LEARNING DICTIONARIES FOR LOCAL SPARSE CODING IN IMAGE CLASSIFICATION

Jayaraman J. Thiagarajan and Andreas Spanias

SenSIP Center, School of ECEE, Arizona State University, Tempe, AZ 85287-5706, USA.
Email: {jjayaram, spanias}@asu.edu

ABSTRACT

Low dimensional embedding of data samples lying on a manifold can be performed using locally linear modeling. By incorporating suitable locality constraints, sparse coding can be adapted to modeling local regions of a manifold. This has been coupled with the spatial pyramid matching algorithm to achieve state-of-the-art performance in object recognition. In this paper, we propose an algorithm to learn dictionaries for computing local sparse codes of descriptors extracted from image patches. The algorithm iterates between a local sparse coding step and an update step that searches for a better dictionary. Evaluation of the local sparse code for a data sample is simplified by first estimating its neighbors using the proposed distance metric and then computing the minimum $\ell_1$ solution using only the neighbors. The proposed dictionary update ensures that the neighborhood of a training sample is not changed from one iteration to the next. Simulation results demonstrate that the sparse codes computed using the proposed dictionary achieve improved classification accuracies when compared to using a $K$-means dictionary with standard image datasets.

Index Terms— Local sparse codes, sparse representations, dictionary learning, linear classifiers.

1. INTRODUCTION

Sparse coding allows for an efficient signal representation using a linear combination of elementary signals. 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

$$y = \Psi x + n,$$

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 non-zero. Relevant cost functions that can promote sparsity such as the $\ell_0$ and the $\ell_1$ norms are commonly used. The $\ell_1$ minimization is given by

$$\min_x ||x||_1 \text{ subject to } ||y - \Psi x||_2 \leq \epsilon$$

where, $||.||_1$ refers to the $\ell_1$ norm and $\epsilon$ is the error goal. Algorithms used to solve (2) include the Basis Pursuit [1] and the least angle regression algorithm [2] with the lasso modification (LARS-LASSO). The dictionary $\Psi$ when adapted to the data has been shown to provide superior performance when compared to predefined dictionaries in several applications [3, 4]. The joint optimization problem of dictionary learning and sparse coding can be expressed as

$$\min_{\Psi, x} \|Y - \Psi X\|_F^2 \text{ s.t. } ||x_i||_0 \leq s, \forall i, ||\psi_j||_2 = 1, \forall j,$$

where $Y$ is the collection of $T$ training vectors, $X$ is the coefficient matrix, $s$ is the sparsity of the coefficient vector and $||.||_F$ denotes the Frobenius norm. The K-SVD algorithm, which is a generalization of the K-hyperline clustering procedure [5], iteratively minimizes the objective in (3) [6].

1.1. Object Recognition

The state-of-the-art Spatial Pyramid Matching (SPM) algorithm for object recognition proposed in [7] partitions the image into increasingly finer regions and evaluates features in the local regions. The features are coded using a dictionary and the code vectors in each region are then pooled together by building histograms. Finally, the histograms of the different regions are concatenated and presented to a non-linear classifier. Figure 1 illustrates the steps involved in the SPM algorithm. The features extracted from the images are either Scale Invariant Feature Transform (SIFT) descriptors or Histogram of Oriented Gradients (HOG) descriptors. Typically, a K-means dictionary is learned from the descriptors and the descriptors are coded using vector quantization.

Though, this approach has been very effective, the complexity of using a non-linear classifier is quite high. Hence, the authors in [8] proposed to replace the vector quantization in SPM by sparse coding, which enabled the use of linear classifiers. Furthermore, an approximate Locality constrained Linear Coding (LLC) algorithm has been developed [9], which allows a fast implementation of Local Coordinate Coding (LCC) based image classification.
In this paper, an algorithm to learn dictionaries for local sparse coding of the image descriptors is presented. We propose a distance metric for computing the weights in local sparse coding that is well suited when normalized dictionaries are used. We solve the joint optimization of local sparse coding and dictionary learning, with the constraint that the dictionary update should not alter the neighborhood relationship between a dictionary atom and the training vectors it represents. In the proposed algorithm, a dictionary atom is updated as a weighted and normalized sum of the training vectors it represents. The coefficients are computed by identifying the dictionary atoms in the neighborhood using the proposed distance metric and performing a sparse coding using the chosen atoms. In contrast to computing the coefficients as a least squares solution in the approximate LLC setup, we employ the LARS-LASSO algorithm to perform sparse coding. Simulation results for object recognition using features demonstrate that the proposed dictionary achieves improved classification performance with Caltech-101, Caltech-256 and Flickr image datasets in comparison to using K-means dictionaries.

2. EXISTING LOCAL CODING SCHEMES

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 [10]. 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, tends 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,

$$\min_x \sum_{k=1}^{K} w(k)|x_k| \quad \text{subject to } \|y - \Psi x\|_2 \leq \epsilon, \quad (4)$$

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 [11] computes local sparse codes using weights based on the Euclidean distance measure as given by

$$w(k) = \|y - \psi_k\|_2^2, \quad (5)$$

where $\psi_k$ is the $k$th element of $\Psi$.

2.1. Approximate LLC Framework

The approximate method for locality constrained linear coding proposed in [9], uses an approach similar to the LLE. In general, the LLC algorithm employs the following criteria:

$$\min_x \sum_{i=1}^{T} \|y_i - \Psi x_i\|_2^2 + \lambda \|w_i \odot x_i\|_2^2 \text{ s.t. } 1^T x_i = 1, \forall i, \quad (6)$$

where $\odot$ denotes element-wise multiplication and $w_i$ measures the similarity between the data sample and all the dictionary atoms. The distance metric used is

$$w_i(k) = \exp \left( \frac{\|y_i - \psi_k\|_2}{\sigma} \right), \forall k, \quad (7)$$

where $\sigma$ is used to adjust the control the magnitude of weights 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 $O(K^2)$ to $O(K + P^2)$, where $K$ denotes the number of dictionary atoms and $P << K$.

### 3. Proposed Dictionary Learning Algorithm for Local Sparse Coding

Typically, dictionaries in linear spatial pyramid matching are learned using the K-means algorithm and the Euclidean distance is used to identify the neighbors. In our setup, we propose to employ the following distance metric:

$$ w(k) = \|y - (y^T \psi_k) \psi_k\|^2. \quad (8) $$

It is important to note that this metric is used by K-hyperline clustering [5] to identify the cluster membership of a data sample. 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 (8) in comparison to (5). Hence, the proposed metric produces better discrimination when compared to using an Euclidean distance when the dictionary elements are normalized.

The proposed dictionary learning algorithm iterates between local sparse coding and dictionary update steps. In the dictionary update step, we ensure that the neighborhood relation between a dictionary atom and the training vectors it represents is not modified over iterations. The dictionary learning is similar to a generalized clustering procedure where the training vectors are assigned to more than one cluster center (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 initial dictionary is obtained using the K-means clustering of the training vectors.

#### 3.1. Local Sparse Coding Step

In the proposed framework, we employ the LARS-LASSO algorithm to obtain a local sparse code. To solve our weighted $\ell_1$ problem, we rewrite (4) as

$$ \min_x \sum_{i=1}^{K} |x_k| \quad \text{subject to} \quad \|y - \Psi W^{-1}x\|_2 \leq \epsilon, \quad (9) $$

where $W$ is a diagonal matrix created from the elements of the vector $w$. We provide the weighted dictionary, $\Psi W^{-1}$, and the desired number of non-zero entries in the coefficient vector to the LARS-LASSO algorithm. The resulting sparse code is reweighted as $Wx$, so that the updated coefficient vector can be used with the dictionary $\Psi$ to reconstruct $y$.

Though LARS-LASSO is significantly faster than using convex optimization, solving (9) is still computationally expensive. Hence, we propose a greedy scheme where we first identify the $P$ nearest neighbors of $y$ using (8) and then compute the corresponding $P$-sparse code as

$$ \min_x \left\| y - \sum_{i \in C} \psi_i x_i \right\| \quad \text{subject to} \quad \|x\|_0 \leq P, \quad (10) $$

where $C$ contains the set of indices of the $P$ nearest dictionary atoms selected by the distance metric in (8). This optimization implies that only the selected dictionary atoms, $\{\psi_i\}_{i \in C}$, can participate in the approximation of $y$.

#### 3.2. Dictionary Update Step

An important consideration in the proposed algorithm is that updating the dictionary atom should not affect its neighborhood relation with the training vectors it represents when trying to reduce the error. This consideration also ensures that the relation between a training sample and all the $P$ nearest dictionary atoms does not change from one iteration to the next. We can achieve this by fixing the coefficients when the dictionary atoms are updated. 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. It is important to note that the K-SVD algorithm is optimized to reduce the representation error. However, our goal is to preserve the neighborhood and hence we update the dictionary using the training vectors directly and not the residual vectors. For each dictionary atom $\psi_k$, we solve

$$ \psi_k = \min_{\psi} \|Y_R - \psi x_R^k\|_F \quad \text{s.t.} \quad \|\psi\|_2 = 1, \quad (11) $$

where $Y_R$ denotes the set of training vector with non-zero coefficients for the dictionary atom $\psi_k$ and $x_R^k$ contains the corresponding coefficients. The objective can be rewritten as

$$ \|Y_R - \psi x_R^k\|_F = \text{tr}[(Y_R - \psi x_R^k)^T(Y_R - \psi x_R^k)], $$

$$ = \text{tr}[Y_R^T Y_R - Y_R^T \psi x_R^k - (x_R^k)^T \psi^T Y_R + (x_R^k)^T \psi^T \psi x_R^k]. $$

Since $\psi^T \psi = 1$, the optimization with respect to $\psi$ will consider only the second and third terms. Since $tr[A + A^T] = 2tr[A]$ and $tr[AB] = tr[BA]$, the objective to be minimized can be simplified as $tr(-Y_R^T \psi x_R^k) = tr(-x_R^k Y_R^T \psi)$. Hence, the dictionary atom $\psi_k$ is updated as

$$\psi_k = \frac{Y_R(x_R^k)^T}{||Y_R(x_R^k)^T||_2}.$$  \hfill (12)

It can be observed that this dictionary update is equivalent to performing a weighted K-means procedure, 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 K-means cluster centers over iterations. We will demonstrate in our experiments that, dictionaries learned by replacing the proposed dictionary update by a K-SVD like update will result in reduced classification performance when compared to dictionaries learned using the proposed method.

4. SIMULATIONS

In this section, we report the performance of the proposed dictionary in object recognition. In our simulations, all images are divided into patches of size $24 \times 24$ with a grid spacing of 8 and one SIFT descriptor is extracted per patch. We then employ spatial pyramid coding, where we generate region-specific codes at different spatial scales. Local sparse codes are generated using the approach described in Section 3.1. The 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 [9] 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. It will be evident from all the results presented in this section that the proposed dictionary achieves improved classification accuracy in comparison to using a standard K-means dictionary.

### 4.1. Images from the Flickr Database

In this experiment, the labeled training data and the test data were obtained from the Flickr online database [12]. Totally 5 image classes were considered (elephant, coffee mug, building, bird and outdoor) and 30 images of varying sizes were chosen for each class. A set of 25000 training vectors were randomly selected from the SIFT feature set of all training images and dictionaries of sizes $128 \times 256$ and $128 \times 512$ were designed using K-means clustering and the proposed algorithm. The features were presented to a linear SVM and the results are reported in Table 1. We tested the classification performance by using 5, 10 and 15 images per class for training and the rest for testing. The performance reported are obtained by averaging results from 10 fold cross validation. For comparison, a dictionary learned by iterating between local sparse coding in Section 3.1 and the K-SVD dictionary update step [6] is also used.

### 4.2. Caltech-101 Dataset

We present the results obtained by performing object recognition in images obtained from the Caltech-101 [13] dataset using the proposed dictionary. The dataset contains a total of 9144 images from 101 classes of objects and an additional class of background images. We used a setup similar to the one used in Section 4.1 to generate features at different spatial resolutions. We tested the performance by using different number of images per category for training and the rest for testing. Dictionaries containing 1024 elements were learned from SIFT descriptors of 50000 randomly chosen patches. The experiment was randomly repeated for 10 times and the average performance is reported in Table 2.

### 4.3. Caltech-256 Dataset

The Caltech-256 dataset [14] contains 30,607 images in 256 categories and its variability makes it extremely challenging in comparison to the Caltech-101 dataset. A setup similar to the one presented in the previous section was used and we tested the performance of the proposed dictionary with 30, 45, and 60 training images per class respectively. We used dictionaries with 2048 atoms. The experiment was randomly

---

**Table 1.** Classification performance on the images obtained from the Flickr database using the proposed dictionaries.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>5</td>
<td>66.32</td>
<td>63.49</td>
<td>66.90</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>73.43</td>
<td>69.74</td>
<td>74.10</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>76.20</td>
<td>74.22</td>
<td>77.30</td>
</tr>
<tr>
<td>512</td>
<td>5</td>
<td>66.90</td>
<td>63.93</td>
<td>68.16</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>73.86</td>
<td>71.49</td>
<td>75.31</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>76.90</td>
<td>76.09</td>
<td>78.74</td>
</tr>
</tbody>
</table>
Table 2. Classification accuracy for the Caltech-101 dataset.

<table>
<thead>
<tr>
<th># Training Vectors</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dictionary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-means</td>
<td>48.43</td>
<td>58.58</td>
<td>63.69</td>
<td>66.9</td>
<td>69.38</td>
<td>71.32</td>
</tr>
<tr>
<td>Proposed</td>
<td>49.57</td>
<td>59.64</td>
<td>64.73</td>
<td>67.96</td>
<td>70.28</td>
<td>72.4</td>
</tr>
</tbody>
</table>

repeated for 10 times and the average performance is reported in Table 3.

5. CONCLUSIONS

In this paper, we presented an iterative procedure to learn dictionaries for local sparse coding in linear spatial pyramid matching. The procedure is similar to generalized clustering where a training vector is assigned to more than one neighboring dictionary atom. The neighborhood is identified using the distance metric employed by K-hyperline clustering and the local sparse code is evaluated by solving an $\ell_1$ minimization using only the neighbors. The dictionary update step ensures that the neighborhood relation between the training vectors and dictionary atoms do not change significantly over iterations. Simulation results for object recognition demonstrate that the proposed dictionary provides improved classification when compared to using K-means dictionaries.

Table 3. Classification accuracy for the Caltech-256 dataset.

<table>
<thead>
<tr>
<th># Training Vectors</th>
<th>30</th>
<th>45</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dictionary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-means</td>
<td>33.19</td>
<td>37.11</td>
<td>40.02</td>
</tr>
<tr>
<td>Proposed</td>
<td>34.07</td>
<td>37.92</td>
<td>40.89</td>
</tr>
</tbody>
</table>

6. REFERENCES