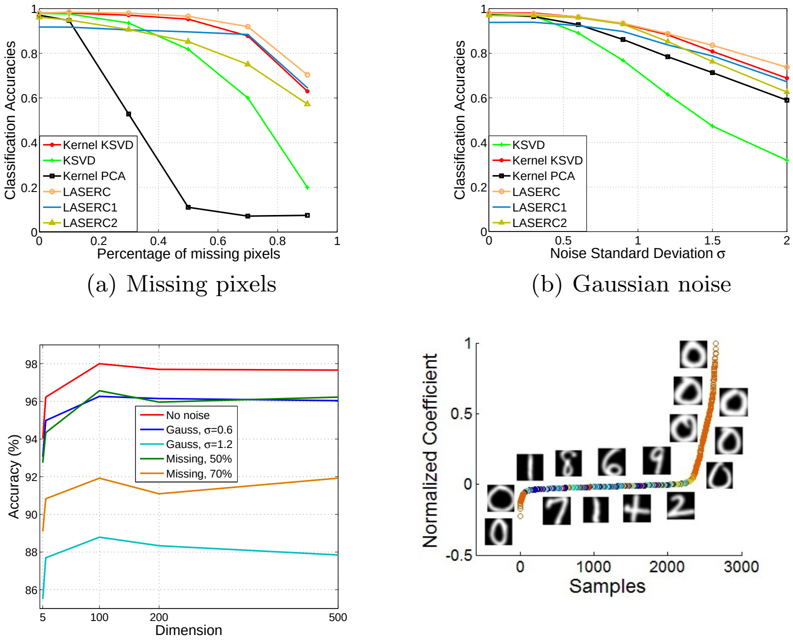
III) We compute the residual for each class as follows:

$$r_i = \|\Phi(\mathbf{s}_t) - \boldsymbol{\mathcal{P}}_i^T \mathbf{D}_i \mathbf{x}_i\|_{\mathcal{H}}^2 = k(\mathbf{s}_t, \mathbf{s}_t) - 2\mathbf{k}_{i,t}^T \mathbf{A}_i \mathbf{D}_i \mathbf{x}_i + \mathbf{x}_i^T \mathbf{D}_i^T \mathbf{A}_i^T \boldsymbol{\mathcal{K}}_i \mathbf{A}_i \mathbf{D}_i \mathbf{x}_i \tag{28}$$

Here, $\mathbf{D}_i \mathbf{x}_i$ is the estimated signal in the output space, and $\boldsymbol{\mathcal{P}}_i^T \mathbf{D}_i \mathbf{x}_i$ is the estimated signal in the feature space. The sparse coding step makes our algorithm more resilient to noise. Finally, each signal is assigned to the class that yields the smallest reconstruction residual. We call this *latent sparse embedding residual classifier* (LASERC). The term “latent” comes from the fact that the residual errors are computed in the feature space instead of the input space which does not take into account the non-linearities of the signal. The output space is also not suitable for classification because it does not retain sufficient discriminatory power due to its low-dimensional nature.

USPS Digit Recognition We apply our classifier on the USPS database which contains ten classes of 256-dimensional handwritten digits. A dictionary is learned for each class using samples from the training set with the following parameters setting: 500 dictionary atoms, $T_0 = 5$, $d = 100$, $\lambda = 1.1$, and the maximum number of iterations is set to 80. A polynomial kernel of degree 4 with the constant of value 1 is used for SE, kernel KSVD, and kernel PCA.

Our first experiment presents the results for the case when the pixels are randomly removed from the test images shown in Fig. 5(a), and when the test samples are corrupted by Gaussian noise with different standard deviations shown



(a) Missing pixels

(b) Gaussian noise

(c) Accuracy versus dimension

(d) Projection Coefficient

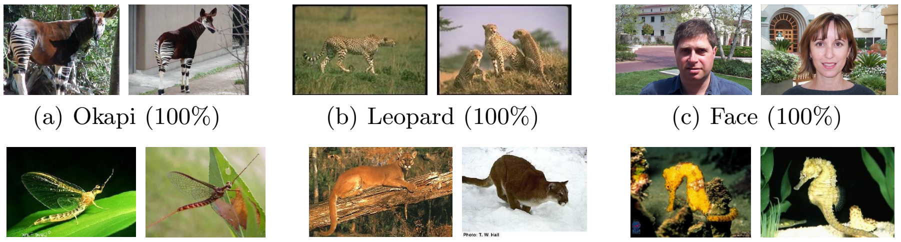
Fig. 5. (a,b) Comparison of classification accuracy against noise level. (c) Accuracy versus dimension. (d) Projection coefficient of all samples onto a dictionary atom.

in Fig. 5(b). In both scenarios, LASERC consistently outperforms kernel KSVD, linear KSVD, and kernel PCA. As the distortion level increases the performance differences between sparse embedding and linear KSVD become more drastic.

It is also worthwhile to investigate cases when the objective function in (7) has only the first term ($\lambda = 0$), denoted by LASERC1, and when there is only the second term ($\lambda \to \infty$), denoted by LASERC2. Fig 5(a) and 5(b) show that LASERC2 performs better for the low-noise cases and worse for the high-noise cases in comparison with LASERC1.

In order to see the effect of dimension on the classification performance of LASERC, we compare the results for different values of dimension $d = \{5, 10, 100, 200, 500\}$. The corresponding sparsity level is $T_0 = \{2, 3, 5, 5, 10\}$. Fig. 5(c) shows that the classification result improves as the dimension increases from 5 → 100. Beyond 100, the accuracy decreases slightly for the noiseless case, but faster for the very noisy cases like Gaussian noise with $\sigma = 1.2$.

We project test samples onto a random dictionary atom of the first class (digit 0). Fig. 5(d) plots the sorted projection coefficients of all the samples, color-coded by their class labels. We can observe from the plot 5(d) that, in the feature space, the dictionary atom is almost perpendicular to all samples except those from the first class (orange color at the two ends). This implies that the learned dictionary has taken into account the non-linearities of signals.



(a) Okapi (100%) (b) Leopard (100%) (c) Face (100%)

(d) Mayfly (20%) (e) Cougar (35%) (f) Seahorse (42%)

Fig. 6. Sample images from the classes corresponding to the highest accuracy (top row) and the lowest accuracy (bottom row) of LASERC.

# train samples	5	10	15	20	25	30
Malik [18]	46.6	55.8	59.1	62.0	-	66.2
Lazebnik [13]	-	-	56.4	-	-	64.6
Griffin [19]	44.2	54.5	59.0	63.3	65.8	67.6
Irani [20]	-	-	65.0	-	-	70.4
Yang [21]	-	-	67.0	-	-	73.2
Wang [22]	51.15	59.77	65.43	67.74	70.16	73.44
SRC [6]	48.8	60.1	64.9	67.7	69.2	70.7
KSVD [15]	49.8	59.8	65.2	68.7	71.0	73.2
D-KSVD [23]	49.6	59.5	65.1	68.6	71.1	73.0
LC-KSVD [24]	54.0	63.1	67.7	70.5	72.3	73.6
LASERC	**55.2**	**65.6**	**69.5**	**73.1**	**75.8**	**77.3**

# train samples	15	30
Griffin [19]	28.3	34.1
Gemert [25]	-	27.2
Yang [26]	34.4	41.2
LASERC	**35.2**	**43.6**

Time	Train (s)	Test (ms)
SVM	2.1	8.1
Ker. KSVD	2598	3578
SRC	N/A	520
D-KSVD	450	12.8
LASERC	**7.2**	**9.4**

Table 1. Comparison of recognition results on Caltech-101 dataset (left), recognition results on Caltech-256 dataset (upper right), and the computing time (lower right).

Caltech-101 and Caltech-256 Object Recognition We perform the second set of object recognition experiments on the Caltech-101 database [27]. This database comprises 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. We used the combination of 39 descriptors as in [28].

We follow the suggested protocol in [13, 29], 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 SE are: $T_0 = \{3, 4, 5, 7, 8, 9\}$, $d = \{5, 10, 15, 20, 25, 30\}$, and $\lambda = 1.1$. 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. We also repeat the same experiment on Caltech-256 dataset.

Table 1 shows the comparison of our classification accuracy with the state of the art. It is interesting that our method significantly outperforms the other discriminative approaches like LC-KSVD [24] and D-KSVD [23]. Thanks to the efficiency of DR, our method achieves a significant speed-up in the training process over the other sparse learning methods as shown in table 1. Fig. 6 shows sample images from the easiest classes as well as the most difficult classes. Fig. 7 shows the recognition accuracy per class, and Fig. 8 shows their confusion matrix.

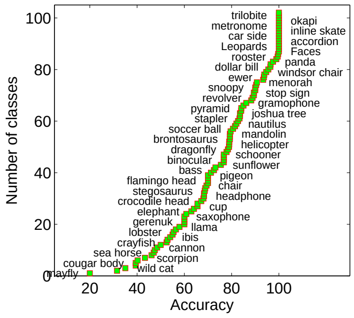


Fig. 7. Caltech-101 Per Class Accuracy

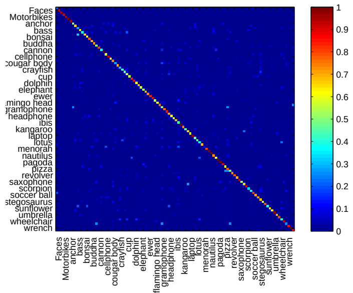


Fig. 8. Caltech-101 Confusion Matrix

6 Conclusions

This paper presented a novel framework for a joint dimensionality reduction and sparse learning. It proposes an efficient algorithm for solving the resulting optimization problem. It designs a novel classification scheme leading to a state-of-the-art performance and robustness on several popular datasets. Through extensive experimental results on real and synthetic data, we showed that sparse learning techniques can benefit significantly from dimensionality reduction in terms of both computation and accuracy.

Acknowledgement

The work of HVN, PVM and RC was partially funded by an ONR grant N00014-12-1-0124.

References

1. Stone, C.J.: Optimal global rates of convergence for nonparametric regression. The Annals of Statistics **10** (1982) 1040–1053
2. Beyer, Goldstein, Ramakrishnan, Shaft: When is "nearest neighbor" meaningful? In: ICDT: 7th International Conference on Database Theory. (1999)
3. Lee, J.A., Verleysen, M.: Nonlinear dimensionality reduction. In: Information Science and Statistics, Springer, 2006
4. Elad, M.: Sparse and redundant representations. In: From theory to applications in signal and image processing, Springer-New York. (2010)
5. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Processing **15**(12) (December 2006) 3736–3745
6. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. PAMI **31**(2) (Feb. 2009) 210–227
7. Ramírez, I., Sprechmann, P., Sapiro, G.: Classification and clustering via dictionary learning with structured incoherence and shared features. In: CVPR, IEEE (2010) 3501–3508

8. Gkioulekas, I., Zickler, T.: Dimensionality reduction using the sparse linear model. In: Advances in Neural Information Processing Systems (NIPS). (2011)
9. Zhang, L., Yang, M., Feng, Z., Zhang, D.: On the dimensionality reduction for sparse representation based face recognition. In: ICPR, IEEE (2010) 1237–1240
10. Qi, H., Hughes, S.: Using the kernel trick in compressive sensing: Accurate signal recovery from fewer measurements. In: IEEE ICASSP. (may 2011) 3940 –3943
11. Nguyen, H.V., Patel, V.M., Nasrabadi, N.M., Chellappa, R.: Kernel dictionary learning. IEEE Int. Conference on Acoustics, Speech and Signal Processing (2012)
12. Mairal, J., Bach, F., Ponce, J.: Task-driven dictionary learning. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34**(4) (april 2012) 791 –804
13. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: IEEE CVPR. Volume 2. (2006) 2169 – 2178
14. Tuzel, O., Porikli, F.M., Meer, P.: Region covariance: A fast descriptor for detection and classification. In: ECCV. (2006) II: 589–600
15. Aharon, M., Elad, M., Bruckstein, A.M.: The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**(11) (2006) 4311–4322
16. Engan, K., Aase, S.O., Husoy, J.H.: Multi-frame compression: Theory and design. Signal Processing **80**(10) (October 2000) 2121–2140
17. Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S.: Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. 27th Asilomar Conference on Signals, Systems and Computers (1993) 40–44
18. Zhang, H., Berg, A., Maire, M., Malik, J.: SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In: IEEE CVPR, 2006.
19. Grifn, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report (2007)
20. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: IEEE CVPR. (June 2008) 1 –8
21. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: IEEE CVPR. (June 2009) 1794 –1801
22. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: IEEE CVPR. (June 2010) 3360 –3367
23. Zhang, Q., Li, B.: Discriminative K-SVD for dictionary learning in face recognition. In: IEEE CVPR. (June 2010) 2691 –2698
24. Jiang, Z., Lin, Z., Davis, L.: Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In: CVPR. (June 2011) 1697 –1704
25. van Gemert, J.C., Geusebroek, J.M., Veenman, C.J., Smeulders, A.W.M.: Kernel codebooks for scene categorization. In: ECCV. (2008) III: 696–709
26. Kulkarni, N., Li, B.: Discriminative affine sparse codes for image classification. In: IEEE CVPR. (June 2011) 1609 –1616
27. Perona, P., Fergus, R., Li, F.F.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: Workshop on Generative Model Based Vision. (2004) 178
28. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: Computer Vision, 2009 IEEE 12th International Conference on. (29 2009-oct. 2 2009) 221 –228
29. Jain, P., Kulis, B., Grauman, K.: Fast image search for learned metrics. In: IEEE CVPR 2008. (June 2008) 1 –8