Local Sparse Coding for Image Classification and Retrieval

Jayaraman J. Thiagarajan, Karthikeyan Natesan Ramamurthy, Andreas Spanias

SenSIP Center, School of ECEE, Arizona State University, USA.

Abstract

The success of sparse representations in image modeling and recovery has motivated its use in computer vision applications. Object recognition has been effectively performed by aggregating sparse codes of local features in an image at multiple spatial scales. Though sparse coding guarantees a high-fidelity representation, it does not exploit the dependence between the local features. By incorporating suitable locality constraints, sparse coding can be regularized to obtain similar codes for similar features. In this paper, we develop an algorithm to design dictionaries for local sparse coding of image descriptors and perform object recognition using the learned dictionaries. Furthermore, we propose to perform kernel local sparse coding in order to exploit the non-linear similarity of features and describe an algorithm to learn dictionaries when the Radial Basis Function (RBF) kernel is used. In addition, we develop a supervised local sparse coding approach for image retrieval using sub-image heterogeneous features. Simulation results for object recognition demonstrate that the two proposed algorithms achieve higher classification accuracies in comparison to other sparse coding based approaches. By performing image retrieval on the Microsoft Research Cam-
bridge image dataset, we show that incorporating supervised information into local sparse coding results in improved precision-recall rates.

Keywords: Sparse coding, Local linear modeling, kernel trick, object recognition, image retrieval.

1. Introduction

Sparse coding allows for an efficient signal representation using a linear combination of elementary features. A finite collection of normalized features is referred to as a dictionary. The generative model for representing the data vector \( y \in \mathbb{R}^N \) using the sparse code \( x \in \mathbb{R}^K \) can be written as

\[
y = \Psi x + n, \tag{1}
\]

where \( \Psi \) is the dictionary of size \( N \times K \) and \( n \) is the noise component not represented using the sparse code. The system is usually underdetermined, \( N < K \), and the goal is to solve for the code \( x \) such that it is sparse, i.e. only few of its entries are significantly different from zero. Cost functions that are commonly used to measure sparsity include the \( \ell_0 \) and the \( \ell_1 \) norms. The error constrained \( \ell_1 \) minimization is given by

\[
\min_x \|x\|_1 \quad \text{subject to} \quad \|y - \Psi x\|_2 \leq \epsilon \tag{2}
\]

where, \( \| \cdot \|_1 \) refers to the \( \ell_1 \) norm and \( \epsilon \) is the error goal. Some of the algorithms used to solve (2) include the Basis Pursuit (Chen et al., 2001), Orthogonal Matching Pursuit (OMP), feature-sign search (Lee et al., 2007) and the least angle regression algorithm (Efron et al., 2004) with the lasso modification (LARS-LASSO). The dictionary \( \Psi \) when adapted to the data...
has been shown to provide superior performance when compared to prede-
fined dictionaries in several applications (Rubinstein et al., 2010; Thiagara-
jan et al., 2011). The joint optimization problem of dictionary learning and
sparse coding can be expressed as

$$\min_{\Psi, X} \parallel Y - \Psi X \parallel_F^2 \text{ s.t. } \parallel x_i \parallel_0 \leq s, \forall i, \parallel \psi_j \parallel_2 = 1, \forall j,$$

(3)

where $Y$ is the collection of $T$ training vectors, $X$ is the coefficient matrix,
s is the sparsity of the coefficient vector and $\parallel . \parallel_F$ denotes the Frobenius
norm. Note that this joint optimization problem is not convex and hence the
sparse codes and the dictionary elements are computed iteratively. As it can
be observed, the problem of dictionary learning is a generalization of data
clustering. For example, the K-SVD algorithm generalizes the K-hyperline
clustering procedure (Thiagarajan et al., 2010) to iteratively minimize the
objective in (3) (Aharon et al., 2006).

The effectiveness of sparse models in several image recovery applications
has motivated its use in computer vision problems. Though raw image
(patches) pixels can be directly employed for classification (Raina et al., 2007;
Wright et al., 2001), several successful frameworks for object recognition and
image retrieval often work with relevant features extracted from an image.
This is performed in order to obtain key information from the images and
to reduce the dimensionality. Sparse codes obtained from the features are
typically aggregated using the orderless bag of words or the state-of-the-art
spatial pyramid matching (SPM) technique which incorporates spatial order-
ing as well (Yang et al., 2009; Jiang et al., 2011; Yu and Zhang, 2009; Wang
et al., 2010; Lazebnik et al., 2006). Usually local descriptors, such as SIFT
or histogram of oriented gradients (HOG), extracted from small patches on
a dense image grid are used to construct spatial pyramids (SP). Incorporating locality constraints, when computing the sparse codes, has resulted in improved discrimination for object recognition (Yu and Zhang, 2009; Wang et al., 2010; Thiagarajan and Spanias, 2011). In this paper, this process is referred to as local sparse coding. In image retrieval applications, it is typical to use a heterogeneous combination of multiple global/local features extracted from an image (Frahm et al., 2010; Cao et al., 2009). In (Thiagarajan et al., 2012), it was shown that using sparse codes of heterogeneous features extracted from reasonably large overlapping regions of an image resulted in an improved image retrieval performance when compared to using local descriptors. We will refer to the large overlapping regions as sub-images. Note that these sub-images are much larger in size than patches used for object recognition.

In this paper, we consider the local sparse coding of image features and propose algorithms to perform object recognition and image retrieval. We pose local sparse coding as a weighted $\ell_1$ minimization problem and propose to design dictionaries using the local codes. In addition, we develop the kernel local sparse coding algorithm and present an approach to design dictionaries when RBF (radial basis function) kernels are used. We perform object recognition, using the two proposed algorithms, on the Caltech-256, Corel-10, Scene-15 and UICUC sports datasets. Note that we follow the approach in (Yang et al., 2009) to construct SP features and the classification is performed using a linear SVM. Simulation results show that the proposed algorithms achieve higher classification accuracies in comparison to other sparse coding based classification systems. Finally we perform im-
age retrieval by supervised local sparse coding of sub-image heterogeneous features. In the proposed approach, we perform simultaneous local sparse approximation for all sub-image features in a class and optimize the dictionary for the local codes. Simulations show that supervised local sparse coding provides improved image retrieval performance on the Microsoft Research Cambridge image database (Ulusoy and Bishop, 2005), when compared to not including the supervised information.

2. Sparse Coding in Image Classification

Though sparse representations are more likely to be separable in high-dimensional spaces, it has not been shown that sparsity can lead to improved recognition performance. In (Rigamonti et al., 2011) the authors argue, through empirical experiments for object recognition, that imposing sparsity does not provide any improvements, unless this sparsity is regularized to promote discrimination. Constraints from discriminative frameworks such as the linear discriminant analysis and linear Support Vector Machines (SVM) can be incorporated in (3) to obtain sparse codes for classification tasks. In other words, exploiting the correlations between different classes during dictionary design will improve the discrimination power of sparse codes. Algorithms that employ such constraints when learning the dictionary for digit recognition and image classification have been reported in (Mairal et al., 2009; Yang et al., 2009; Bradley and Bagnell, 2008; Jiang et al., 2011; Zhang and Li, 2010). Furthermore, approaches that promote discrimination by performing simultaneous sparse approximation of all training features in a class have been reported in (Thiagarajan et al., 2009; Bengio et al., 2009).
Several state-of-the-art object recognition systems are based on Spatial Pyramid Matching (SPM) (Lazebnik et al., 2006). As shown in the illustration of SPM in Figure 1, an image is partitioned into increasingly finer regions and features are evaluated in the local regions. Local descriptors are extracted from small image patches and they are coded using a dictionary (learned using features from several training images), and the code vectors in each spatial region are then pooled together by building histograms. Finally, the histograms of the different spatial regions are concatenated and presented to a non-linear classifier. Typically, the features extracted from the images are either Scale Invariant Feature Transform (SIFT) descriptors or Histogram of Oriented Gradients (HOG) descriptors. In Figure 1 K-means dictionary is used to code the descriptors, based on vector quantization. Though, this approach has been very effective, the complexity of using a non-linear classifier is quite high. Hence, the authors in (Yang et al., 2009) proposed to replace...
the vector quantization in SPM by sparse coding, which enabled the use of linear classifiers. Furthermore the authors of (Gao et al., 2010a) showed that the sparse representations can be efficiently performed in a high dimensional feature space using the kernel trick. The kernel trick maps the non-linear separable features into a high dimensional feature space, in which similar features are easily grouped together and linear separable (Shawe-Taylor and Cristianini, 2004).

Note that sparse coding does not explicitly consider the correlation between the sparse codes, which is crucial for classification tasks. Though sparse coding achieves lesser reconstruction error using multiple dictionary atoms, it might select different atoms for similar features. The overcomplete nature of the dictionary makes the sparse coding process highly sensitive to the variance in the features. This can be circumvented by suitably regularizing the sparse coding problem. Laplacian sparse coding (Gao et al., 2010b) addresses this by exploiting the dependence between the local features, thereby ensuring the consistence of the sparse codes. In (Wang et al., 2010), the authors proposed the locality-constrained linear coding approach, which explicitly encourages the code to have non-zero coefficients for dictionary atoms in the neighborhood of the encoded data. In (Yu and Zhang, 2009) it was showed theoretically that, under certain assumptions, locality is more important than sparsity for non-linear function learning using the obtained local codes.

2.1. Locality in Sparse Models

In several signal processing applications, we typically observe that the signals lying in a very high dimensional space often constitute an underlying
physical process of much lesser dimensionality. Some of the popular methods used for learning these low dimensional manifolds include ISOMAP, locally linear embedding (LLE) and Hessian LLE (Cayton, 2005). In particular, the LLE is an unsupervised learning algorithm which exploits the fact that, the local geometry of a non-linear function can be well approximated using a linear model. When the dictionary $\Psi$ represents the set of anchor points that characterize the local geometry, the idea of using sparse coding to model the local neighborhood of data merits attention. However, sparse coding, in the absence of additional constraints, tries to reduce the error of the representation without any consideration of locality. It is possible to include additional locality constraints by considering the general class of the weighted $\ell_1$ minimization problems,

$$
\min_{x} \sum_{k=1}^{K} w(k)|x_k| \quad \text{subject to} \quad \|y - \Psi x\|_2 \leq \epsilon,
$$

(4)

where $w(1), ..., w(K)$ are positive weights. It can be clearly seen that large weights could be used to encourage zero entries in the sparse code $x$. The LCC algorithm proposed in (Yu and Zhang, 2009) computes local sparse codes using weights based on the Euclidean distance measure as given by

$$
w(k) = \|y - \psi_k\|_2^2,
$$

(5)

where $\psi_k$ is the $k$th column of $\Psi$. Since the dictionary elements are normalized, the following distance metric can be alternatively employed to compute the neighborhood.

$$
w(k) = \|y - (y^T \psi_k)\psi_k\|_2^2.
$$

(6)

Note that this metric proposed in (Thiagarajan and Spanias, 2011) is used by K-hyperline clustering (Thiagarajan et al., 2010) to identify the membership
Figure 2: Comparison of the distance measures used in (5) and (6). In the figures, ① \(\|y - \psi_i\|_2\), ② \(\|y - (y^T \psi_i) \psi_i\|_2\), ③ \(\|y - \psi_j\|_2\), ④ \(\|y - (y^T \psi_j) \psi_j\|_2\).

of a data sample. In K-hyperline clustering, a cluster center is evaluated as the rank-1 SVD (singular value decomposition) of the data samples belonging to that cluster. This weighting scheme directly considers the coherence between the normalized dictionary elements and the data sample \(y\). As it can be clearly observed in Figure 2, the difference in the distances between a data sample and its two closest dictionary elements is higher with (6) in comparison to (5).

Since the weighted \(\ell_1\) minimization in (4) is computationally expensive, the approximate method for locality constrained linear coding (LLC) was proposed in (Wang et al., 2010). In general, the LLC algorithm employs the following criteria:

\[
\min_{\mathbf{x}} \sum_{i=1}^{T} \|y_i - \Psi \mathbf{x}_i\|_2^2 + \lambda \|\mathbf{w}_i \odot \mathbf{x}_i\|_2^2 \quad \text{s.t.} \quad \mathbf{1}^T \mathbf{x}_i = 1, \forall i, \tag{7}
\]

where \(\odot\) denotes element-wise multiplication and \(\mathbf{w}_i\) measures the similarity between the data sample and all the dictionary atoms. The distance metric
used is
\[
    w_i(k) = \exp \left( \frac{\|y_i - \psi_k\|^2}{\sigma} \right), \forall k,
\]
where \(\sigma\) is used to adjust the kernel weight in the neighborhood of a data sample. In order to speed up this procedure, the \(P\) nearest dictionary atoms are first identified and a smaller linear system is solved using a least squares procedure on the chosen dictionary atoms. This reduces the computational complexity from \(\mathcal{O}(K^2)\) to \(\mathcal{O}(K + P^2)\), where \(K\) denotes the number of dictionary atoms and \(P \ll K\).

3. Local Sparse Coding

In this section, we describe the proposed dictionary learning algorithm for local sparse coding of image descriptors. Similar to the dictionary learning approaches for regular sparse coding, the proposed algorithm iterates between local sparse coding and dictionary update steps. However, we need to take into account the neighborhood relation between a dictionary atom and the training vectors it represents, when updating the dictionary. The dictionary learning is a generalized clustering procedure where the training vectors are assigned to more than one dictionary atom in the local sparse coding step. Similarly, the training vectors participate in the update of more than one dictionary atom in the update step. The dictionary is initialized using cluster centroids obtained from K-means clustering of the training vectors. The joint optimization problem for local sparse coding and dictionary learning can be expressed as

\[
    \left\{ \hat{\Psi}, \{\hat{x}_i\}_{i=1}^T \right\} = \min_{\Psi, \{x_i\}_{i=1}^T} \sum_{i=1}^T \|y_i - \Psi x_i\|^2_2 + \lambda \sum_{i=1}^T \|w_i x_i\|_1.
\]
In order to solve for local sparse codes, we rewrite the weighted minimization problem in (9) as

\[ \hat{x}_i = \min_{x_i} \|y_i - \Psi W_i^{-1} x_i\|_2^2 + \lambda \|x_i\|_1, \forall i = 1 \text{ to } T. \]  

(10)

Here \( W_i \) is a diagonal matrix containing the weights corresponding to the dictionary elements in \( \Psi \) computed as

\[ W_i(k, k) = \|y_i - \psi_k\|_2^2, \forall k = 1 \text{ to } K. \]  

(11)

Setting \( \Gamma_i = \Psi W_i^{-1} \), (10) can be expressed as

\[ \hat{x}_i = \min_{x_i} \|y_i - \Gamma_i x_i\|_2^2 + \lambda \|x_i\|_1, \forall i = 1 \text{ to } T. \]  

(12)

This is equivalent to standard sparse coding with the modified dictionary \( \Gamma_i \). We can employ the search algorithm or the LARS-LASSO method to efficiently solve (12) for the sparse codes. Though these algorithms are significantly faster than using convex optimization, a greedy approach can be used to evaluate the local code for test data. For a test data sample \( y \), we first identify the \( P \) nearest neighbors and then compute the corresponding \( P \)-sparse code as

\[ \min_x \left\| y - \sum_{k \in C} \psi_k x_k \right\|_2 \quad \text{subject to} \quad \|x\|_0 \leq P, \]  

(13)

where \( C \) contains the set of indices of the \( P \) nearest dictionary atoms selected by the distance metric in (6). This optimization implies that only the selected dictionary atoms, \( \{\psi_k\}_{k \in C} \), can participate in the approximation of \( y \). This can be solved using the Orthogonal Matching Pursuit or the LARS algorithm.
3.1. Dictionary Learning

As described earlier, updating the dictionary atom should not affect its neighborhood relation with the training vectors it represents when trying to reduce the error. The non-zero coefficients in the local sparse code reveals the relationship between a data sample and the dictionary atoms. We can preserve the relationship by fixing the coefficients when the dictionary atoms are updated, such that the approximation error is reduced. This can be contrasted with the dictionary update of K-SVD, where the dictionary atom and the corresponding coefficient are updated together using an SVD procedure. Given the sparse codes and the weighting matrix, the dictionary update step can be expressed as

$$\hat{\Psi} = \min_{\Psi} \sum_{i=1}^{T} \| y_i - \Psi \hat{z}_i \|_2, \text{ s.t. } \forall k, \| \psi_k \|_2^2 = 1. \quad (14)$$

Here $$\hat{z}_i = W_i^{-1} \hat{x}_i$$ denotes the reweighted coefficient vector obtained from the local sparse coding step. In order to update the $$k$$th dictionary atom, we can simplify the objective function as

$$\left\| Y - \sum_{j=1}^{K} \psi_j \hat{z}_{j,r} \right\|_F^2 = \left\| Y - \left( \sum_{j \neq k} \psi_j \hat{z}_{j,r} \right) - \psi_k \hat{z}_{k,r} \right\|_F^2,$$

$$= \left\| E_k - \psi_k \hat{z}_{k,r} \right\|_F^2. \quad (15)$$

Here $$\hat{z}_{k,r}$$ denotes the row vector containing the set of coefficients of all training vectors corresponding to the dictionary atom $$\psi_k$$. The objective can be
rewritten as

\[ \|E_k - \psi_k \hat{z}_{k,r}\|^2_F = \text{tr}[(E_k - \psi_k \hat{z}_{k,r})^T(E_k - \psi_k \hat{z}_{k,r})], \]

\[ = \text{tr}[E_k^T E_k - E_k^T \psi_k \hat{z}_{k,r} - (\hat{z}_{k,r})^T \psi_k^T E_k + (\hat{z}_{k,r})^T \psi_k^T \psi_k \hat{z}_{k,r}]. \tag{16} \]

Since \( \psi_k^T \psi_k = 1 \), the optimization with respect to \( \psi_k \) will consider only the second and third terms. Since \( \text{tr}[A + A^T] = 2\text{tr}[A] \) and \( \text{tr}[AB] = \text{tr}[BA] \), the objective to be minimized can be simplified as

\[ \text{tr}
\left(-E_k^T \psi_k \hat{z}_{k,r}\right)
\]

\[ = \text{tr}
\left(-\hat{z}_{k,r} E_k^T \psi_k\right). \]

Hence, the dictionary atom \( \psi_k \) is updated as

\[ \hat{\psi}_k = \frac{E_k(\hat{z}_{k,r})^T}{\|E_k(\hat{z}_{k,r})^T\|_2}. \tag{17} \]

It can be observed that this dictionary update is equivalent to computing the weighted mean of the residual error vectors, where the weights are obtained using local sparse coding. Since the coefficients are fixed during the update, the dictionary atoms change slowly from the initial atoms over iterations.

4. Kernel Local Sparse Coding

It is well known that the kernel trick can be employed to capture the non-linear similarity of features. We propose to perform the local sparse coding technique in a high dimensional feature space, in order to improve the discrimination power of the codes. Let us define a non-linear transformation to a feature space \( \Phi \) as \( \Phi : \mathbb{R}^M \mapsto \mathbb{F} \). We denote the set of training vectors \( \mathbf{Y} \) in the feature space as \( \Phi(\mathbf{Y}) \). The kernel matrix \( \mathbf{K} \in \mathbb{R}^{N \times N} \) is a Gram matrix of all the feature vectors, \( \mathbf{K} = \Phi(\mathbf{Y})^T \Phi(\mathbf{Y}) \). All computations in the feature space will be performed using exclusively kernel similarities. In this paper,
we consider the radial basis function kernel to perform local sparse coding. Given two data samples $y_1$ and $y_2$, the kernel similarity can be computed as

$$K(y_1, y_2) = \exp(-\gamma(||y_1 - y_2||_2^2)).$$

The joint optimization of local sparse coding and dictionary learning can be performed in the feature space as

$$\{\hat{\Psi}, \{\hat{x}_i\}_{i=1}^T\} = \min_{\Psi, \{x_i\}_{i=1}^T} \sum_{i=1}^T \|\Phi(y_i) - \Phi(\Psi)W_i^{-1}x_i\|_2^2 + \lambda \sum_{i=1}^T ||x_i||_1,$n

where

$$W_i(k, k) = ||\Phi(y_i) - \Phi(\psi_k)||_2^2$$

$$= K(y_i, y_i) + K(\psi_k, \psi_k) - 2K(y_i, \psi_k)$$

$$= 2(1 - K(y_i, \psi_k)).$$

For each data sample $y_i$, kernel local sparse coding can be performed by simplifying the objective function in (20) as

$$x_i^T W_i^{-1} K(\Psi, \Psi) W_i^{-1} x_i - 2K(y_i, \Psi) W_i^{-1} x_i + \lambda ||x_i||_1$$

$$= L(x_i) + \lambda ||x_i||_1.$$

This can be solved using the feature-sign search algorithm, with the additional complexity of evaluating the kernel matrices.

4.1. Dictionary Learning

In order to perform dictionary update using the kernel matrices, the problem can be written as

$$\min_{\Psi} \sum_{i=1}^T \|\Phi(y_i) - \Phi(\Psi)\hat{z}_i\|_2^2 \quad \text{s.t.} \quad ||\psi_k|| \leq 1, \forall k.$$
The objective can be equivalently expressed as

\[ f = \sum_{i=1}^{T} \left[ 1 + \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \hat{z}_i(k_1)\hat{z}_i(k_2)K(\psi_{k_1}, \psi_{k_2}) - 2 \sum_{k_1=1}^{K} \hat{z}_i(k_1)K(\psi_{k_1}, y_i) \right]. \]  

(24)

Since the objective function to be minimized involves the kernel matrices, we choose to iteratively update one dictionary atom at a time. In order to update the \( k \)th dictionary atom, we differentiate the objective in (23) as

\[ \frac{\partial f}{\partial \psi_k} = -4\gamma \sum_{i=1}^{T} \left[ \sum_{k_2=1}^{K} \hat{z}_i(k)\hat{z}_i(k_2)K(\psi_k, \psi_{k_2})(\psi_k - \psi_{k_2}) - \hat{z}_i(k)K(\psi_k, y_i)(\psi_k - y_i) \right]. \]  

(25)

By setting the derivative to zero, we can estimate the modified dictionary element. However, the kernel matrices \( K(\psi_k, \Psi) \) and \( K(\psi_k, Y) \) are unknown and hence we resort to using \( \psi_k \) from the earlier iteration of the algorithm to compute those kernel matrices. In other words, the dictionary atom update in the \( n \)th iteration can be performed as

\[ \sum_{i=1}^{T} \left[ \sum_{k_2=1}^{K} \hat{z}_i(k)\hat{z}_i(k_2)K(\psi_k^{n-1}, \psi_{k_2})(\psi_k^n - \psi_{k_2}) - \hat{z}_i(k)K(\psi_k^{n-1}, y_i)(\psi_k^n - y_i) \right] = 0. \]  

(26)

Solving this we can obtain,

\[ \psi_{k,n} = \frac{\Psi diag[K(\psi_{k,n-1}, \Psi)]\hat{Z}\hat{z}_{k,r}^T - Y diag[K(\psi_{k,n-1}, Y)]\hat{z}_{k,r}^T}{K(\psi_{k,n-1}, \Psi)\hat{Z}\hat{z}_{k,r}^T - K(\psi_{k,n-1}, Y)\hat{z}_{k,r}^T} . \]  

(27)

Here \( \hat{z}_{k,r} \) is a row vector containing the set of coefficients of all training vectors corresponding to the dictionary atom \( \psi_k \).

5. Object Recognition

In this section, we evaluate the performance of the proposed local sparse coding algorithms in object recognition. Following the ScSPM approach
described in (Yang et al., 2009), we compute sparse codes for the local descriptors in an image and aggregate them at multiple spatial scales. In our simulations, all images are divided into patches of size 24 × 24 with a grid spacing of 8 and one SIFT descriptor is extracted per patch. The descriptors are coded using the following state-of-the-art methods: (a) regular sparse coding using K-means dictionary (SC) (Yang et al., 2009), (b) kernel sparse coding (KSC) (Gao et al., 2010a), (c) Laplacian sparse coding (Laplacian SC) (Gao et al., 2010b), (d) locality-constrained linear coding using K-means dictionary (LLC) (Wang et al., 2010), (e) proposed local sparse coding algorithm (LSC), and (f) proposed kernel local sparse coding algorithm (KLSC). We then employ spatial pyramid coding, where we generate region-specific codes at different spatial scales. Each image is processed at spatial scales 1, 2 and 4 respectively. In the first scale, where the full image is considered, the sparse codes are stacked into a matrix and max pooling (Wang et al., 2010) is performed to identify the maximum coefficient value at each index. At scale 2, the image is split into 4 regions and 4 max pooled feature vectors are generated. Similarly, 16 feature vectors are obtained at scale 4. Finally, all the 21 feature vectors are stacked into a single column vector and used as the training feature to a linear SVM. We evaluated the object recognition performance using Caltech-256, UIUC sports, Corel-10 and Scene-15 datasets. We fixed the number of dictionary elements \( K = 1024 \) in all cases and randomly chose 75,000 patches to generate the dictionary for each dataset. We set the parameter \( \lambda = 0.3 \) for the feature-sign search algorithm. For the parameter \( \gamma \) in the Gaussian kernel function, we selected the values \( \frac{1}{64}, \frac{1}{128}, \frac{1}{128}, \frac{1}{256} \) on Scene-15, UIUC sports, Caltech-256 and Corel-10 respectively. It is evident from
Table 1: Comparison of the classification accuracies on the Caltech-256 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th># Training Vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15</td>
</tr>
<tr>
<td>SC (Yang et al., 2009)</td>
<td>27.73</td>
</tr>
<tr>
<td>KSC (Gao et al., 2010a)</td>
<td>29.77</td>
</tr>
<tr>
<td>Laplacian SC (Gao et al., 2010b)</td>
<td>30.01</td>
</tr>
<tr>
<td>LLC (Wang et al., 2010)</td>
<td>30.08</td>
</tr>
<tr>
<td>LSC (Proposed)</td>
<td><strong>30.19</strong></td>
</tr>
<tr>
<td>K-LSC (Proposed)</td>
<td><strong>30.28</strong></td>
</tr>
</tbody>
</table>

all the results presented in this section that the proposed algorithms achieve improved classification accuracies in comparison to the other approaches. All results presented were averaged over 10 iterations and the training/testing sets were randomly chosen in each iteration.

5.1. Caltech-256

The Caltech-256 dataset (Griffin et al., 2007) contains 30,607 images in 256 categories and its variability makes it extremely challenging in comparison to the Caltech-101 dataset. The intra-class variance is quite high and the objects of interest are not necessarily in the center of the images. We tested the performance of the different algorithms with 30, 45, and 60 training images per class respectively. Table 1 shows the classification accuracies obtained using the different algorithms. As it can be observed, kernel local sparse coding algorithm outperformed the other sparse coding based approaches in all cases.
Table 2: Comparison of the classification accuracies on the UIUC sports dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC (Yang et al., 2009)</td>
<td>82.74</td>
</tr>
<tr>
<td>KSC (Gao et al., 2010a)</td>
<td>84.92</td>
</tr>
<tr>
<td>Laplacian SC (Gao et al., 2010b)</td>
<td>85.31</td>
</tr>
<tr>
<td>LLC (Wang et al., 2010)</td>
<td>85.34</td>
</tr>
<tr>
<td><strong>LSC (Proposed)</strong></td>
<td><strong>85.56</strong></td>
</tr>
<tr>
<td><strong>K-LSC (Proposed)</strong></td>
<td><strong>85.93</strong></td>
</tr>
</tbody>
</table>

5.2. UIUC Sports Dataset

The UIUC sports dataset (Li and Fei-Fei, 2007) contains 1792 images from 8 different classes: badminton, bocce, croquet, polo, rock climbing, rowing, sailing and snow boarding. The number of images in each class varies from 137 to 250. Following the standard evaluation procedure (Gao et al., 2010a), we randomly select 70 images from each class for training and the rest for testing. The object recognition performance is reported in Table 2.

5.3. Corel-10 Dataset

The Corel-10 dataset (Lu and Ip, 2009) contains 1000 images belonging to 10 categories: skiing, beach, buildings, tigers, owls, elephants, flowers, horses, mountains and food. Each class contains 100 images and we randomly select 50 images from each class for training. The results obtained are presented in Table 3.
Table 3: Comparison of the classification accuracies on the Corel-10 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC (Yang et al., 2009)</td>
<td>86.2</td>
</tr>
<tr>
<td>KSC (Gao et al., 2010a)</td>
<td>89.43</td>
</tr>
<tr>
<td>Laplacian SC (Gao et al., 2010b)</td>
<td>88.4</td>
</tr>
<tr>
<td>LLC (Wang et al., 2010)</td>
<td>88.41</td>
</tr>
<tr>
<td>LSC (Proposed)</td>
<td>88.68</td>
</tr>
<tr>
<td>K-LSC (Proposed)</td>
<td>89.71</td>
</tr>
</tbody>
</table>

5.4. Scene-15 Dataset

The Scene-15 dataset (Lazebnik et al., 2006) contains 4485 images belonging to 15 different categories and is typically used for evaluating scene classification performance. The number of images per class vary between 200 and 400 images. The scene categories of this dataset include suburb, coast, forest, highway, inside city, mountain, open country, street, tall building, office, bedroom, industrial, kitchen, living room and store. We train the linear SVM by using 100 images from each class for training and the rest for testing.

6. Image Retrieval

In this section, we present a supervised coding approach for performing image retrieval using sub-image heterogeneous features. Recent methods for complex visual recognition tasks typically extract multiple features that describe different aspects of the underlying visual characteristics of an image. The features can either be global or local, and combining these features has resulted in improved visual recognition (Frahm et al., 2010; Cao et al., 2009).
Table 4: Comparison of the classification accuracies on the Scene-15 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC (Yang et al., 2009)</td>
<td>80.28</td>
</tr>
<tr>
<td>KSC (Gao et al., 2010a)</td>
<td>83.08</td>
</tr>
<tr>
<td>Laplacian SC (Gao et al., 2010b)</td>
<td>89.75</td>
</tr>
<tr>
<td>LLC (Wang et al., 2010)</td>
<td>89.78</td>
</tr>
<tr>
<td><strong>LSC (Proposed)</strong></td>
<td><strong>89.94</strong></td>
</tr>
<tr>
<td><strong>K-LSC (Proposed)</strong></td>
<td><strong>90.61</strong></td>
</tr>
</tbody>
</table>

We employ the approach in (Cao et al., 2009), where the base feature is obtained by stacking multiple features, with each feature normalized to unit $\ell_2$ norm separately.

Following the feature extraction method in (Thiagarajan et al., 2012) we first divide an image into highly overlapping sub-images, that are much larger than image patches used for extracting local descriptors. For example, in an image of size $256 \times 256$, the sub-images can be of size $128 \times 128$. Since the sub-images are large, they correspond to significant portions of objects/scenes in the image. We extract the base (heterogeneous) feature from each sub-image of all the training images and learn a dictionary for the features. This enables us to identify the key representative sub-image features found across all images. This dictionary corresponds to a “bag of visual phrases” where each dictionary element corresponds to a visual phrase. This can be contrasted with the “bag of visual words”, which is the dictionary that we would obtain if we had considered small image patches. Note that, we directly learn the visual phrases using sub-image features rather than
constructing phrases using the visual words. In (Thiagarajan et al., 2012), each sub-image feature is sparse coded using the dictionary and the codes of all sub-images in an image are aggregated in a spatial pyramid (Lazebnik et al., 2006). Aggregation is performed using max-pooling, and the max-pooled codes from all regions are stacked together and normalized to unit $\ell_2$ norm. This aggregated feature vector represents the importance of each visual phrase in the image.

In Figure 3, we show one sample image each from three classes of the Microsoft Research Cambridge database (Ulusoy and Bishop, 2005). The feature computed by aggregating the sub-image features at one spatial scale are shown on the right. In the first image, there is predominantly only one object present in the image (bicycle). The second image (chimney) contains geometric structures (walls, rooftops) in addition to the background (sky).
In the third image, all sub-images are quite similar to cloud-like patterns. The feature vectors for the images shown have disjoint non-zero supports and this leads to high discrimination among classes.

6.1. Supervised Local Sparse Coding

The sub-image features can be coded using the local sparse coding or the kernel local sparse coding algorithm described in Sections 3 and 4. However, the prior knowledge of the labels/tags associated with the training images are not considered when learning the dictionary. Since small image patches were used to train the dictionaries for classification, we could not assign labels to them. Whereas in the case of image retrieval, assigning labels to sub-images will enable us to perform supervised local sparse coding during training.

We propose to employ a coding scheme that exploits the fact that once a dictionary atom has been chosen to represent a sub-image feature in a class, it may as well be used to represent other sub-image features of the same class. In addition to exploiting the class label information, this will ensure that aggregating the sparse codes from all sub-images will result in a sparse feature. In (Bengio et al., 2009), the authors proposed a mixed-norm regularization method to perform group sparse coding of visual descriptors in an image. Furthermore, simultaneous sparse coding has been successfully employed to obtain sparse codes for a set of data samples.

In our setup, we assume that all sub-image features in a class can be modeled using the same neighborhood. In other words, we want to exploit the clusterability of the features within a class. For each class, the local neighborhood is identified by computing a representative feature for that class and then determining the weights corresponding to all dictionary atoms.
If we employ the distance metric in (5), mean of all features is used as the representative. Whereas, the rank-1 SVD of the features is used as the representative when (6) is employed. Given the neighborhood, we perform supervised coding of the sub-image features in a class as

$$\hat{X}^{(g)} = \min_X \| Y^{(g)} - \Psi W_{(g)}^{-1} X \|_F^2 \quad \text{s.t.} \quad \| X \|_{row-0} \leq L,$$  

(28)

where $Y^{(g)}$ is the set of training sub-image features in class $g$, $W_{(g)}$ is the estimated neighborhood for that class and $\| X \|_{row-0}$ is the row-$\ell_0$ pseudo norm of the coefficient matrix i.e., $\| X \|_{row-0} = \text{rowsupp}(X)$. Here,

$$\text{rowsupp}(X) = \{ k \in [1...K] : x_{k,t} \neq 0 \text{ for some } t \}.$$  

(29)

Several algorithms exist to solve this simultaneous sparse approximation problem (Rakotomamonjy, 2011) and any algorithm can be chosen depending on the accuracy and computational complexity requirements of the system.

The dictionary update step in Section 3.1 can be extended to the case of supervised local sparse coding easily. The objective function to be minimized when updating the dictionary atom $\psi_k$ can be written as

$$\| Y^{(1)} - \Psi \hat{Z}^{(1)} \|_F^2 + \ldots + \| Y^{(G)} - \Psi \hat{Z}^{(G)} \|_F^2,$$  

(30)

where $G$ denotes the total number of classes. Substituting $\hat{Z}^{(g)} = W_{(g)}^{-1} \hat{X}^{(g)}$, the objective can be simplified as

$$\| Y^{(1)} - \Psi \hat{Z}^{(1)} \|_F^2 + \ldots + \| Y^{(G)} - \Psi \hat{Z}^{(G)} \|_F^2,$$  

(31)

$$= \| E^{(1)}_k - \psi_k \hat{z}_{k,r}^{(1)} \|_F^2 + \ldots + \| E^{(G)}_k - \psi_k \hat{z}_{k,r}^{(G)} \|_F^2,$$

$$= \left\| \begin{bmatrix} E^{(1)}_k & \ldots & E^{(G)}_k \end{bmatrix} - \psi_k \begin{bmatrix} \hat{z}_{k,r}^{(1)} & \ldots & \hat{z}_{k,r}^{(G)} \end{bmatrix} \right\|_F^2,$$

$$= \| \tilde{E}_k - \psi_k \tilde{z}_k \|_F^2.$$  

(32)
In effect, the updated dictionary atom can be obtained using the procedure in (17) with the concatenated error matrix $\tilde{E}_k$ and the coefficient vector $\tilde{z}_k$ (corresponding to $\psi_k$).

6.2. Microsoft Cambridge Image Dataset

In our simulations, we obtained images from the Microsoft Research Cambridge image database (Ulusoy and Bishop, 2005) and resized them to $256 \times 256 \times 3$. The dataset consists of 4,323 images belonging to 19 different classes. We used 75% of images randomly chosen from each class for training and the rest for testing. For each image, we extracted two global features: Histogram of Oriented Gradients (HoG), GIST and one local feature: Local Binary Pattern (LBP). We generated the heterogeneous base feature for each image by $\ell_2$ normalizing and stacking the three features. To generate the sub-image features, we divided each image into overlapping patches of size $128 \times 128$ and obtained the base feature for each patch. We employed the algorithm in Section 6.1 to learn a dictionary containing 512 atoms using sub-image features from the training set. The supervised local sparse codes from each image were aggregated in 2 spatial scales. In the first spatial level, all sub-image features were pooled together to form a $512 \times 1$ vector. In spatial level 2, each image was divided into 4 regions and sparse codes of only sub-images from that region were aggregated. Aggregated codes from both levels were stacked to form a $2560 \times 1$ vector and $\ell_2$ normalized. For each test image, all sub-image features were simultaneously coded using (28) and spatial pyramid features were constructed.

For comparison we generated spatial pyramid features by aggregating local sparse codes obtained by not incorporating the class label information.
Figure 4: Precision vs Recall curves for different classes from the Microsoft Research Cambridge image database (Ulusoy and Bishop, 2005). In each case, a sample set of test images are shown. The images are best viewed in color and 300% zoom-in.
We employed a simple Euclidean distance based nearest neighbor search on the features to identify visually similar images. Figure 4 demonstrates the Precision vs Recall curves, obtained using the original heterogeneous features, proposed local sparse coded sub-image features with/without label information for 5 different classes from the dataset. Note that the curves are obtained by averaging the retrieval results of all test images in every class. As it can be observed, in all cases both local sparse coding based features performed significantly better than the heterogeneous features. Furthermore, the performance improvement by employing supervised coding is also evident from the results.

7. Conclusions

In this paper, we proposed dictionary learning algorithms to perform local sparse coding of image features for object recognition and image retrieval. The proposed approaches can be interpreted as a generalization of clustering where a training vector is assigned to more than one dictionary atom in its neighborhood. Since the kernel trick can be used to exploit the non-linear similarity of the features, we proposed to perform local sparse coding in a high-dimensional feature space obtained by an implicit mapping function. A dictionary learning procedure for the case of the RBF kernel was also described. When applied for object recognition, the proposed algorithms achieved improved recognition performance in comparison to other sparse coding based classification approaches. Incorporating supervised (label/tag) information when coding sub-image heterogeneous features for image retrieval, resulted in highly discriminative local sparse codes. The
The proposed dictionary learning algorithm was adapted to the case where the features in a class were coded simultaneously using a common neighborhood. Using simulations, we demonstrated the gain in image retrieval performance obtained by performing supervised coding.

References


Gao, S., Tsang, I., Chia, L., Zhao, P., 2010b. Local features are not lonely laplacian sparse coding for image classification. In: IEEE CVPR.


