Tensor Sparse Coding for Region Covariances

Ravishankar Sivalingam, Daniel Boley, Vassilios Morellas, and Nikolaos Papanikolopoulos

Department of Computer Science & Engineering,
University of Minnesota, Minneapolis, MN 55455, USA
{ravi,boley,morellas,npapas}@cs.umn.edu

Abstract. Sparse representation of signals has been the focus of much research in the recent years. A vast majority of existing algorithms deal with vectors, and higher–order data like images are dealt with by vectorization. However, the structure of the data may be lost in the process, leading to a poorer representation and overall performance degradation. In this paper we propose a novel approach for sparse representation of positive definite matrices, where vectorization will destroy the inherent structure of the data. The sparse decomposition of a positive definite matrix is formulated as a convex optimization problem, which falls under the category of determinant maximization (MAXDET) problems [1], for which efficient interior point algorithms exist. Experimental results are shown with simulated examples as well as in real–world computer vision applications, demonstrating the suitability of the new model. This forms the first step toward extending the cornucopia of sparsity-based algorithms to positive definite matrices.

Keywords: Positive definite matrices, region covariances, sparse coding, MAXDET optimization

1 Introduction

In the past decade or so there has been a deluge of research on sparse representations of signals [2–4] and recovery of such sparse signals from noisy and/or under-sampled observations [5, 6]. Much of the work has been associated with vector-valued data, and higher–order signals like images (2-D, 3-D, or higher) have been dealt with primarily by vectorizing them and applying the aforementioned vector methods. See [7] for a review of a few representative examples of sparse representation in computer vision and pattern recognition applications. However, more recently some have realized the advantages of maintaining the higher–order data in their original form [8] in order to preserve some inherent ordering, which may be destroyed upon vectorization.

One such data type consists of $n \times n$ symmetric positive semi-definite matrices ($\mathbf{S}_+^n$). The kernel matrix in many popular 'kernelized' machine learning algorithms [9] belongs to this class. In medical imaging, the revolutionary new field of Diffusion Tensor Imaging (DTI) represents each voxel in a 3-D brain scan as a 3×3 positive definite matrix, called the diffusion tensor, whose principal

eigenvector gives the direction of water diffusion in that region. More recently in the image processing and computer vision community, a new feature known as the region covariance descriptor has emerged [10, 11], which represents an image region by the covariance of n-dimensional feature vectors at each pixel in that region. This is currently being used in conjunction with machine learning algorithms for human detection and tracking, object recognition, texture classification, query-based retrieval of image regions, and much more [12].

In this paper we propose a novel approach for sparse representation of positive definite matrices, named *tensor sparse coding*[1]. The sparse decomposition of a positive definite matrix in terms of a given dictionary is formulated as a convex optimization problem, which belongs to the class of MAXDET problems [1] and for which efficient interior point methods are available. We believe that this extension of sparse coding techniques to the space of positive definite matrices will benefit the development of sparsity-related algorithms tailored to these problem domains as well. This forms the first step toward extending the cornucopia of sparsity-based algorithms to this new class of data points, and all algorithms that primarily use the sparse coding stage follow readily from our approach.

The rest of the paper is organized as follows: In the remainder of this section, we provide a brief description about region covariances, and related work on these descriptors. Section 2 describes the problem statement, and our tensor sparse coding approach is explained in Sect. 3. Experiments on both synthetic and actual datasets are shown in Sect. 4, wrapping up with our conclusions and future research directions in Sect. 5.

1.1 Region Covariance Descriptors

Region covariances were introduced by Tuzel et al. [10] as a novel region descriptor for object detection and classification. Given an image $\mathcal{I}$, let ϕ define a mapping function that extracts an n-dimensional feature vector z_i from each pixel $i \in \mathcal{I}$, such that

$$\phi(I, x_i, y_i) = z_i \ , \tag{1}$$

where $z_i \in \mathbf{R}^n$, and (x_i, y_i) is the location of the i^{th} pixel. A given image region R is represented by the $n \times n$ covariance matrix C_R of the feature vectors $\{z_i\}_{i=1}^{|R|}$ of the pixels in region R. Thus the region covariance descriptor is given by

$$C_R = \frac{1}{|R| - 1} \sum_{i=1}^{|R|} (z_i - \mu_R)(z_i - \mu_R)^T \ , \tag{2}$$

where, μ_R is the mean vector,

$$\mu_R = \frac{1}{|R|} \sum_{i=1}^{|R|} z_i \ . \tag{3}$$

[1] from the 'tensor' in 'diffusion tensor' [13].

The feature vector z usually consists of color information (in some preferred color–space, usually RGB) and information about the first and higher order spatial derivatives of the image intensity, depending on the application intended.

Although covariance matrices can be positive semi–definite in general, the covariance descriptors themselves are regularized by adding a small constant multiple of the identity matrix, making them strictly positive definite. Thus, the region covariance descriptors belong to $\mathbf{S}^n_{++}$, the space of $n \times n$ positive definite matrices which forms a connected Riemannian manifold. Given two covariance matrices C_i and C_j, the Riemannian distance metric $d_{\text{geo}}(C_i, C_j)$ gives the length of the geodesic connecting these two points on this manifold. This is given by [13],

$$d_{\text{geo}}(C_i, C_j) = \left\| \log\left(C_i^{-1/2} C_j C_i^{-1/2} \right) \right\|_F \quad , \tag{4}$$

where $\log(\cdot)$ represents the matrix logarithm and $\| \cdot \|_F$ is the Frobenius norm. Many existing classification algorithms for region covariances use the geodesic distance in a K-nearest-neighbor framework. The geodesic distance can also be used with a modified K-means algorithm for clustering.

Methods for fast computation of region covariances using *integral images* [11] enable the use of these compact features for many practical applications that demand real–time performance. For texture characterization, spatial derivatives are suitable features [10], whereas for face recognition, region covariances are constructed from outputs of a bank of Gabor filters [14]. Hu et al. [15] use covariance descriptors for probabilistic tracking using particle filtering. Palaio and Batista [16] also perform multi–object tracking using region covariances and particle filters. In [17], Paisitkriangkrai et al. boost the covariance features to improve the classification accuracy. In [12], Tuzel et al. use LogitBoost on the covariance descriptors for pedestrian detection. Sivalingam et al. [18] learn a modified distance metric over the manifold from pairwise constraints, for semi-supervised clustering.

2 Problem Statement

We begin with a known dictionary consisting of k $n \times n$ positive definite matrices $\mathcal{A} = \{A_i\}_{i=1}^k$, where each $A_i \in \mathbf{S}^n_{++}$ is referred to as a dictionary atom. Given a positive definite matrix S, our goal is to represent the new matrix as a linear combination of the dictionary atoms, *i.e.*,

$$S = x_1 A_1 + x_2 A_2 + \ldots + x_k A_k = \sum_{i=1}^{k} x_i A_i \quad , \tag{5}$$

where $x = (x_1, x_2, \ldots, x_k)^T$ is the vector of coefficients.

Since only a non-negative linear combination of positive definite matrices is guaranteed to yield a positive definite matrix, we impose a non-negativity constraint on the coefficient vector x, $x \in \mathbf{R}^k_+$.

It is to be noted that the given matrix S need not always be exactly representable as a sparse non-negative linear combination of the dictionary atoms. Hence, we will aim to find the best approximation $\hat{S}$ to S, by minimizing the residual approximation error in some sense. Clearly, we require the approximation $\hat{S}$ to be positive definite,

$$\hat{S} \succeq 0 \Longrightarrow x_1 A_1 + x_2 A_2 + \ldots + x_k A_k \succeq 0 \ . \tag{6}$$

Although this would be ensured by construction, due to the non-negativity of x and the strictly positive definite dictionary atoms, we leave this constraint in for reasons apparent later in the discussion.

We further require that the representation be sparse, *i.e.*, S is to be represented by a sparse linear combination of the dictionary atoms. To this effect, we impose a constraint on the ℓ_0 "pseudo-norm" of x,

$$\|x\|_0 \leq T \ , \tag{7}$$

where T is a pre-defined parameter, denoting the maximum number of non-zero elements of x.

3 Approach

3.1 The LogDet Divergence

If X^{-1} and Y^{-1} are the covariance matrices of two multivariate Gaussians P_X and P_Y with the same (or zero) mean, then the KL-divergence between the two distributions [19] is given by,

$$KL\left(P_Y \| P_X\right) = \frac{1}{2} D_{\mathrm{ld}}(Y^{-1}, X^{-1}) = \frac{1}{2} D_{\mathrm{ld}}(X, Y) \ , \tag{8}$$

where $D_{\mathrm{ld}}(\cdot)$ is the LogDet (or Burg matrix) divergence [20], given by,

$$D_{\mathrm{ld}}(X, Y) = tr\left(XY^{-1}\right) - \log\det\left(XY^{-1}\right) - n \ . \tag{9}$$

Here n is the dimension of the matrices X and Y, and $tr(\cdot)$ denotes the trace of the matrix. Note that, in general, the divergence is asymmetric, *i.e.,* $D_{\mathrm{ld}}(X, Y) \neq D_{\mathrm{ld}}(Y, X)$.

Further, it is a well-known fact that there exists a bijection between regular exponential families and a large class of Bregman divergences, called regular Bregman divergences [21]. For example, the squared-error loss function which is minimized in vector sparse coding methods comes from the squared Euclidean distance, which is the Bregman divergence corresponding to the multivariate Gaussian distribution. Thus, the minimization of a squared error objective function corresponds to the assumption of Gaussian noise. The Wishart distribution [22], which is a distribution over $n \times n$ positive definite matrices, with positive definite parameter matrix Θ and degrees of freedom $p \geq n$, is given by

$$\Pr(X|\Theta, p) = \frac{|X|^{(p-n-1)/2} \exp\left(-\frac{1}{2} tr(\Theta^{-1} X)\right)}{2^{pn/2} |\Theta|^{p/2} \Gamma_n(p/2)} \ , \tag{10}$$

where $|\cdot|$ is the determinant. The Bregman divergence corresponding to the Wishart distribution is the LogDet divergence $D_{\mathrm{ld}}(X, \Theta)$ [23]. If we assume that the positive definite matrix is drawn from a Wishart distribution, we can minimize the LogDet divergence between the matrix and its approximation. Further, the LogDet divergence (9) is convex in X (but not in Y) and hence is a perfect candidate for our problem formulation.

3.2 Formulation

Motivated by the above-mentioned reasons, we define our optimization problem as one which tries to minimize the LogDet divergence $D_{\mathrm{ld}}(\hat{S}, S)$ between the approximation $\hat{S}$ and the given matrix S.

$$D_{\mathrm{ld}}(\hat{S}, S) = tr\left(\left(\sum_{i=1}^{k} x_i A_i\right) S^{-1}\right) - \log\det\left(\left(\sum_{i=1}^{k} x_i A_i\right) S^{-1}\right) - n \ . \quad (11)$$

For numerical stability, we ensure that the arguments are also symmetric. Since the trace and the log det are invariant under a similarity transformation, we map $X \mapsto S^{-1/2} X S^{1/2}$, where X is the argument.

$$\begin{aligned} D_{\mathrm{ld}}(\hat{S}, S) = tr\left(S^{-1/2}\left(\sum_{i=1}^{k} x_i A_i\right) S^{-1/2}\right) & \\ - \log\det\left(S^{-1/2}\left(\sum_{i=1}^{k} x_i A_i\right) S^{-1/2}\right) - n & \quad (12) \\ = tr\left(\sum_{i=1}^{k} x_i \hat{A}_i\right) - \log\det\left(\sum_{i=1}^{k} x_i \hat{A}_i\right) - n \ , & \quad (13) \end{aligned}$$

where $\hat{A}_i = S^{-1/2} A_i S^{-1/2}$. Therefore,

$$D_{\mathrm{ld}}(\hat{S}, S) = \sum_{i=1}^{k} x_i \ tr\hat{A}_i - \log\det\left(\sum_{i=1}^{k} x_i \hat{A}_i\right) - n \ . \quad (14)$$

We discard n from the objective function as it is a constant.

The best approximation $\hat{S}$ would result in an exactly positive semidefinite residual $E = S - \hat{S}$, so that incrementing any x_i is not possible without pushing the residual to be indefinite, *i.e.*, leading to $\hat{S} \not\preceq S$, since subtracting even the "smallest" positive definite matrix from a positive semidefinite matrix will make it indefinite. Therefore, the minimum eigenvalue of the residual $\lambda_{\min}(S - \hat{S})$ should be as close to zero as possible. Hence we impose the constraint

$$\hat{S} \preceq S \quad \text{or} \quad x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \preceq I_n \ , \quad (15)$$

where I_n is the $n \times n$ identity matrix. Combining with (6), we get

$$0 \preceq x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \preceq I_n \ . \quad (16)$$

Since the constraint (7) is non-convex, a convex relaxation of this constraint involves minimizing the ℓ_1 norm of x instead of the ℓ_0 pseudo-norm. Under certain assumptions [24], the ℓ_1 penalty has been proven to yield the same (or similar) results as minimizing $\|x\|_0$ for sparse decompositions.

Combining all the above constraints with the objective function we wish to minimize, we have the following optimization problem:

$$\min_x \quad \sum_{i=1}^{k} x_i \; tr\hat{A}_i - \log\det\left(\sum_{i=1}^{k} x_i\hat{A}_i\right) + \lambda\|x\|_1 \tag{17}$$

$$\text{s.t.} \quad x \geq 0 \tag{18}$$

$$x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \;\succeq\; 0 \tag{19}$$

$$x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \;\preceq\; I_n \;\;, \tag{20}$$

where $\lambda \geq 0$ is a parameter which represents a trade–off between a sparser representation and a closer approximation. Further, since the x_i's are non–negative, the ℓ_1 norm simply becomes the sum of the components of x, *i.e.*,

$$\|x\|_1 = \sum_{i=1}^{k} x_i \;\;, \tag{21}$$

yielding the optimization problem :

$$\min_x \quad \sum_{i=1}^{k} x_i\left(tr\hat{A}_i + \lambda\right) - \log\det\left(\sum_{i=1}^{k} x_i\hat{A}_i\right) \tag{22}$$

$$\text{s.t.} \quad x \geq 0 \tag{23}$$

$$x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \;\succeq\; 0 \tag{24}$$

$$x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \;\preceq\; I_n \;\;. \tag{25}$$

Concurrent with other vector sparse coding techniques, we may express this optimization problem in an alternate form which puts a different form of constraint on the sparsity of x. Instead of a penalty term $\lambda\|x\|_1$ in the objective function, we may enforce the sparsity by adding the constraint $\|x\|_1 \leq T$ resulting in the following variation of the above problem:

$$\min_x \quad \sum_{i=1}^{k} x_i tr\hat{A}_i - \log\det\left(\sum_{i=1}^{k} x_i\hat{A}_i\right) \tag{26}$$

$$\text{s.t.} \quad x \geq 0 \tag{27}$$

$$\sum_{i=1}^{k} x_i \leq T \tag{28}$$

$$x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \;\succeq\; 0 \tag{29}$$

$$x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \;\preceq\; I_n \;\;. \tag{30}$$

We denote the optimization problem defined by (22)–(25) as Type I, and that defined by (26)–(30) as Type II.

3.3 The MAXDET problem

The above formulations of tensor sparse coding fall under a general class of optimization problems known as determinant maximization, or MAXDET, problems [1], of which semi-definite programming (SDP) and linear programming (LP) are special cases. The MAXDET problem is defined as:

$$\min_{x} \quad c^T x + \log\det G(x)^{-1} \tag{31}$$

$$\text{s.t.} \quad G(x) \triangleq G_0 + x_1 G_1 + \ldots + x_k G_k \succ 0 \tag{32}$$

$$F(x) \triangleq F_0 + x_1 F_1 + \ldots + x_k F_k \succeq 0 \ , \tag{33}$$

where $x \in \mathbf{R}^k$, $G_i \in \mathbf{S}^n$ and $F_i \in \mathbf{S}^N$. These problems are convex, well-behaved, and efficient interior point methods exist for solving them. Note that the $G(x)$ inside the $\log\det$ term also explicitly appears as a constraint in the standard form of the MAXDET problem, leading to our inclusion of the same in our formulation.

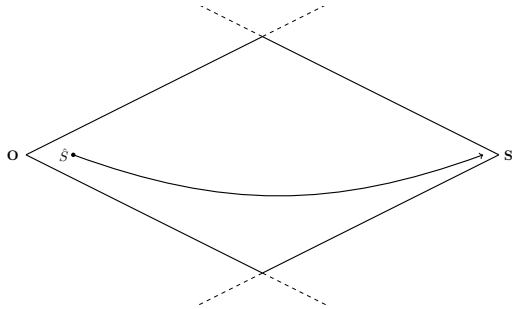

Fig. 1. The feasible set consists of the region of intersection of two positive semidefinite cones, one centered at the origin O, and the other an inverted cone centered at S. $\hat{S}$ lies in the strict interior of this cone, and is pushed towards S by the $\log\det$ term in the objective. The linear term serves as a regularizer on the coefficients x_i.

Thus, we have formulated two variations of our tensor sparse coding problem (Type I and II), both of which are convex and of the standard MAXDET form. The approximation $\hat{S}$ lies inside the intersection of the two positive semidefinite cones, one centered at the origin and the inverted positive semidefinite cone centered at S, which forms a closed convex set (See Fig. 1). The $-\log\det$ term in the objective function pushes the approximation $\hat{S}$ toward S, motivating a better approximation. We use `CVX` [25] to solve the MAXDET optimization problem.

4 Experiments

4.1 Numerical Example

Our first set of experiments were run on a synthetic data set, comprised of precision (inverse of covariance) matrices. We start with an $n \times n$ covariance matrix C

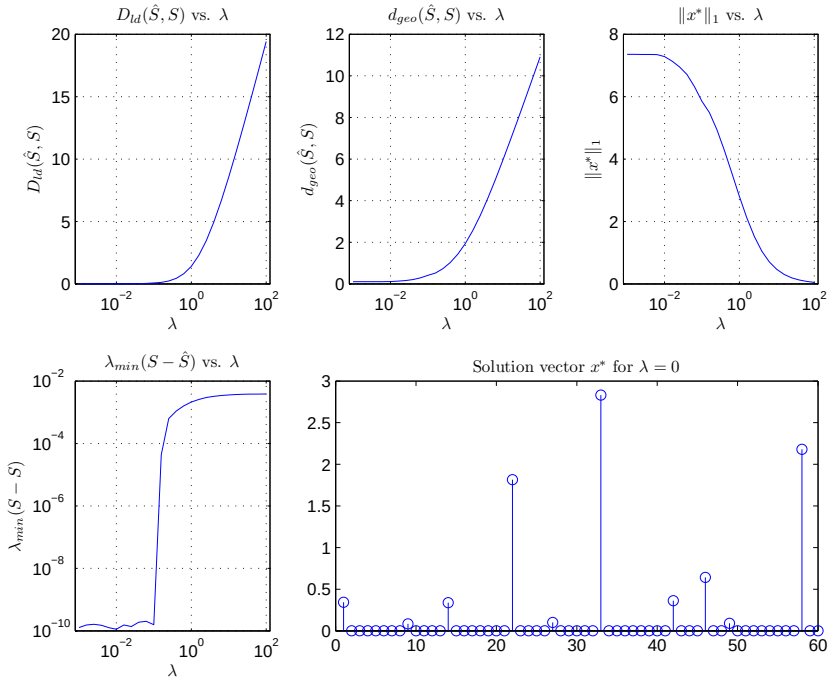

Fig. 2. Plot of the various quantities vs. λ for $n = 5$, $k = 60$. We show $D_{\mathrm{ld}}(\hat{S}, S)$, $d_{\mathrm{geo}}(\hat{S}, S)$, $\|x^*\|_1$, as well as $\lambda_{\min}(S - \hat{S})$, plotted in logarithmic scale. The λ values are varied logarithmically. The solution vector x^* in the unconstrained case is also shown on the right, and is observed to be sparse even without explicitly enforcing any sparsity.

and generate sets of samples from a multivariate Gaussian distribution $\mathcal{N}(0, C)$. There are $O(n^2)$ samples per set, from which we compute the inverse covariance for each of these sets. These precision matrices forms our data set. We select k of these matrices to form our dictionary $\mathcal{A} = \{A_i\}_{i=1}^k$. The sample point S to be sparse-coded is also generated in this manner. The precision matrix of a multivariate Gaussian distribution follows a Wishart distribution [22], and therefore our optimization problem is well suited to this model. The quantities we consider to represent the performance of the reconstruction are the LogDet Divergence $D_{\mathrm{ld}}(\hat{S}, S)$, the geodesic distance $d_{\mathrm{geo}}(\hat{S}, S)$, the ℓ_1 norm of the optimal coefficient vector $\|x^*\|_1$ and $\lambda_{\min}(S - \hat{S})$, the minimum eigenvalue of the residual.

Effect of normalization. In vector sparse-coding and dictionary learning, the dictionary atoms are usually normalized to have unit length. In a similar fashion, we tried different ways to normalize the atoms in our dictionary, *viz.*, by spectral norm, $\|A_i\|_2$, Frobenius norm, $\|A_i\|_F$, or by trace, $tr(A_i)$. Since all matrix norms are equivalent, we only get a proportional change in the quality of approximation,

as is expected. Throughout the rest of this section, we stick to normalization by spectral norm.

Effect of sparsity constraints. Figure 2 shows the effect of varying λ on the quality of reconstruction, under the Type I problem. The geodesic distance can be seen to vary in a smooth and similar fashion to the LogDet divergence, reaffirming our choice of objective function. We also show the actual solution vector x^* for $\lambda = 0$, where it can be seen that even the unconstrained case results in a sparse solution vector. This is due to the fact that we require a non-negative coefficient vector, and it is widely noted in the vector-domain that non-negative decompositions result in sparsity, under certain conditions [26–28].

4.2 Classification Experiments

We evaluate the tensor sparse coding algorithm in a classification framework, where the training data is used as a dictionary $\mathcal{A}$, and the test point S is approximated by a sparse non-negative linear combination of the dictionary atoms. In all the following experiments, we use the Type I objective function for sparse coding, with $\lambda = 10^{-3}$.

The datasets used are comprised of region covariance descriptors from various applications such as human appearance clustering, texture classification and face recognition. The classification is performed in 4 different ways as follows:

- Geodesic KNN – K-nearest-neighbor classification with $K = 5$, using the Riemannian geodesic distance.
- Kernel SVM classification – Using the multi-class SVM approach, with the kernel matrix computed as

$$K(C_i, C_j) = \exp\left(-\frac{d^2_{\text{geo}}(C_i, C_j)}{2\sigma^2}\right) \quad , \tag{34}$$

  with $\sigma = 1$, we perform classification of the test set with the help of the software `LIBSVM` [29] for the SVM classification.
- $SC + WLV$ – In this method, the coefficient vector x is used as a weight vector to vote for the different class labels. In other words, the label k^* of S is computed as

$$k^* = \arg\max_k \sum_{A_i \in \mathcal{C}_k} x_i \quad , \tag{35}$$

  where $\mathcal{C}_k$ denotes class k. Each dictionary atom A_i votes with its own class label, and its vote is weighted by the corresponding coefficient x_i. The class which gets the highest vote is assigned as the class label of S, hence the name *Weighted Label Voting* (WLV).
- $SC + REC$ – Another method involving sparse coding is adapted from [30], where after the sparse coefficient vector is obtained, the positive definite matrix is reconstructed from atoms (and corresponding coefficients) from

each class in the dictionary separately. The class which gives the minimum residual reconstruction error (REC), in terms of the LogDet divergence, is assigned to the new descriptor S.

As mentioned earlier, much of the relevant literature on region covariances use the geodesic KNN for classification. Also, the SVM is powerful and popular classifier in computer vision applications. Hence our choice of these two algorithms to compare our results. The geodesic KNN and the kernel SVM classification are performed directly on the covariance descriptors. The last two methods involve sparse coding, and since our problem formulation is derived under the LogDet divergence and corresponds to the precision matrix, we perform the sparse coding over the inverse of the covariance descriptors.

Human Appearance Descriptors. We use a subset of the 18-class *Cam5* dataset from [18], from which we choose the 16 classes which contain at least 10 data points each. From each of these 16 classes, we select 5 points for training and 5 for testing. The dictionary $\mathcal{A}$ is therefore comprised of $k = 80$ atoms. The descriptors are 5×5 covariances computed from the {R,G,B,I_x,I_y} features at each pixel corresponding to the human foreground blobs. The classification accuracy for this dataset averaged over 100 random train-test splits is shown in Table 1. The sparse coding results provide a notable increase in accuracy compared to the KNN or SVM techniques.

.

Table 1. Classification accuracy for the *Cam5* dataset.

Classifier	Mean Accuracy (%)	Std. Dev (%)
Geodesic KNN	62.76	2.59
Kernel SVM	72.59	4.94
SC + WLV	75.54	3.17
SC + REC	77.20	3.06

Gabor-based Region Covariances for Face Recognition. We compare the classification performance of the $SC + WLV$ and $SC + REC$ methods with the geodesic KNN for the process of face recognition. We test over a subset of the FERET face database [31], using similar pre-processing as in Pang et al. [14]. The images from 10 subjects are taken from the grayscale FERET database, and correspond to the two letter codes *'ba'*, *'bd'*, *'be'*, *'bf'*, *'bg'*, *'bj'*, and *'bk'*. In each experiment, 3 of these are taken as training images, and the remaining 4 as test images, yielding a total of $\binom{7}{3} = 35$ different train-test splits.

The images are convolved with Gabor filters with 8 orientations $u = 0, \ldots, 7$, and up to 3 scales $v = 0, 1, 2$. The Gabor filters are constructed with the same

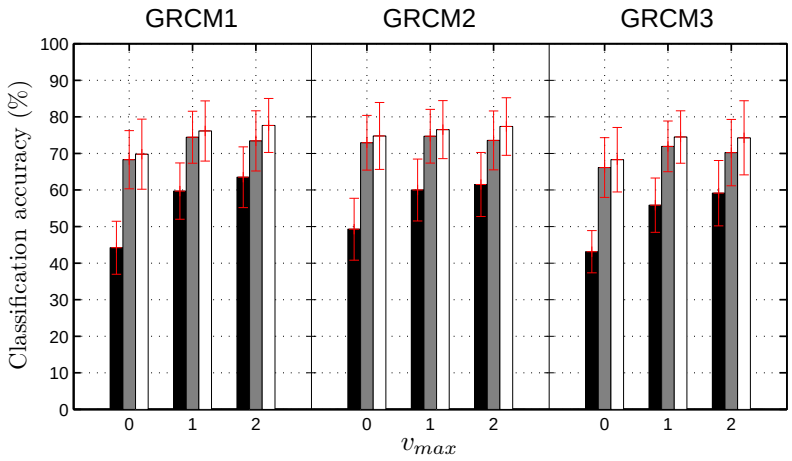

Fig. 3. Classification accuracy for 10 classes with the geodesic KNN (black), $SC{+}WLV$ (gray) and the $SC + REC$ (white) classifiers, for the Gabor-based region covariance datasets GRCM1, GRCM2, and GRCM3. The results are averaged over 35 trials, and 1σ standard deviation bars are shown.

parameters as explained in [14]. Let $g_{uv}(x, y)$ denote the Gabor-filter output at orientation u and scale v. Let $v_{\max}$ be the maximum scale of the Gabor filter in a dataset. We compute 3 datasets of region covariances for each value of $v_{\max} = 0, 1, 2$, comprised of different sets of features, as follows:

- GRCM1 – $\{\ x,\ y,\ g_{00}(x,y), \ldots, g_{7v_{\max}}(x,y)\ \}$
- GRCM2 – $\{\ x,\ y,\ I,\ g_{00}(x,y), \ldots, g_{7v_{\max}}(x,y)\ \}$
- GRCM3 – $\{\ g_{00}(x,y), \ldots, g_{7v_{\max}}(x,y)\ \}$

yielding a total of 9 different datasets. For each of these 9 datasets, we average over the 35 runs of distinct train-test splits. The classification accuracies for the geodesic KNN and the two tensor sparse coding classification algorithms are shown in Fig. 3. It can be seen that even with fewer feature dimensions, the tensor sparse coding outperforms the KNN classifier significantly. The kernel SVM performs very poorly ($< 30\%$ accuracy) on this dataset, and hence is not shown.

Texture Classification. We now use the region covariances for texture classification, on the Brodatz dataset [32]. We use the training images in the database used to construct the 5-texture (*'5c'*, *'5m'*, *'5v'*, *'5v2'*, *'5v3'*), 10-texture (*'10'*, *'10v'*) and 16-texture (*'16c'*, *'16v'*) mosaics. From each image, 32×32 blocks are cut out, and a 5×5 region covariance descriptor is computed for each block using the grayscale intensities and absolute values of the first- and second-order spatial derivatives, $\{I, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|\}$.

Each image is 256×256 pixels, yielding 64 data points per image. For a k-class problem, we get $64k$ data points, where $k = 5$, 10, or 16. In each case,

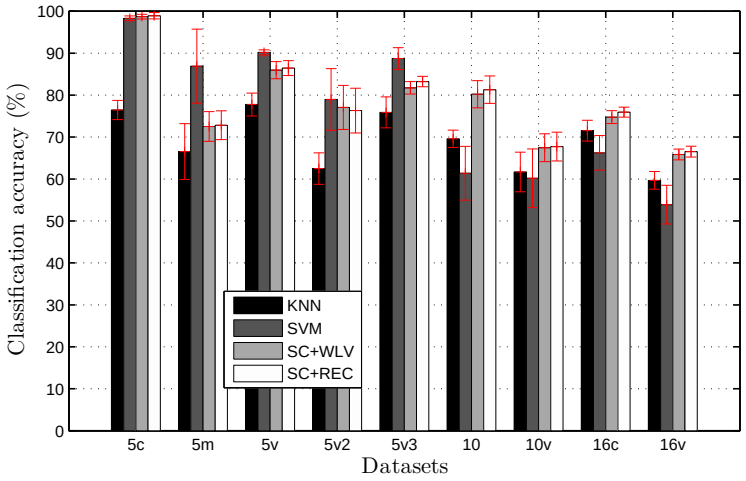

Fig. 4. Texture classification results on the Brodatz dataset, consisting of five 5-class, two 10-class and two 16-class problems. The results are averaged over 20 trials, and 1σ standard deviation bars are also shown.

5 data points from each class are used to construct the dictionary $\mathcal{A}$, $|\mathcal{A}| = 5k$, and the remaining $59k$ points are used for testing. The classification results are averaged over 20 random train-test splits, and are shown in Fig. 4. The sparse-coding-based methods consistently beat the KNN classifier, and is competitive with the SVM classifier. In fact, as number of classes increases, the sparse coding methods overtake the SVM classifier.

5 Conclusions and Future Work

We have proposed a novel sparse coding technique for positive definite matrices, which is convex and belongs to the standard class of MAXDET optimization problems. The performance of the tensor sparse coding in terms of accuracy of reconstruction, sparsity of the decomposition, as well as variations for different input parameters is analyzed. Results are shown not only for synthetic data but also for data sets from real-world computer vision applications, demonstrating the suitability of our model. In classification performance, the algorithms based on tensor sparse coding beat the state-of-the-art methods by a reasonable margin.

This work opens the door for the many sparsity-related algorithms to the space of positive definite matrices, and many techniques that require only a sparse coding step follow through readily from our work. Future work involves applying the above techniques to areas such as Diffusion Tensor Imaging. We are currently working on developing dictionary learning techniques over the positive definite matrix data, so that we may also learn a suitable dictionary in a data-driven manner, depending on the application at hand.

Acknowledgments. We are thankful to Professor Zhi-Quan (Tom) Luo, Ajay Joshi and Anoop Cherian for their thoughtful input. 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 #CNS-0324864, #CNS-0420836, #IIP-0443945, #IIP-0726109, #CNS-0708344, #CNS-0821474, #IIP-0934327, #IIS-0534286, and #IIS-0916750.

References

1. Vandenberghe, L., Boyd, S., Wu, S.: Determinant Maximization with Linear Matrix Inequality Constraints. In: SIAM Journal on Matrix Analysis and Applications (19), pp. 499-533 (1998)
2. Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. Roy. Statist. Soc. Ser. B 58(1), pp. 267-288 (1996)
3. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least Angle Regression. In: Annals of Statistics 32(2), pp. 407-499 (2004)
4. Tropp, J., Gilbert, A.: Signal Recovery from Random Measurements via Orthogonal Matching Pursuit. In: IEEE Transactions on Information Theory 53(12), pp. 4655-4666 (2007)
5. Donoho, D.: Compressed Sensing. In IEEE Transactions on Information Theory 52(4), pp. 1289-1306 (2006)
6. Candès, E., Wakin, M.: An Introduction to Compressive Sampling. In: IEEE Signal Processing Magazine 25(2), pp. 21-30 (2008)
7. Wright, J., Yi M., Mairal, J., Sapiro, G., Huang, T.S., Yan, S.: Sparse Representation for Computer Vision and Pattern Recognition. In: Proceedings of the IEEE 98(6), pp. 1031-1044 (2010)
8. Hazan, T., Polak, S., Shashua, A.: Sparse Image Coding Using a 3D Non-negative Tensor Factorization. In: International Conference of Computer Vision, pp. 50-57 (2005)
9. Müller, K.R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to Kernel-based Learning Algorithms. In: IEEE Transactions on Neural Networks 12(2), pp. 181-201 (2001)
10. Tuzel, O., Porikli, F., Meer, P.: Region Covariance: A Fast Descriptor for Detection and Classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 697-704. Springer, Heidelberg (2006)
11. Porikli, F., Tuzel, O.: Fast Construction of Covariance Matrices for Arbitrary Size Image Windows. In: Proceedings of IEEE International Conference on Image Processing, pp. 1581-1584 (2006)
12. Tuzel, O., Porikli, F., Meer, P.: Pedestrian Detection via Classification on Riemannian Manifolds. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 30(10), pp. 1713-1727 (2008)
13. Pennec, X., Fillard, P., Ayache, N.: A Riemannian Framework for Tensor Computing. In: International Journal of Computer Vision 66(1), pp. 41-66 (2005)
14. Pang, Y., Yuan, Y., Li, X.: Gabor-Based Region Covariance Matrices for Face Recognition. In: IEEE Transactions on Circuit and Systems for Video Technology 18(7), pp. 989-993 (2008)
15. Hu, H., Qin, J., Lin, Y., Xu, Y.: Region Covariance based Probabilistic Tracking. In: 7th World Congress on Intelligent Control and Automation, pp. 575-580 (2008)

16. Palaio, H., Batista, J.: Multi-object Tracking using an Adaptive Transition Model Particle Filter with Region Covariance Data Association. In: 19th International Conference on Pattern Recognition, pp. 1-4 (2008)
17. Paisitkriangkrai, S., Shen, C., Zhang, J.: Fast Pedestrian Detection Using a Cascade of Boosted Covariance Features. In: IEEE Transactions on Circuits and Systems for Video Technology 18(8), pp. 1140-1151 (2008)
18. Sivalingam, R., Morellas, V., Boley, D., Papanikolopoulos, N.: Metric Learning for Semi-supervised Clustering of Region Covariance Descriptors. In: Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras, pp. 1-8 (2009)
19. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-Theoretic Metric Learning. In: Proceedings of International Conference on Machine Learning, pp. 209-216 (2007)
20. Kulis, B., Sustik, M., Dhillon, I.: Learning Low-Rank Kernel Matrices. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 505-512 (2006)
21. Banerjee, S., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman Divergences. In: Journal of Machine Learning Research 6, pp. 1705-1749 (2005)
22. Wishart, J.: The Generalised Product Moment Distribution in Samples from a Normal Multivariate Population. In: Biometrika 20A(1/2), pp. 32-52 (1928)
23. Wang, S., Jin, R.: An Information Geometry Approach for Distance Metric Learning. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, pp. 591-598 (2009)
24. Tropp, J.: Just Relax: Convex Programming Methods for Identifying Sparse Signals in Noise. In: IEEE Transactions on Information Theory 52(3), pp. 1030-1051 (2006)
25. Grant, M., Boyd, S.: CVX: Matlab Software for Disciplined Convex Programming, ver. 1.21 (2010), http://cvxr.com/cvx
26. Donoho, D., Tanner, J.: Sparse Nonnegative Solution of Underdetermined Linear Equations by Linear Programming. In: Proceedings of the National Academy of Sciences 102(27), pp. 9446-9451 (2005)
27. Donoho, D., Stodden, V.: When does Non-negative Matrix Factorization give a Correct Decomposition into Parts? In: Thrun, S., Saul, L., Sch´olkopf, B. (eds.) Advances in Neural Information Processing Systems 16, MIT Press, Cambridge, pp. 1141-1148 (2004)
28. Lee, D. D. and Seung, H. S.: Algorithms for Non-negative Matrix Factorization. In: Leen, T.K., Dietterich, T.G., Tresp, V. (eds.) Advances in Neural Information Processing Systems 13, MIT Press, Cambridge, pp. 556-562 (2000)
29. Chang, C.-C., Lin, C.-J.: LIBSVM: A Library for Support Vector Machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
30. Wright, J., Yang, A., Ganesh, A., Sastri, S.S., Ma, Y.: Robust Face Recognition via Sparse Representation. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 31(2), pp. 210-227 (2009)
31. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET Evaluation Methodology for Face-recognition Algorithms. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10), pp. 1090-1104 (2000)
32. Randen, T., Husoy, J.H.: Filtering for Texture Classification: A Comparative Study. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 21(4), pp. 291-310 (2009)