Positive Definite Dictionary Learning for Region Covariances

Ravishankar Sivalingam Daniel Boley Vassilios Morellas Nikolaos Papanikolopoulos
Department of Computer Science & Engineering, University of Minnesota
Minneapolis, MN, USA
{ravi,boley,morellas,npapas}@cs.umn.edu

Abstract

Sparse models have proven to be extremely successful in image processing and computer vision, and most efforts have been focused on sparse representation of vectors. The success of sparse modeling and the popularity of region covariances have inspired the development of sparse coding approaches for positive definite matrices. While in earlier work [1], the dictionary was pre-determined, it is clearly advantageous to learn a concise dictionary adaptively from the data at hand. In this paper, we propose a novel approach for dictionary learning over positive definite matrices. The dictionary is learned by alternating minimization between the sparse coding and dictionary update stages, and two different atom update methods are described. The online versions of the dictionary update techniques are also outlined. Experimental results demonstrate that the proposed learning methods yield better dictionaries for positive definite sparse coding. The learned dictionaries are applied to texture and face data, leading to improved classification accuracy and strong detection performance, respectively.

1. Introduction

Sparse models have proven to be extremely successful in image processing, computer vision, and machine learning. However, a majority of the effort has been focused on the sparse representation of vectors and low-rank models for general matrices. The success of sparse modeling, along with the growing popularity of region covariances for many vision problems, has inspired the development of sparse coding approaches for these positive definite descriptors, either by vectorizing them [2] or keeping them in their original form [1]. While the dictionary was previously formed from some or all of the training set, it is advantageous to learn a concise dictionary representation from the available data.

In this paper, we present a novel approach for dictionary learning over positive definite matrices, while maintaining their structure, *i.e.*, without vectorization. The method was inspired by the K-SVD algorithm of Aharon *et al*. [3]. The dictionary is learned by alternating minimization, oscillating between the sparse coding and dictionary update stages. In the dictionary update stage, we update one atom at a time, sequentially, while retaining the sparsity structure of the coefficients. The corresponding sparse coefficients are also updated. We present two approaches to the atom update step - one based on gradient descent and another alternative method where the atom update has a closed-form solution.

Experimental results demonstrate that the dictionary learning approach reduces the reconstruction error for positive definite sparse coding. In classification applications using region covariances, where the class-wise residual error is used as a feature, the dictionary learning improves the classification accuracy. In a classification setting, where the class-wise residual error is used as a decision variable for label assignment, the dictionary learning yields a substantial gain in accuracy over dictionaries formed by sampling the training set. In an object detection framework, a single concise dictionary learned from training data demonstrates very strong detection capabilities.

The rest of this paper is organized as follows: In Section 1.1, we give a brief overview of related work on region covariances, and discuss other vector dictionary learning techniques. Section 1.2 covers some basic notation used throughout the paper. An overview of the positive definite sparse coding formulation from [1] is presented in Section 1.3. In Section 2, we propose our dictionary learning formulation, with a gradient descent-based approach in Section 2.1 and an alternative approach in Section 2.2, which is solvable in closed-form. Online versions of the two update methods are presented in Section 2.4. Experiments, both synthetic and real, are presented in Section 3, and Section 4 wraps up with the conclusions, outlining the future work.

1.1. Related Work

Region covariances were introduced by Tuzel *et al*. [4] as a novel region descriptor for object detection and texture classification, along with the ability for fast construction of covariances over arbitrary-sized windows in constant time, using integral images [5]. Region covariances belong

to the space of positive definite matrices $\mathbb{S}_n^{++}$, forming a connected Riemannian manifold. These descriptors have been used for texture segmentation and classification [4, 6], detection of pedestrians [7], and other objects [8]. Region covariances have also been used extensively for object tracking [9, 10] and for image retrieval and recognition in a surveillance setting [11, 12]. [13, 14] use Gabor-based region covariances for face recognition, and in [2, 15] Guo *et al*. use derivatives of optical flow for action recognition.

While covariance descriptors have risen in popularity, the methods used in most applications remain confined to k-nearest-neighbors or kernel SVMs, using the geodesic distance measure [16]. Very recently, there have been new attempts on sparse coding for region covariances. In Guo *et al*. [2], the positive definite matrices are taken to the tangent space of the Riemannian manifold, by taking the matrix logarithm. This is the space of symmetric matrices $\mathbb{S}_n$, where the data points can be easily vectorized and sparse-coded. Note that this is not truly a sparse linear representation of the positive definite matrices, but a non-linear one due to the matrix log operation. In Sivalingam *et al*. [1], the positive definite nature of the matrices is preserved, and a completely different sparse coding objective is formulated to deal with the data points in $\mathbb{S}_n^{++}$ itself.

In the sparse representation of vectors, the most popular dictionary learning algorithm is the K-SVD algorithm of Aharon *et al*. [3]. The positive definite dictionary learning approach presented here is inspired by the K-SVD algorithm, and shares certain similarities in the approach. In Mairal *et al*. [17], the authors extended the dictionary learning formulation to incorporate a discriminative term between dictionaries of different classes. Ramirez *et al*. [18] modify the learning framework based on the incoherence among dictionary atoms, presenting clustering and classification scenarios. A fast online dictionary learning technique was introduced in Mairal *et al*. [19]. Dictionary learning under more structured circumstances are presented in [20, 21]. To the best of our knowledge, this is the first work in dictionary learning over positive definite matrices.

1.2. Notation

In this section we define the notation to be used throughout the paper. $\mathbb{S}_n^{+}$ denotes the class of $n \times n$ symmetric positive semidefinite matrices, while $\mathbb{S}_n^{++}$ refers to strictly positive definite matrices. $A \succ 0$ ($A \succeq 0$) denotes A is positive (semi)definite. $A \succ B$ ($A \succeq B$) indicate that $(A - B)$ is positive (semi)definite. $\mathcal{A} = \{A_i\}_{i=1}^{K}$ is the dictionary and $\mathcal{S} = \{S_j\}_{j=1}^{N}$ is the data set, with $A_i \in \mathbb{S}_n^{++}\ \forall\ i$, $S_j \in \mathbb{S}_n^{++}\ \forall\ j$.

Given a data point S, the sparse coding procedure finds an approximation $\hat{S} = \sum_{i=1}^{K} x_i A_i$, such that $\mathbf{x} \in \mathbb{R}_K^{+}$. We denote this as $\hat{S} = \mathbf{x} \otimes \mathcal{A}$. $\mathbf{x}_j$ denotes the coefficient vector corresponding to S_j.

The LogDet divergence [22] $D_{\text{ld}} : \mathbb{S}_n^{+} \times \mathbb{S}_n^{++} \to \mathbb{R}_+$ is defined by:

$$D_{\text{ld}}(X, Y) = \operatorname{tr}\left(XY^{-1}\right) - \log\det\left(XY^{-1}\right) - n. \quad (1)$$

1.3. Sparse Coding for Positive Definite Matrices

Given a dictionary $\mathcal{A}$ and data point S, the positive definite sparse coding problem [1] is given by

$$\min_{\mathbf{x} \geq 0} \quad D_{\text{ld}}\left(\mathbf{x} \otimes \mathcal{A}, S\right) + \lambda \|\mathbf{x}\|_1 \quad (2a)$$

$$\text{s.t.} \quad 0 \preceq \mathbf{x} \otimes \mathcal{A} \preceq S, \quad (2b)$$

where λ is the regularization parameter inducing sparsity on $\mathbf{x}$. The sparse coding is convex, and is solved by reducing the optimization in (2) to the MAXDET form [23], for which efficient Interior Point (IP) algorithms exist.

The residual reconstruction error $E = S - \hat{S}$ is always at least positive semidefinite, $E \succeq 0$, due to the constraint (2b). In reality, $E \succ 0$ since the smallest eigenvalue is of the order of 10^{-8} to 10^{-10}, or larger when λ increases.

In [1], the dictionary $\mathcal{A}$ was randomly initialized, or was formed from a subset of the training examples (for a classification setting). However, by adapting the dictionary to the data, the reconstruction accuracy of the sparse coding can be improved. In the next section, we describe our formulation and approach for learning the dictionary from the given data.

2. Positive Definite Dictionary Learning

In this section, we describe our formulation to learn the dictionary $\mathcal{A}$ that can best reconstruct the given data set $\mathcal{S}$ in a sparse fashion. Given a training data set $\mathcal{S} = \{S_j\}_{j=1}^{N}$, the problem of learning a dictionary $\mathcal{A} = \{A_i\}_{i=1}^{K}$ can be formulated as:

$$\min_{\mathcal{A}, X} \quad \sum_{j=1}^{N} D_{\text{ld}}\left(\mathbf{x}_j \otimes \mathcal{A}, S_j\right) + \lambda \|\mathbf{x}_j\|_1 \quad (3a)$$

$$\begin{aligned} \text{s.t.} \quad & x_{ij} \geq 0 && \forall\, i, j \\ & A_i \succeq 0 && \forall\, i \\ & 0 \preceq \mathbf{x}_j \otimes \mathcal{A} \preceq S_j && \forall\, j. \end{aligned} \quad (3b)$$

The dictionary learning problem is non-convex in both $(\mathcal{A}, X)$, but is convex in each argument given the other fixed. Hence we solve the optimization by alternating minimization, repeating the following two steps:

1. Given $\mathcal{S}$ and $\mathcal{A}$ fixed, solve for X.
2. Given $\mathcal{S}$ and X fixed, solve for $\mathcal{A}$.

The first stage involves sparse coding of each data point $S_j \in \mathcal{S}$ independently. The second stage comprises updating the dictionary $\mathcal{A}$. The dictionary is initialized with either randomly sampled data points from $\mathcal{S}$, or randomly generated positive definite matrices.

Since the decomposition $\mathbf{x} \otimes \mathcal{A}$ is unique only up to scale, and this can affect the constraint or regularization on $\mathbf{x}$ in the sparse coding stage, we normalize all the dictionary atoms A_i to have unit trace, $\mathrm{tr} A_i = 1$.

The dictionary update is inspired by the K-SVD algorithm of Aharon *et al*. [3]. We update each dictionary atom $A_i \in \mathcal{A}$ at a time, sequentially. During this time, the sparsity structure of the coefficients is kept constant, while allowing the non-zero coefficients of A_i to change. This subset of data points, for which x_{ij} is non-zero, is referred to as the *active set*, $\omega_i = \{j | 1 \leq j \leq N, x_{ij} \neq 0\}$.

2.1. Atom Update - Gradient Descent

To update atom A_i, we optimize the following using a steepest descent approach:

$$\min_{A_i \succeq 0} \sum_{j \in \omega_i} D_{\mathrm{ld}} \left(\mathbf{x}_j \otimes \mathcal{A}, S_j \right). \tag{4}$$

$$\mathbf{x}_j \otimes \mathcal{A} = \hat{S}_j = \sum_{i' \neq i} x_{i'j} A_{i'} + x_{ij} A_i. \tag{5}$$

Writing the objective in (4) as a function of A_i, we have

$$f(A_i) = \sum_{j \in \omega_i} \mathrm{tr} \left(x_{ij} A_i S_j^{-1} \right) - \log \det \hat{S}_j + C, \tag{6}$$

where C encompasses those terms independent of A_i. The gradient descent direction dA_i is given by:

$$dA_i = -\nabla f(A_i) = \sum_{j \in \omega_i} x_{ij} \left(\hat{S}_j^{-1} - S_j^{-1} \right). \tag{7}$$

Since $\hat{S}_j \preceq S_j$, we have $S_j^{-1} \preceq \hat{S}_j^{-1}$, yielding a positive semidefinite descent direction. The gradient descent update is, therefore,

$$A_i \leftarrow A_i + \alpha dA_i, \tag{8}$$

with stepsize $\alpha \geq 0$ determined using line search techniques.

2.2. Atom Update - Alternative Formulation

Here we propose an alternative atom update approach to the gradient descent method explained above. This method is much faster and in practice yields a better reduction in the original objective function compared to gradient descent. From (5),

$$\hat{S}_j = \sum_{i' \neq i} x_{i'j} A_{i'} + x_{ij} A_i = \hat{S}_j^{(i)} + x_{ij} A_i. \tag{9}$$

$\hat{S}_j^{(i)}$ is the reconstruction of S_j without the contribution of A_i, leading to the new residual,

$$E_j^{(i)} = S - \hat{S}_j^{(i)} = E_j + x_{ij} A_i. \tag{10}$$

The residual $E_j^{(i)}$ is strictly positive definite, since $x_{ij} > 0$ and $A_i \succeq 0$. Plugging this back into (4),

$$\min_{A_i \succeq 0} \sum_{j \in \omega_i} D_{\mathrm{ld}} \left(\hat{S}_j^{(i)} + x_{ij} A_i, \hat{S}_j^{(i)} + E_j^{(i)} \right). \tag{11}$$

Instead of directly minimizing (11), here we will attempt to minimize the LogDet divergence between the product $x_{ij} A_i$ and the new residual $E_j^{(i)}$.

$$\min_{A_i \succeq 0} \sum_{j \in \omega_i} D_{\mathrm{ld}} \left(x_{ij} A_i, E_j^{(i)} \right). \tag{12}$$

Since $D_{\mathrm{ld}}(\alpha X, \alpha Y) = D_{\mathrm{ld}}(X, Y)$,

$$\min_{A_i \succeq 0} \sum_{j \in \omega_i} D_{\mathrm{ld}} \left(A_i, \tilde{E}_j^{(i)} \right). \tag{13}$$

where $\tilde{E}_j^{(i)} = E_j^{(i)} / x_{ij}$.

The intuition behind this approach comes from the K-SVD algorithm of Aharon *et al*. [3], where the atom update step comprises fitting a new atom into a similar residual error. Although the LogDet divergence does not decouple in such a way, it follows the similar idea of finding the best atom A_i that fits the residual error $\tilde{E}_j^{(i)}$. As will be seen empirically, this produces a much greater reduction in the residual reconstruction error at each atom update step than the gradient descent optimization of the original objective function.

Writing out the expression for the LogDet divergence and ignoring the constant n,

$$\min_{A_i \succeq 0} \sum_{j \in \omega_i} \mathrm{tr} \left(A_i \left(\tilde{E}_j^{(i)} \right)^{-1} \right) - \log \det \left(A_i \left(\tilde{E}_j^{(i)} \right)^{-1} \right).$$

Taking the derivative with respect to A_i, and setting it to zero, we get an expression which corresponds to the harmonic mean of $\{\tilde{E}_j^{(i)}\}$, $j \in \omega_i$.

$$A_i = \left(\frac{1}{|\omega_i|} \sum_{j \in \omega_i} \left(\tilde{E}_j^{(i)} \right)^{-1} \right)^{-1}. \tag{14}$$

Since we require the atom A_i to be normalized by its trace, we can ignore the scaling term due to $|\omega_i|$. The updated atom A_i is therefore given by:

$$A_i = \left(\sum_{j \in \omega_i} \left(E_j^{(i)} / x_{ij} \right)^{-1} \right)^{-1}. \tag{15}$$

We also refer to this as the *parallel-sum* update, since (15) is the positive definite generalization of parallel sums for positive scalars [24].

2.3. Coefficient Correction

In the gradient descent-based update method, once the atom A_i is updated, each of the corresponding coefficients x_{ij} for $j \in \omega_i$ can be independently determined by an efficient line search [23]. In this step, it is important to respect

the original constraints $0 \preceq \hat{S} \preceq S$.

$$\min_{x_{ij} \geq 0} \quad D_{\text{ld}}\left(\hat{S}_j^{(i)} + x_{ij}A_i, S_j\right) + \lambda x_{ij} \quad (16a)$$

$$\text{s.t.} \quad 0 \preceq \left(\hat{S}_j^{(i)} + x_{ij}A_i\right) \preceq S_j \quad (16b)$$

In the first update method, the atom is only slightly perturbed, with a small positive semidefinite increment. Hence a line search for updating just the corresponding coefficients was sufficient. However, in the alternative update, since the updated atom is completely new, *i.e.*, more than just a perturbed version of the previous value, the coefficient distribution amongst the atoms for a given data point may not still be valid. Hence after each atom is updated, we sparse code all the data points using this atom once again. As will be seen empirically in Section 3.1, while the gradient-based update results in a very small decrease in the objective function, the alternative update, in spite of the fact that it did not attempt to minimize the original objective directly, results in a much greater reduction in the net residual reconstruction error.

2.4. Online Dictionary Learning

Both of the atom update equations in (7-8) and (15) are conducive to online generalization[1]. Suppose at time t we get a new data point S_t, which sparse coded over the existing dictionary $\mathcal{A}_{t-1}$ results in the reconstruction $\hat{S}_t$, with coefficients $x_{i,t}$ for $i = 1, \ldots, K$. The atoms which are used, *i.e.* $x_{i,t} > 0$, can be updated sequentially. The atoms which are unused do not change.

In the gradient descent method, the online update can be written as

$$A_{i,t} \leftarrow A_{i,t-1} + \alpha x_{i,t}\left(\hat{S}_t^{-1} - S_t^{-1}\right). \quad (17)$$

The online version of the parallel-sum update is given by

$$A_{i,t}^{-1} \leftarrow A_{i,t-1}^{-1} + \left(E_t^{(i)}/x_{i,t}\right)^{-1}. \quad (18)$$

where $E_t^{(i)}$ the residual computed using (10).

2.5. Computational Complexity

Since the dictionary update involves the inversion of at most N $n \times n$ matrices, each atom update step is $\mathbf{O}(Nn^3)$, in both the gradient descent and parallel sum update methods. Since n is usually of the order of $10 \sim 20$ for region covariance descriptors in most computer vision applications, it is still a very practical algorithm. Moreover, the region covariances provide a very low-dimensional condensed representation capable of greater performance than vector descriptors of much higher dimensions. For *e.g.*, in [4], consider the performance of 560-dimensional texton histograms vs. 5×5 region covariances for texture classification.

The sparse coding step accounts for a higher complexity, and a naïve implementation of the interior point algorithms for MAXDET [23] gives $\mathbf{O}(\max(n^2K^2, K^2))$. Current work involves development of a specialized implementation taking into account the problem structure in the positive definite sparse coding.

[1] if we account for the atom normalization with some book-keeping.

3. Experimental Results

In this section we demonstrate the use of the positive definite dictionary learning algorithms. Experiments on synthetic data compare the performance of the two different update methods, and show the reduction of residual error due to each method. Experiments based on computer vision applications of texture classification and face detection show that learning a concise dictionary representation not only improves classification performance, but also serves as a simple and straightforward object detector based on the reconstruction error of covariances extracted from candidate image regions.

3.1. Synthetic Experiments

In this experiment, we generate synthetic data consisting of 5×5 positive definite matrices, using the following approach:

- Generate a dictionary of K random positive definite atoms. K was chosen to be 8, 15, and 30 to reflect under-complete, complete and over-complete dictionaries.
- Generate $N = 300$ random sparse vectors $\mathbf{x}$ with $T = 4$ non-zero entries. Each $\mathbf{x}$ is generated by first sampling T out of K locations uniformly at random, and then populating those T entries by i.i.d. sampling from $\mathcal{U}(0, 1)$.
- Synthesize N training data points, where each point S is computed as the sample covariance of a set of $5n^2$ samples from the multivariate Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{x} \otimes A)$.
- Learn a new dictionary using the training points $\mathcal{S} = \{S_j\}_{j=1}^N$.

Since we do not (and in this sparse coding formulation, can not) impose that the training signals be sparse coded with $T = 4$ coefficients during the dictionary learning procedure, we will not exactly recover the same dictionary. We can only compare the residual of the learned dictionary with that of the original dictionary. Note that the original dictionary will not have a zero error, since there is an inverse Wishart [25] perturbation due to extracting the sample covariances.

In Figure 1, we show the objective function, the total residual error, decreasing with the number of iterations, for

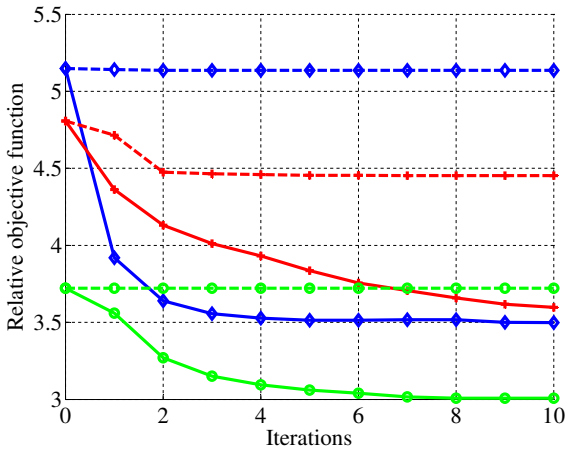

Figure 1: Normalized objective function for the two dictionary learning algorithms. The dotted lines denote gradient descent and the continuous lines represent the parallel-sum update. The plots are shown for $K = 8$ (blue,$\diamond$), $K = 15$ (red,$+$) and $K = 30$ (green,o).

both the training algorithms. The objective function is normalized with respect to that of the original dictionary, so that the true dictionary will have a total residual error of 1. The parallel-sum update algorithm gives a much better reduction in residual error, compared to the gradient descent update. In fact, for $K = 8$ and $K = 30$, there is very little reduction in the objective function, in the scale of that of the original dictionary. This is due to the fact that the gradient descent only perturbs the atoms very slightly during the update step, while the parallel-sum update enables the atoms to make huge jumps in the positive definite space.

For this experiment as well as the following, the regularization parameter λ is set to 10^{-3}. For the remaining experiments, we only use the parallel-sum update for the dictionary learning.

3.2. Texture Classification

In this section we demonstrate the effect of dictionary learning in a classification setting for region covariances. We compare the classification performance with learned dictionaries versus dictionaries formed by randomly sampling the training set.

We use a subset of the texture mosaics from the popular Brodatz texture dataset [26]. Intensity and gradient features $\{I, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|\}$ were extracted, and 5×5 covariance descriptors were computed over 32×32 blocks (in steps of 16 pixels) in the training images. A separate dictionary was learned for each class, with $N = 225$ training covariances, and $K = 5$ dictionary atoms, per class. Each class dictionary is trained independently, without using training samples from the other class in a discriminative fashion. For the random dictionary, K training points were sampled to fill in the dictionary.

Mosaic	5-NN Classification (%)	Random Dictionary (%)	Learned Dictionary (%)
1	8.94	2.76	**0.00**
2	8.13	6.99	**0.81**
3	**6.50**	12.85	8.13
4	17.89	26.02	**11.38**
5	17.07	9.43	**4.88**
10	**0.00**	0.69	0.23
11	**1.61**	12.51	2.99
12	1.38	0.78	**0.00**

Table 1: Texture classification results: Error rates (%) on the Brodatz texture dataset - 1-5 are the five 5-texture mosaics, and 10-12 are the three 2-texture mosaics.

The dictionary-based classification is performed as follows - each test point is independently sparse coded with each class dictionary, using the sparse coding formulation in (2). The residual error due to each class dictionary $\mathcal{A}_k$ is denoted as $D^*_{\text{ld}}(\mathbf{x}^* \otimes \mathcal{A}_k, S)$, where $\mathbf{x}^*$ is the optimal coefficient vector for that $(S, \mathcal{A}_k)$ pair. The test point is assigned the label k^* of the class whose dictionary results in the minimum residual reconstruction error.

$$k^* = \arg\min_k \; D^*_{\text{ld}}(\mathbf{x}^* \otimes \mathcal{A}_k, S).$$

Test points were generated by sampling 32×32 blocks from the mosaic image in regions of uniform texture. No spatial regularization of any sort is used in the classification procedure, and each test point is labeled independently.

The error rates of classification on the five 5-texture mosaics and the three 2-texture mosaics from [26] are shown in Table (1). A k-NN classifier with $k = 5$, retaining the entire training set and using the Riemannian geodesic distance [16], is also shown for comparison. The error rates for the random dictionary are averaged over 5 runs, each sampling a different dictionary. Note that the purpose of this experiment is not to show that covariance dictionaries are the best classifiers for texture. Rather, the message to take away from this experiment is that learning the dictionary from the data provides a substantial performance gain compared to randomly sampled dictionaries, for classification applications. This is well-known and accepted for vector dictionary learning, and is demonstrated here for positive definite dictionaries.

The residual reconstruction error for training the 5-class dictionaries for the first five Brodatz texture mosaics are shown in Figure 2. The objective function is normalized with respect to the residual error before training, and hence at iteration 0, the objective function value will be 1. The plot is to demonstrate that the residual error decrease due to the parallel-sum update algorithm is also very strong for practical datasets, and as can be seen, the total residual error is reduced to almost half of the original.

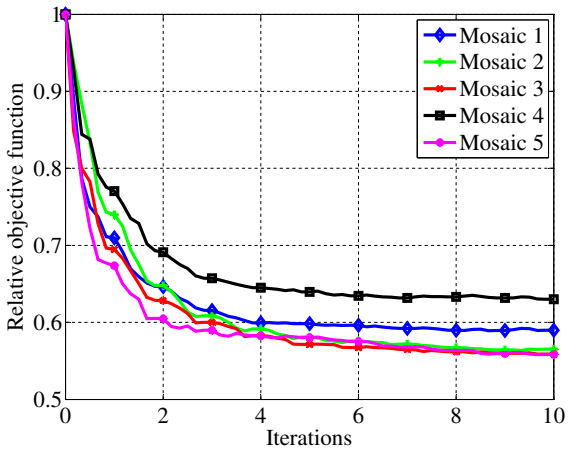

Figure 2: Normalized objective function over the parallel-sum dictionary learning iterations for the five 5-texture mosaics. The objective function is the total of residual errors of all 5 dictionaries, at each stage. The plot also shows the objective function values during the dictionary update, after updating each atom.

3.3. Face Detection

In the previous sections we have seen empirically that the residual reconstruction error in sparse coding is reduced by learning positive definite dictionaries, and classification performance with learned covariance dictionaries is much better than random non-adaptive dictionaries. In this section, we explore the use of the covariance dictionary for face detection. For training, we use images from the FERET face database [27]. 7 images, each from 109 subjects were chosen, and 19×19 covariance descriptors were extracted. The features used were $\{x, y, I\}$ and Gabor filter responses, $g_{00}(x, y), \ldots, g_{71}(x, y)$, with 8 orientations and 2 scales. The preprocessing and feature extraction are performed following the approach of Pang *et al.* [13]. The $N = 763$ training covariances were used to train a dictionary of size $K = 38$.

For testing, images from the GRAZ01 *person* dataset [28] were processed to extract the same spatial, intensity and Gabor features. Covariance descriptors were computed over regularly spaced, overlapping windows (60×60, in steps of 15 pixels). All the covariances were sparse coded with the learned face dictionary, and the residual error $D_{\text{ld}}(\hat{S}, S)$ is used to obtain a detection score at each window. The score was computed as

$$\text{score} = e^{-\frac{1}{2}\left(\frac{D_{\text{ld}}(\hat{S},S)}{\sigma_d^2}\right)^2}, \qquad (19)$$

where σ_d is the bandwidth, computed as the standard deviation of the residual errors in the entire image.

The original images and the corresponding *face score* images are shown in Figure 3. Note that this is a very simple and straightforward application of the region covariance

Figure 3: Face detection results: Sample images (from the GRAZ01 [28] dataset) and their corresponding face score maps are shown. High values of the face score (white) indicate the likelihood of a face being present, centered at that location. The score images are normalized with respect to their maximum value, for viewing reasons.

dictionary learning, with no complex post-processing. The residual error from sparse coding an image region with the *face dictionary* gives us an good estimate of the probability of that window being a face. A mode-finding procedure over this probability map will give the best face detections. Future work includes searching over windows at multiple scales, as well as learning multiscale dictionaries in terms of the covariance descriptors.

4. Conclusions & Future Work

In this paper, we proposed a novel dictionary learning methodology for positive definite matrices. The dictionary learning was formulated with an alternating minimization approach, and two different atom update procedures were elaborated. Update equations for online dictionary learning were also presented. Synthetic experiments were shown to validate the learning approaches. Practical computer vision examples were also demonstrated in the classification as well as detection setting, both indicating the performance of trained covariance dictionaries.

Future work includes analysis of regret bounds for the online dictionary learning updates, as well as faster methods for coefficient updates. Multiscale extensions either in the covariance descriptors themselves or in the dictionary learning procedure would be very suitable for detection applications such as that shown here. While we manually fix the number of dictionary atoms here, automatic selection of dictionary size is another interesting issue to be addressed. Scalability of the learning methods to much larger matrix dimensions is also being investigated.

Acknowledgments. This material is based upon work supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under contract #911NF-08-1-0463 (Proposal 55111-CI), and the National Science Foundation through grants #IIP-0443945, #CNS-0821474, #IIP-0934327, #CNS-1039741, #IIS-1017344, #IIP-1032018, #SMA-1028076, and #IIS-0916750.

References

[1] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos, "Tensor sparse coding for region covariances," in *ECCV 2010*, Springer Berlin / Heidelberg.

[2] K. Guo, P. Ishwar, and J. Konrad, "Action recognition using sparse representation on covariance manifolds of optical flow," in *7th IEEE Intl. Conf. on Advanced Video & Signal-based Surveillance, 2010*, pp. 188 –195, Aug.-Sept. 2010.

[3] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, pp. 4311 –4322, Nov. 2006.

[4] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *ECCV 2006*, Springer Berlin / Heidelberg.

[5] F. Porikli and O. Tuzel, "Fast construction of covariance matrices for arbitrary size image windows," in *IEEE Intl. Conf. on Image Processing, 2006*, pp. 1581 –1584, Oct. 2006.

[6] H. Wildenauer, B. Micuk, and M. Vincze, "Efficient texture representation using multi-scale regions," in *ACCV 2007*, vol. 4843, Springer Berlin / Heidelberg, 2007.

[7] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 1713 –1727, Oct. 2008.

[8] F. Porikli and T. Kocak, "Robust license plate detection using covariance descriptor in a neural network framework," in *IEEE Intl. Conf. on Video and Signal Based Surveillance, 2006*, pp. 107–107, Nov. 2006.

[9] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," in *IEEE Computer Society Conf. on Computer Vision & Pattern Recognition, 2006*, vol. 1, pp. 728 – 735, June 2006.

[10] C. Prakash, B. Paluri, S. Nalin Pradeep, and H. Shah, "Fragments based parametric tracking," in *ACCV 2007*, Springer Berlin / Heidelberg.

[11] Y. Ma, B. Miller, and I. Cohen, "Video sequence querying using clustering of objects appearance models," in *Advances in Visual Computing*, Springer Berlin / Heidelberg.

[12] R. Sivalingam, V. Morellas, D. Boley, and N. Papanikolopoulos, "Metric learning for semi-supervised clustering of region covariance descriptors," in *3rd ACM/IEEE Intl. Conf. on Distributed Smart Cameras, 2009*, pp. 1 –8, Aug.-Sept. 2009.

[13] Y. Pang, Y. Yuan, and X. Li, "Gabor-based region covariance matrices for face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, pp. 989 –993, July 2008.

[14] H. Huo and J. Feng, "Face recognition via AAM and multi-features fusion on riemannian manifolds," in *ACCV 2009*, Springer Berlin / Heidelberg.

[15] K. Guo, P. Ishwar, and J. Konrad, "Action change detection in video by covariance matching of silhouette tunnels," in *IEEE Intl. Conf. on Acoustics Speech & Signal Processing, 2010*, pp. 1110 –1113, Mar. 2010.

[16] X. Pennec, P. Fillard, and N. Ayache, "A riemannian framework for tensor computing," *Int. J. Comput. Vision*, vol. 66, pp. 41–66, January 2006.

[17] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *IEEE Conf. on Computer Vision & Pattern Recognition, 2008.*, pp. 1 –8, June 2008.

[18] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *IEEE Intl. Conf. on Computer Vision & Pattern Recognition, 2010.*, pp. 3501 –3508, June 2010.

[19] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *26th Annual Intl. Conf. on Machine Learning*, 2009.

[20] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Trans. Signal Process.*, vol. 58, pp. 1553 – 1564, Mar. 2010.

[21] L. Bar and G. Sapiro, "Hierarchical dictionary learning for invariant classification," in *IEEE Intl. Conf. on Acoustics Speech & Signal Processing, 2010.*, pp. 3578 –3581, Mar. 2010.

[22] B. Kulis, M. Sustik, and I. Dhillon, "Learning low-rank kernel matrices," in *23rd Intl. Conf. on Machine Learning.*, pp. 505–512, ACM, 2006.

[23] L. Vandenberghe, S. Boyd, and S.-P. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 19, pp. 499–533, April 1998.

[24] W. N. Anderson and R. J. Duffin, "Series and parallel addition of matrices," *Journal of Mathematical Analysis & Applications*, vol. 26, no. 3, pp. 576 – 594, 1969.

[25] J. WISHART, "The generalised product moment distribution in samples from a normal multivariate population," *Biometrika*, vol. 20A, no. 1-2, pp. 32–52, 1928.

[26] T. Randen and J. H. Husøy, "Filtering for texture classification: A comparative study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, pp. 291–310, April 1999.

[27] P. J. Phillips, H. Moon, P. Rauss, and S. A. Rizvi, "The FERET evaluation methodology for face-recognition algorithms," in *IEEE Intl. Conf. on Computer Vision & Pattern Recognition 1997*.

[28] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer, "Weak hypotheses and boosting for generic object detection and recognition," in *ECCV 2004*, Springer Berlin / Heidelberg.