Tensor Sparse Coding for Positive Definite Matrices

Ravishankar Sivalingam, *Graduate Student Member, IEEE,* Daniel Boley, *Senior Member, IEEE,* Vassilios Morellas, *Member, IEEE,* and Nikolaos Papanikolopoulos, *Fellow, IEEE*

Abstract—In recent years, there has been extensive research on sparse representation of vector-valued signals. In the matrix case, the data points are merely vectorized and treated as vectors thereafter (for *e.g.*, image patches). However, this approach cannot be used for all matrices, as it may destroy the inherent structure of the data. Symmetric positive definite (SPD) matrices constitute one such class of signals, where their implicit structure of positive eigenvalues is lost upon vectorization. This paper proposes a novel sparse coding technique for positive definite matrices, which respects the structure of the Riemannian manifold and preserves the positivity of their eigenvalues, without resorting to vectorization. Synthetic and real-world computer vision experiments with region covariance descriptors demonstrate the need for and the applicability of the new sparse coding model. This work serves to bridge the gap between the sparse modeling paradigm and the space of positive definite matrices.

Index Terms—Sparse coding, positive definite matrices, region covariance descriptors, computer vision, optimization.

1 Introduction

In the past decade there has been extensive research on sparse representations of signals [1], [2], [3] and recovery of such sparse signals from noisy and/or under-sampled observations [4], [5]. 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 any of the available vector techniques. A review of the applications of sparse representation in computer vision and pattern recognition is presented in Wright *et al.* [6].

More recently some researchers have realized the advantages of maintaining the higher–order data in their original form [7] to preserve the inherent structure, which may be lost upon vectorization. One such data type consists of $n \times n$ symmetric positive semi-definite (SPSD) matrices ($\mathbb{S}_+^n$). A symmetric matrix A is positive semidefinite if, for any vector $\mathbf{x}$,

$$\mathbf{x}^T A \mathbf{x} \geq 0.$$

This is also succinctly denoted as $A \succeq 0$. This fundamental property arises from the implicit structure in the matrix A, *i.e.*, A has non-negative eigenvalues. By implicit, we mean that this necessary condition cannot be easily expressed in terms of the elements of A directly, unlike say, symmetry of a matrix. When the eigenvalues of a symmetric A are strictly positive, we call A a symmetric positive definite (SPD) matrix ($\mathbb{S}_{++}^n$), denoted by $A \succ 0$. Correspondingly, for any vector $\mathbf{x} \neq \mathbf{0}$, we have $\mathbf{x}^T A \mathbf{x} > 0$.

Positive definite matrices are a very natural generalization of positive scalars and positive vectors. They are used widely in probability and statistics, as well as to model certain physical phenomena. The covariance matrix of any (non-degenerate) multivariate distribution is a positive definite matrix. Kernel matrices from many popular machine learning algorithms [8] are positive semidefinite. In medical imaging, the revolutionary new field of Diffusion Tensor Imaging (DTI) represents each voxel in a 3-D brain scan by a 3×3 SPD 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 been introduced [9], [10], which represents an image region by the covariance of n-dimensional feature vectors at each pixel in that region. These feature descriptors are currently being used for human detection and tracking, object recognition, texture classification, query-based retrieval of image regions, etc. [11]. Unlike general vectors, SPD matrices do not form a Euclidean space when vectorized. Rather, they form a connected Riemannian manifold, and the distance between two points is measured using the geodesic connecting them on this manifold [12], [13], [14].

In this work we propose a novel algorithm for sparse representation of symmetric positive definite matrices called *tensor*[1] *sparse coding*. The sparse decomposition of a positive definite signal in terms of a given dictionary of positive definite *atoms* is formulated as a convex optimization problem. The proposed formulation belongs to the class of MAXDET optimization problems [15]

• *The authors are with the Department of Computer Science and Engineering, University of Minnesota, Twin Cities, 200 Union Street SE, Minneapolis, MN 55455. Email: {ravi,boley,morellas,npapas}@cs.umn.edu.*
This work is an extended version of an earlier publication: "Tensor Sparse Coding for Region Covariances" published in ECCV 2010.

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

which can be solved through efficient interior point (IP) algorithms. We believe that this extension of sparse coding techniques to the space of SPD matrices will benefit the development of sparse models tailored to the relevant problem domains. This forms the first step toward extending the vast toolbox of sparsity-based algorithms to this class of data points.

The rest of the paper is organized as follows: In Section 2, we provide a brief description about region covariance descriptors, since they form the primary motivation behind this work and are used in all of our experiments. Section 3 presents an overview of previous work on region covariances, especially those approaches that deal with the manifold geometry of these descriptors. Section 4 describes the sparse coding problem for SPD matrices, and our tensor sparse coding approach is explained in Section 5. In Section 6, synthetic experiments are presented to demonstrate the need for a direct tensor approach as opposed to those involving vectorization. Experiments on real-world data, *i.e.*, region covariances for human appearance modeling, texture classification and face recognition, are presented in Section 7 and show the applicability of sparse modeling by comparing with previous techniques on these positive definite descriptors. Section 8 illustrates the similarity between the geodesic distance on the Riemannian manifold and the objective used in our formulation. Conclusions and future research directions are presented in Section 9.

2 Region Covariance Descriptors

Region Covariance Descriptors (RCDs) were first introduced by Tuzel et al. [9] as a novel region descriptor for object detection and texture classification. Given an image $\mathcal{I}$, let ϕ define a function that extracts an n-dimensional feature vector $\mathbf{z}_i$ from each pixel $i \in \mathcal{I}$, such that

$$\phi(I, x_i, y_i) = \mathbf{z}_i,$$

where $\mathbf{z}_i \in \mathbb{R}^n$, 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 $\{\mathbf{z}_i\}_{i=1}^{|R|}$ of the pixels in region R. Thus the region covariance descriptor is given by

$$C_R = \frac{1}{|R|-1} \sum_{i=1}^{|R|} (\mathbf{z}_i - \mu_R)(\mathbf{z}_i - \mu_R)^T,$$

where, μ_R is the mean vector,

$$\mu_R = \frac{1}{|R|} \sum_{i=1}^{|R|} \mathbf{z}_i.$$

The feature vector $\mathbf{z}$ 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 $\mathbb{S}_{++}^n$. Given two covariance matrices A and B, the Riemannian distance metric $D_{\text{geo}}(A, B)$ gives the length of the geodesic connecting these two points on this manifold. This is given by [12],

$$D_{\text{geo}}(A, B) = \left\| \log\left(A^{-1/2} B A^{-1/2}\right) \right\|_F,$$

where $\log(\cdot)$ represents the matrix logarithm and $\|\cdot\|_F$ is the Frobenius norm.

The geodesic distance is affine-invariant, in that any non-singular congruence transformation on the covariances does not change the distance:

$$D_{\text{geo}}(XAX^T, XBX^T) = D_{\text{geo}}(A, B),$$

for any invertible X. This corresponds to a linear transformation of the feature vectors $\mathbf{z}_i \mapsto X\mathbf{z}_i$. Therefore, region covariances can be invariant to illumination, orientation and scale of the image region, depending on the features used and how the regions are defined. 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 [16].

Arsigny et al. [17] proposed another metric known as the Log-Euclidean distance, which is the distance between two positive definite matrices measured on the tangent space of $\mathbb{S}_{++}^n$ at the identity matrix. The tangent space of SPD matrices at any point on the manifold is $\mathbb{S}^n$, the space of $n \times n$ symmetric matrices, and the tangent operator is the matrix logarithm. If A is SPD, then $\log A$ is a symmetric matrix, with no constraints on its eigenvalues. The Log-Euclidean metric is given by

$$D_{\text{LE}}(A, B) = \|\log A - \log B\|_F.$$

This is a lower bound on the actual geodesic distance, and the bound is exact when the two matrices commute [18]. Some works in the literature use this metric due to its simplicity, and since the tangent space is Euclidean the symmetric matrices in this space can be vectorized for further processing. Other relevant metrics for positive definite matrices are also elaborated in [19], showing results from diffusion tensor imaging.

3 Related Work

As mentioned earlier, region covariances were first introduced in [9]. Porikli and Tuzel [20] describe a technique for fast construction of region covariances for rectangular image windows, using integral images, enabling the use of these compact features for many practical applications that demand real–time performance. Since then, they have been used for tracking [10], [21], texture classification and segmentation [22], [23], object detection [11], [24], [25], face recognition [26], and action

recognition [27]. In [28], Cargill *et al.* provide a performance evaluation of the covariance descriptor as a suitable feature for generic target detection.

In [10], the authors track non-rigid objects with an update mechanism based on Lie algebra defined at the tangent space of the identity matrix. [11] present a boosting framework over region covariances. Zheng *et al.* [29] apply a manifold learning method for tracking people with region covariances. Sivalingam *et al.* [16] describe a framework for metric learning over positive semi-definite matrices. Wang and Wu [30] perform object tracking using region covariances by incrementally learning a low-dimensional model for the covariances in an adaptive manner.

Porikli [31] provides a collective description of most of the different learning algorithms used above for region covariances. The most successful algorithms are those which respect the structure of the Riemannian manifold.

In machine learning, multiple kernel learning attempts to learn a convex or conic combination of pre-defined kernel matrices that optimizes certain objectives. These pre-determined kernels can be parametric kernels with different parameter choices. [32], [33] optimize a performance measure over the conic combination of the individual kernel matrices, without specifying an actual target kernel - they have a constraint on the trace of the target kernel only. [34] minimizes the Logdet divergence between a target kernel (the optimal kernel formed from the ground truth labels) and a convex combination of a set of pre-defined kernels. The combination weights are optimized using project gradient descent over the simplex, and there is no sparsity constraint suggested. Further, optimizing an ℓ_1 sparsity term or constraint is not feasible here since a convex combination is used (the weights are non-negative and sum to 1). In our work, we use a conic combination of positive definite (or semidefinite) matrices, with a sparsity constraint on the coefficient vector.

There has also been work on regression over positive semidefinite matrices where the response variable is a scalar, *i.e.*,

$$y = f(WX),$$

where $X \in \mathbf{R}^{n \times n}$, $W \in \mathbf{S}_+^n$, and $y \in \mathbf{R}$. A quadratic loss function over the response variables and their predictions is optimized. [35] uses the von Neumann divergence term as a regularizer for the optimization over the positive definite W. [36] uses Riemannian optimization over positive semidefinite W to learn a regression model. They also describe a connection between the Riemannian affine-invariant metric and the LogDet divergence. Nesterov and Todd [37] explore the connections between Riemannian geometry and self-concordant barrier functions used in interior-point methods.

In the area of metric learning, Davis *et al.* [38] learn a distance metric (or kernel matrix) based on pairwise constraints on the data samples, and optimize a Logdet divergence measure between a given matrix A_0 and the learned matrix A, $A, A_0 \in \mathbf{S}_+^n$. This is carried out to satisfy the pairwise constraints as much as possible, while staying close to the original matrix.

In our work, the goal is to represent a positive definite matrix by a linear (or conic) combination of a set of positive definite matrices, while trying to enforce sparsity on the coefficients. Some other works trying to learn similar sparse decompositions are given below: In [39], Guo *et al.* take the covariance descriptors to the tangent space, by the logarithm map and perform vector sparse coding in this Euclidean space. The resultant algorithm gives good performance for action recognition in video. Wang and Vemuri [40] also learn sparse representations over positive definite matrices in the tangent space, via the logarithm and exponential maps. In a similar approach, Sra and Cherian [41] learn a generalized dictionary of rank-1 positive semidefinite atoms to sparsely represent covariance descriptors. However, the authors in the above two approaches use the Frobenius norm as the error metric. Pfander *et al.* [42] decompose a general matrix as a sparse linear combination of a dictionary of matrices by multiplying all the involved matrices on a known vector reducing the matrix problem to a known vector problem with well-established guarantees. Wang *et al.* [43] present the *Common Component Analysis* problem, where the authors learn a common low-dimensional subspace for a set of high-dimensional covariance matrices.

We present a novel sparse coding approach that uses a distortion function derived from the Wishart probability distribution. This approach maintains the positive definite matrices as such without vectorization, and therefore is more respectful of the matrix manifold geometry than vectorizing the matrices and treating them as points in Euclidean space.

4 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 $A_i \in \mathbb{S}_{++}^n$ is referred to as a dictionary atom. Given a signal $S \in \mathbb{S}_{++}^n$, our goal is to represent S as a linear combination of the dictionary atoms, *i.e.*,

$$S = x_1 A_1 + x_2 A_2 + \ldots + x_K A_K = \sum_{i=1}^{K} x_i A_i, \qquad (1)$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_K)^T$ is the coefficient vector.

With a slight abuse of notation, we will henceforth represent the sum $\sum_{i=1}^K x_i A_i$ as $\mathcal{A}\mathbf{x}$ for the sake of convenience[2].

Since only a non-negative linear combination of positive definite matrices is guaranteed to yield a positive definite matrix, we impose the constraint $\mathbf{x} \geq 0$ on the coefficient vector. However, we will also explore the effect of removing this constraint in later sections.

2. This can be distinguished from the regular $A\mathbf{x}$ matrix-vector multiplication through the calligraphic notation of $\mathcal{A}$.

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. In other words, S need not be exactly sparse in the space of the dictionary $\mathcal{A}$. Hence, we will try to find the best approximation $\hat{S} = \mathcal{A}\mathbf{x}$ to S, by minimizing the residual approximation error.

$$S \approx \hat{S} = \mathcal{A}\mathbf{x}^*, \text{ where } \mathbf{x}^* = \arg\min_{\mathbf{x}} \; d\left(\mathcal{A}\mathbf{x}, S\right), \qquad (2)$$

and $d(\cdot,\cdot)$ is an appropriate distortion measure over positive definite matrices.

Since we are reconstructing a positive definite signal S, we also require the approximation $\hat{S}$ to be positive definite,

$$\hat{S} \succeq 0 \Longrightarrow x_1 A_1 + x_2 A_2 + \ldots + x_K A_K \succeq 0.$$

Although this would be ensured by construction due to the non-negativity of $\mathbf{x}$ and the strictly positive definite dictionary atoms, we nonetheless retain this constraint explicitly for reasons which will become clear shortly.

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 $\mathbf{x}$:

$$\|\mathbf{x}\|_0 \leq T,$$

where T is a predefined parameter denoting the maximum number of non-zero elements in $\mathbf{x}$.

Next we need to select the distortion measure in Equation (2). While the Riemannian geodesic distance (1) would be our first choice - however it is a non-convex function (consider $|\log x|$) and therefore difficult to optimize directly. Hence we search for another loss function to optimize. The Logdet divergence, as we will elaborate next, is a well-suited distortion measure, not only due to its significant relation with Wishart and Gaussian distributions, but also because it results in a well-known and tractable convex optimization problem.

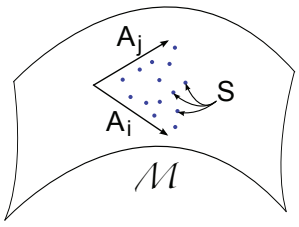


Fig. 1. Data points S on the manifold of positive definite matrices are to be represented by a linear combination of atoms A_i from the dictionary $\mathcal{A}$.

5 Approach

5.1 The Logdet Divergence

The Logdet divergence [44] $D_{\mathrm{ld}}\left(X,Y\right)$ is a Bregman divergence [45] between two matrices $X \in \mathbb{S}_+^n$ and $Y \in \mathbb{S}_{++}^n$, and is given by,

$$D_{\mathrm{ld}}(X,Y) = \mathrm{tr}\left(XY^{-1}\right) - \log\det\left(XY^{-1}\right) - n. \qquad (3)$$

It is asymmetric (and therefore, a divergence) $D_{\mathrm{ld}}\left(X,Y\right) \neq D_{\mathrm{ld}}\left(Y,X\right)$, and is convex only in the first argument. It is also known as Stein's loss in covariance estimation in statistics, or the Burg matrix divergence (a matrix generalization of the Burg divergence).

The Logdet divergence is equal to twice the *Kullback-Leibler* divergence (K-L divergence) between two multivariate Gaussians with equal mean [38]. Consider:

$$P_x = \mathcal{N}\left(\mu_x, \Sigma_x\right),$$
$$P_y = \mathcal{N}\left(\mu_y, \Sigma_y\right),$$

where $\mu_x, \mu_y \in \mathbb{R}^n$ and $\Sigma_x, \Sigma_y \in \mathbb{S}_{++}^n$. The K-L divergence between P_x and P_y is given by

$$D_{\mathrm{KL}}\left(P_x \| P_y\right) = \frac{1}{2}\Big(\mathrm{tr}\left(\Sigma_y^{-1}\Sigma_x\right) - \log\det\left(\Sigma_y^{-1}\Sigma_x\right) + \left(\mu_x - \mu_y\right)^T \Sigma_y^{-1}\left(\mu_x - \mu_y\right) - n\Big).$$

When $\mu_x = \mu_y$,

$$D_{\mathrm{KL}}\left(P_x \| P_y\right) = \frac{1}{2}\left(\mathrm{tr}\left(\Sigma_y^{-1}\Sigma_x\right) - \log\det\left(\Sigma_y^{-1}\Sigma_x\right) - n\right),$$
$$\therefore D_{\mathrm{KL}}\left(P_x \| P_y\right) = \frac{1}{2} D_{\mathrm{ld}}\left(\Sigma_x, \Sigma_y\right).$$

According to Banerjee et al. [46], there exists a bijection between regular exponential families and a large class of Bregman divergences known as regular Bregman divergences. 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 [47], which is a distribution over $n \times n$ positive definite matrices, with positive definite parameter matrix $\Theta \in \mathbb{S}_{++}^n$ and degrees of freedom $p \geq n$, is given by

$$\Pr(X|\Theta, p) = \frac{|X|^{(p-n-1)/2} \exp\left(-\frac{1}{2}\mathrm{tr}(\Theta^{-1}X)\right)}{2^{pn/2}|\Theta|^{p/2}\Gamma_n(p/2)}, \qquad (4)$$

where $|\cdot|$ is the determinant. The Logdet divergence $D_{\mathrm{ld}}(X,\Theta)$ is the Bregman divergence corresponding to the Wishart distribution $\Pr(X|\Theta,p)$ [48].

The Wishart distribution is also a conjugate prior for the inverse sample covariance matrix (or precision matrix) of a multivariate Gaussian distribution. Correspondingly, the inverse Wishart distribution is the conjugate prior for the sample covariance matrix. [49]. Since

$$D_{\mathrm{ld}}(X,Y) = D_{\mathrm{ld}}(Y^{-1}, X^{-1}),$$

the Bregman divergence for the inverse Wishart distribution $\Pr(X^{-1}|\Theta^{-1},p)$ is $D_{\mathrm{ld}}(\Theta^{-1}, X^{-1})$. Here Θ^{-1} refers to the true covariance of the multivariate Gaussian distribution and X^{-1} the sample covariance matrix.

In the sparse coding framework, if Σ^* is the true covariance, and S is the sample covariance signal provided, the goal is to estimate the true covariance as a sparse linear combination of certain basis atoms. Therefore, the Logdet divergence $D_{\mathrm{ld}}(\Sigma^*, S)$ appears to be a suitable candidate as the objective function for the sparse coding formulation.

Note that the Logdet divergence is also affine-invariant like the geodesic distance, in terms of its arguments:

$$D_{\mathrm{ld}}(XAX^T, XBX^T) = D_{\mathrm{ld}}(A, B),$$

for any invertible X.

In Section 8 we will also show a further similarity between the Riemannian geodesic distance (1) and the Logdet divergence (3).

5.2 Tensor Sparse Coding Formulation

Motivated by the aforementioned reasons, the optimization problem is defined as minimizing the Logdet divergence $D_{\mathrm{ld}}(\hat{S}, S)$ between the approximation $\hat{S}$ and the given matrix S.

$$D_{\mathrm{ld}}(\hat{S}, S) = \mathrm{tr}\left(S^{-1}\mathcal{A}\mathbf{x}\right) - \log\det\left(S^{-1}\mathcal{A}\mathbf{x}\right) - n. \quad (5)$$

In order to reduce the problem to a canonical form, and to improve numerical stability, we apply the invariant property of the trace and the $\log\det$ under similarity transformations. The objective function is unaffected by the similarity transformation $X \mapsto S^{1/2}XS^{-1/2}$, where X is the argument of the trace or $\log\det$.

$$\begin{aligned} D_{\mathrm{ld}}(\hat{S}, S) &= \mathrm{tr}\left(S^{-1/2}\left(\mathcal{A}\mathbf{x}\right)S^{-1/2}\right) \\ &\quad - \log\det\left(S^{-1/2}\left(\mathcal{A}\mathbf{x}\right)S^{-1/2}\right) - n \quad (6) \\ &= \mathrm{tr}\left(\hat{\mathcal{A}}\mathbf{x}\right) - \log\det\left(\hat{\mathcal{A}}\mathbf{x}\right) - n, \quad (7) \end{aligned}$$

where $\hat{\mathcal{A}} = \{\hat{A}_i\}_{i=1}^K$, and $\hat{A}_i = S^{-1/2}A_iS^{-1/2}$. Exploiting the linearity of the trace, setting $\mathbf{c} : c_i = \mathrm{tr}\hat{A}_i$, and discarding the constant n,

$$f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} - \log\det\left(\hat{\mathcal{A}}\mathbf{x}\right). \quad (8)$$

While the approaches in this paper use a given fixed dictionary, future work in this framework on learning the dictionary $\mathcal{A}$ from the data necessitates an added constraint that the residual $E = S - \hat{S}$ be positive semidefinite.

$$\hat{S} = \mathcal{A}\mathbf{x} \preceq S \quad \text{or} \quad \hat{\mathcal{A}}\mathbf{x} \preceq I_n, \quad (9)$$

where I_n is the $n \times n$ identity matrix. This constraint is useful scenarios where we learn the dictionary from data or augment the dictionary with new atoms. When this is not the case, we can relax this upper cone constraint. In the Section 7, we show results both from retaining this constraint (denoted by "2-cone") and relaxing it ("1-cone").

The ℓ_0 sparsity constraint in Equation (3) is non-convex and and therefore we replace this with its nearest convex relaxation - the ℓ_1 norm of $\mathbf{x}$:

$$\|\mathbf{x}\|_1 = \sum_{i=1}^K |x_i|.$$

Under certain assumptions [50], minimizing the ℓ_1 penalty has been proven to yield equivalent results as minimizing $\|\mathbf{x}\|_0$ for sparse vector decompositions. Hence it is appealing to perform the same relaxation here as well.

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

$$\begin{aligned} \min_{\mathbf{x}\geq 0} \quad & \mathbf{c}^T\mathbf{x} - \log\det\left(\hat{\mathcal{A}}\mathbf{x}\right) + \lambda\|\mathbf{x}\|_1 \quad (10) \\ \text{s.t.} \quad & 0 \preceq \hat{\mathcal{A}}\mathbf{x} \preceq I_n, \quad (11) \end{aligned}$$

where $\lambda \geq 0$ is a parameter which represents a trade–off between a sparser representation and a more accurate reconstruction. Since $x_i \geq 0$, the ℓ_1 norm simply becomes the sum of the components of $\mathbf{x}$:

$$\|\mathbf{x}\|_1 = \sum_{i=1}^K x_i, \quad (12)$$

yielding the optimization problem:

$$\begin{aligned} \min_{\mathbf{x}\geq 0} \quad & \hat{\mathbf{c}}^T\mathbf{x} - \log\det\left(\hat{\mathcal{A}}\mathbf{x}\right) \quad (13a) \\ \text{s.t.} \quad & 0 \preceq \hat{\mathcal{A}}\mathbf{x} \preceq I_n, \quad (13b) \end{aligned}$$

where $\hat{c}_i = c_i + \lambda$.

Concurrent with other vector sparse coding techniques, we may express this optimization problem in an alternate form which puts a hard constraint on the ℓ_1 norm of $\mathbf{x}$ instead of a penalty term $\lambda\|\mathbf{x}\|_1$ in the objective function.

$$\begin{aligned} \min_{\mathbf{x}\geq 0} \quad & \mathbf{c}^T\mathbf{x} - \log\det\left(\hat{\mathcal{A}}\mathbf{x}\right) \quad (14a) \\ \text{s.t.} \quad & \sum_{i=1}^K x_i \leq T \quad (14b) \\ & 0 \preceq \hat{\mathcal{A}}\mathbf{x} \preceq I_n, \quad (14c) \end{aligned}$$

We denote the optimization problems defined by Equations (13a–13b) and Equations (14a–14c) as Type I (ℓ_1-*regularized*) and Type II (ℓ_1-*constrained*) respectively. These two formulations are equivalent for appropriate choices of λ and T.

5.3 The MAXDET problem

The above formulations of tensor sparse coding fall under a general class of optimization problems known as determinant maximization (MAXDET) problems [15], of which semi-definite programming (SDP) and linear

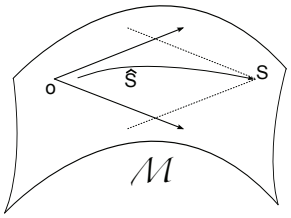


Fig. 2. The feasible set consists of the region of intersection of two positive semidefinite cones, one centered at the origin O, and the other an inverted cone centered at S. $\hat{S}$ is pushed towards S by the $\log\det$ term in the objective. The linear term serves as a regularizer on the coefficients x_i.

programming (LP) are special cases. The MAXDET problem [15] is defined as:

$$\min_{\mathbf{x}} \quad \mathbf{c}^T\mathbf{x} + \log\det G(\mathbf{x})^{-1} \tag{15a}$$

$$\text{s.t.} \quad G(\mathbf{x}) \triangleq G_0 + x_1G_1 + \ldots + x_KG_K \succ 0 \tag{15b}$$

$$F(\mathbf{x}) \triangleq F_0 + x_1F_1 + \ldots + x_KF_K \succeq 0, \tag{15c}$$

where $\mathbf{x} \in \mathbb{R}^K$, $G_i \in \mathbb{S}^n$ and $F_i \in \mathbb{S}^N$. The MAXDET problem is convex and can be solved by efficient interior point (IP) methods.

Note that the $G(\mathbf{x})$ 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.

The optimization problems in Type I and II are presented here in their canonical MAXDET form. Comparing to the optimization problem Type I, we have

$$c_i = \mathrm{tr}\hat{A}_i + \lambda, \qquad \text{for } i = 1, \ldots, K, \tag{16a}$$

$$G(\mathbf{x}) = \sum_{i=1}^{K} x_i\hat{A}_i \succ 0, \tag{16b}$$

$$F(\mathbf{x}) = \left[\begin{array}{c|c} \mathrm{diag}(\mathbf{x}) & 0 \\ \hline 0 & I_n - \sum_{i=1}^{K} x_i\hat{A}_i \end{array}\right] \succeq 0, \tag{16c}$$

with $N = K + n$.

Comparing to the optimization problem Type II, we have

$$c_i = \mathrm{tr}\hat{A}_i, \qquad \text{for } i = 1, \ldots, K, \tag{17a}$$

$$G(\mathbf{x}) = \sum_{i=1}^{K} x_i\hat{A}_i \succ 0, \tag{17b}$$

$$F(\mathbf{x}) = \left[\begin{array}{c|c|c} \mathrm{diag}(\mathbf{x}) & 0 & 0 \\ \hline 0 & T - \sum_{i=1}^{K} x_i & 0 \\ \hline 0 & 0 & I_n - \sum_{i=1}^{K} x_i\hat{A}_i \end{array}\right] \succeq 0, \tag{17c}$$

with $N = K + n + 1$.

As is evident in the canonical forms of Equations (16) and (17), there exists more structure in the problem than is given by the basic MAXDET formulation. While G_i's and F_i's are only required to be symmetric in the MAXDET optimization, here the G_i's ($i \neq 0$) are positive definite, F_0 is diagonal and positive semidefinite, and the F_i's ($i \neq 0$) are block-diagonal, having a positive semidefinite block and a strictly negative definite block.

When the upper cone constraint $\mathcal{A}\mathbf{x} \preceq S$ is relaxed, the problem dimension (N) in the MAXDET formulation is reduced, decreasing the time taken for the sparse coding routine.

Thus, we have formulated two variations of our tensor sparse coding problem (ℓ_1-regularized and ℓ_1-constrained), both of which are convex and have been expressed in the standard MAXDET form. The feasible set consists of the region of intersection of two positive semidefinite cones (see Figure 2), one centered at the origin O, and the other - an inverted cone centered at S. The approximation $\hat{S}$ lies in the strict interior of this closed convex set. The $-\log\det$ term in the objective pushes the approximation $\hat{S}$ toward S, motivating a better approximation. The linear term serves as a weighted regularizer on the coefficients x_i.

6 Synthetic Experiments

Our first set of experiments were run on a synthetic data set. The dictionary $\mathcal{A} = \{A_i\}_{k=1}^K$ is generated as follows: each positive definite dictionary atom is computed as $A_k = W_kW_k^T$, where $W_k \in \mathbf{R}^{n\times n}$ and each $W_k(i,j)$, $i,j = 1,\ldots,n$, is sampled i.i.d from $\mathcal{U}(0,1)$. For Sections 6.1 and 6.3, the sample point S to be sparse-coded is also generated in this manner. For Section 6.2, a known k-sparse vector $\mathbf{x}^* \in \mathbf{R}_+^K$ is first generated - the support of $\mathbf{x}^*$ is generated by selecting k of K locations uniformly at random without replacement, and the non-zero values in $\mathbf{x}^*$ are sampled i.i.d. from $\mathcal{U}(0,1)$. The true signal is constructed as $S^* = \mathcal{A}\mathbf{x}^*$, and the test signal S to be sparse-coded is obtained as the sample covariance from a set of N i.i.d. multivariate Gaussian samples from $\mathcal{N}(\mathbf{0}, S^*)$ (with $N = 10n^2$). The sample covariance matrix of a multivariate Gaussian distribution follows a Wishart distribution [47], 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 $\|\hat{\mathbf{x}}\|_1$ of the estimated coefficient vector $\hat{\mathbf{x}}$ and the minimum eigenvalue $\lambda_{\min}(S - \hat{S})$ of the residual $(S - \hat{S})$.

6.1 Effect of sparsity constraints

Figure 3 shows the effect of varying λ on the quality of reconstruction, under the Type I sparse coding 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 $\mathbf{x}^*$ for $\lambda = 0$, where it can be seen that even the unconstrained case results in a sparse solution vector. This is due to the constraint 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 [51], [52], [53].

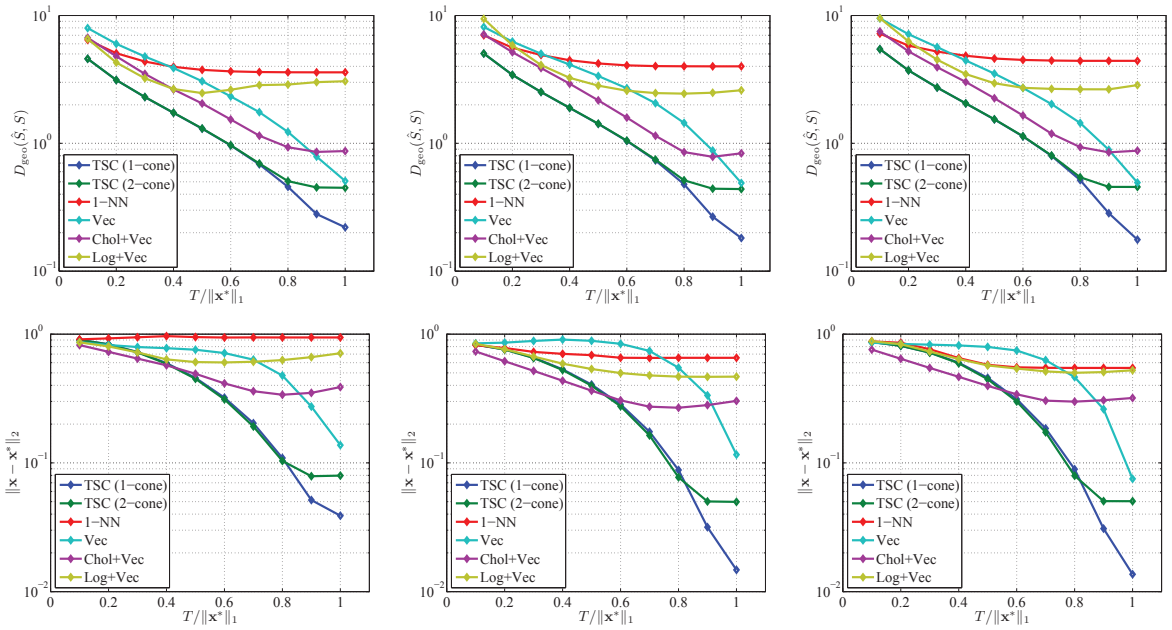


Fig. 4. Comparison of 1-NN, tensor and vector sparse coding - geodesic distance (upper row) and coefficient estimation error (lower row). The x-axis shows the normalized ℓ_1 constraint parameter $T/\|\mathbf{x}^*\|_1$, *i.e.*, the ℓ_1 'budget' is varied as a fraction of the ℓ_1 norm of the true solution $\mathbf{x}^*$. The problem sizes are $(n, K, k) = (5, 15, 3)$ for column 1, $(6, 18, 3)$ for column 2, and $(7, 28, 3)$ for column 3 (Best viewed in color).

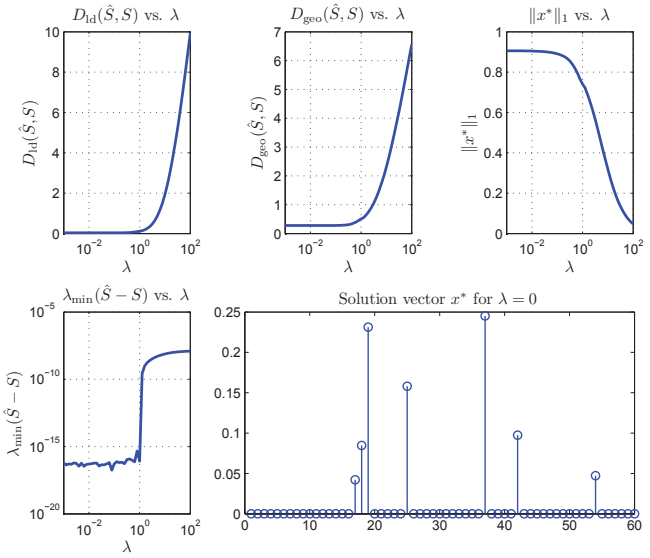


Fig. 3. Plot of the various quantities vs. λ for $n = 5$, $K = 60$. We show $D_{\text{ld}}(\hat{S}, S)$, $D_{\text{geo}}(\hat{S}, S)$, $\|\hat{\mathbf{x}}\|_1$, as well as $\lambda_{\min}(S - \hat{S})$, plotted in logarithmic scale. The λ values are varied logarithmically. The solution vector $\hat{\mathbf{x}}$ in the unconstrained case is also shown on the right, and is observed to be sparse even without explicitly enforcing any sparsity.

6.2 Comparison with Vector Sparse Coding

In order to clarify the need for a direct tensor sparse coding method, instead of vectorizing the SPD matrix and performing vector sparse coding, the advantages of the former over the latter must be demonstrated. The data is generated according to the procedure in the beginning of Section 6. Since we know the true $\mathbf{x}^*$ that generated the test signal S from the dictionary, we can consider the efficiency in recovering this true coefficient vector. The ℓ_1-constrained sparse coding technique is used, where the constraint T is varied as a fraction of the true required 'budget' $\|\mathbf{x}^*\|_1$, *i.e.*, $T \in [0, \|\mathbf{x}^*\|_1]$. We show results for cases where the constraint $\hat{S} \preceq S$ is retained ("2-cone") and relaxed ("1-cone"). For a baseline, we also show the performance of the 1-nearest-neighbor reconstruction (1-NN), where $\mathbf{x}^*$ is an all-zero vector except for a non-zero coefficient at the index corresponding to the nearest atom.

For the vector sparse coding case, we vectorize, for both the signal and the dictionary, and solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{x} \geq 0} \quad & \|\mathbf{s} - D\mathbf{x}\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{x}\|_1 \leq T, \end{aligned}$$

where $\mathbf{s} = \text{vecu}(S)$, $D = [\mathbf{a}_1 \dots \mathbf{a}_K]$ where $\mathbf{a}_i = \text{vecu}(A_i)$, and vecu is a function denoting the vectorization of the upper triangular part of the argument matrix. We retain the non-negativity constraint on the coefficients here as well for a fair comparison. The matrix reconstruction is then obtained as $\hat{S} = \text{vecu}^{-1}(\hat{\mathbf{s}})$ where $\hat{\mathbf{s}} = D\mathbf{x}$ and vecu^{-1} denotes the inverse of the upper triangular vectorization operation. This is repeated for matrix logarithms (since $\log : \mathbf{S}^n_{++} \mapsto \mathbf{S}^n$) and the Cholesky factors of the positive definite matrices.

We compare the geodesic distance between the reconstruction and the true covariance $D_{\text{geo}}(\hat{S}, S^*)$ as well as the error in the coefficient vector $\|\mathbf{x} - \mathbf{x}^*\|_2^2$ in the tensor and vector sparse coding approaches.

This is performed over 100 different coefficient vectors, given a fixed dictionary. The ℓ_1-constrained sparse coding is used for both the tensor and vector cases, and the constraint T is varied as a fraction of the true required 'budget' $\|\mathbf{x}^*\|_1$.

Figure 4 shows the comparison of geodesic distance between the reconstruction and the true covariance, for varying 'budget' constraints on the ℓ_1 norm of $\mathbf{x}$. Clearly the tensor sparse coding provides a more rigorous reconstruction in terms of the distance metric on the manifold. In fact even when the full ℓ_1 budget is provided, the vector case does not provide as good a reconstruction as the tensor algorithm that operates directly in the space of SPD matrices. The plot is shown in a log-scale to clearly show the gap between the two curves at $T = \|\mathbf{x}^*\|_1$. The "1-cone" and "2-cone" curves are alike up to a certain T, but after that the effect of the extra constraint in preventing a more closer approximation is visible.

From a sparse signal recovery viewpoint, we may compare the coefficient estimation error, also shown in Figure 4. In this case as well, the tensor sparse coding outperforms the vector method above a certain ℓ_1 constraint limit. The results are shown for three different problem sizes (n, K, k): $(5, 15, 3)$, $(6, 18, 3)$ and $(7, 28, 3)$.

This experiment validates the importance of being able to perform sparse coding of positive definite matrices directly without resorting to vectorization.

6.3 Effect of normalization

Given a set of signals $\mathcal{S} = \{S_j\}_{j=1}^N$, dictionary learning usually entails learning both the dictionary $\mathcal{A}$ as well as the sparse coefficients $\mathbf{x}_j, j = 1, \ldots, N$. However, the product $S = \mathcal{A}\mathbf{x}$ can only be determined up to a scaling factor, and one can arbitrarily scale up the "size" of the atoms in order to reduce the $\|\mathbf{x}\|_1$ term in the objective. Therefore, as is common in the vector dictionary learning literature, we attempt to normalize the dictionary atoms to have unit "size" in some sense. Three different normalization schemes were tested on the dictionary atoms:

1) by spectral norm, $\|A_i\|_2 = 1$,
2) by Frobenius norm, $\|A_i\|_F = 1$, and
3) by trace, $\mathrm{tr}(A_i) = 1$.

As we vary λ, we only get a proportional change in the four quantities mentioned above. This can be explained by the fact that all matrix norms are equivalent. Therefore, throughout the rest of this work, we adhere to normalization by trace: $\mathrm{tr}(A_i) = 1$.

7 Recognition 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 modeling, texture classification and face recognition.

The baseline comparison for classification is K-nearest-neighbor (KNN) approach using the geodesic distance (1), which we refer to as *geodesic KNN* (or geo-KNN). We also compare with a multi-class support vector machine (SVM) classifier with a radial basis kernel, computed as:

$$K(C_i, C_j) = \exp\left(-\frac{D^2_{\mathrm{geo}}(C_i, C_j)}{2\sigma^2}\right),$$

with bandwidth σ. This is referred to as *geodesic SVM* (or geo-SVM). Both of the above approaches operate directly on the region covariances for classification. The parameters K and σ for the baseline approaches were chosen based on cross-validation.

Much of the relevant literature on region covariances uses geodesic KNN for classification. Further, SVMs are powerful classifiers and very popular in computer vision applications. These reasons motivated our choice of the two algorithms to compare our results.

7.1 Human Appearance Descriptors

In this section, we present experiments on classification of human appearances, based on region covariance features. We use a subset of the 18-class *Cam5* dataset from [16], from which we choose the 16 classes which contain at least 10 data points each. The dataset contains a total of 407 images from these 16 classes. Representative images from the dataset are shown in Figure 5. 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. From each of the 16 classes, we select 5 points for training and the remaining are used for testing.

Our sparse coding method is used for classification as follows: The training data from each class forms a dictionary $\mathcal{A}_m$, $m = 1, \ldots, M$, where M is the number of classes ($M = 16$ here). The class dictionaries are concatenated into one large dictionary $\mathcal{A}$:

$$\mathcal{A} = [\ \mathcal{A}_1 \mid \mathcal{A}_2 \mid \ \ldots \ \mid \mathcal{A}_M\ ].$$

The test signal S is sparse coded over this *combined* dictionary, to yield a sparse coefficient vector $\mathbf{x}$. This vector consists of the coefficients corresponding to atoms from different classes $1, \ldots, M$, and can be written as

$$\mathbf{x} = \left[\ \mathbf{x}_1^T \mid \mathbf{x}_2^T \mid \ \ldots \ \mid \mathbf{x}_M^T\ \right]^T.$$

The class-wise reconstruction $\hat{S}_m$ is then obtained as $\hat{S}_m = \mathcal{A}_m \mathbf{x}_m$, and the class-wise reconstruction error is computed as $E_m = D_{\mathrm{ld}}\left(\hat{S}_m, S\right)$. The label m^* of the dictionary offering the minimum reconstruction error is then assigned to the test signal S.

$$m^* = \arg\min_m \ \ D_{\mathrm{ld}}\left(\hat{S}_m, S\right).$$

This approach is adapted from [54], and we refer to this as the *combined dictionary approach*.

We apply this combined dictionary approach to the problem of classifying human appearances, forming a dictionary $\mathcal{A}$ of $K = 80$ atoms. For this experiment, in addition to the reconstruction error-based classification (REC), we also compute a weighted label vote (WLV) for each class from the corresponding coefficient values, and use this as a score for classification:

$$m^* = \arg\max_m \ \|\mathbf{x}_m\|_1.$$

Fig. 5. Representative images from the *Cam5* dataset.

Classifier	Accuracy (%)
Geo-KNN ($K = 5$)	66.95 (4.89)
Geo-SVM ($\sigma = 0.5$)	77.64 (5.96)
VSC + WLV (Vec)	62.00 (3.89)
VSC + REC (Vec)	62.16 (3.67)
VSC + WLV (Chol)	73.53 (2.98)
VSC + REC (Chol)	76.40 (2.84)
TSC + WLV	**78.62** (1.49)
TSC + REC	77.85 (2.50)

TABLE 1

Mean classification accuracy for the *Cam5* dataset. Results are averaged over 100 trials and standard deviation values are also shown in parentheses.

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. We also show the REC and WLV classification accuracies with the vectorized upper-triangular parts of the covariances. This is obtained using traditional vector sparse coding (VSC), *i.e.*, the Lasso problem of [1]. In addition, the vectorized upper-triangular part of the Cholesky factor of each positive definite matrix descriptor is also used in the vector sparse coding framework for both REC and WLV classification. These results are also included in Table 1.

The tensor sparse coding approaches for appearance recognition outperform the KNN and SVM baseline algorithms, and also the vector sparse coding-based approaches. This demonstrates that sparse coding techniques that retain the positive definiteness of the data points yield better results not only with synthetic data but also in practical computer vision applications.

7.2 Face Recognition

In this section, we present experimental results for face recognition from grayscale images. This is performed over a subset of the FERET face database [55], consisting of grayscale images of 10 subjects, where each individual represents a separate class. The frontal or near-frontal images corresponding to the two-letter codes *'ba'*, *'bd'*, *'be'*, *'bf'*, *'bg'*, *'bj'*, and *'bk'* are used for our experiments, leading to a total of 70 face images. We extract Gabor-based region covariances from each face image following the approach of Pang et al. [26].

We crop the images based on the eye positions, and resize them to be of size 60×60 pixels. The Gabor filters [26] corresponding to 8 orientations ($u = 0, \ldots, 7$) and 5 scales ($v = 0, \ldots, 4$) are applied to each image, resulting in 40 different filter responses g_{uv}. In addition, we also test on features such as (x, y) spatial location of pixels in the image, image intensity I, derivatives of image intensity I_x, I_y, I_{xx}, I_{yy} and gradient orientation $\arctan I_y/I_x$. The different sets of features used in the covariance descriptor construction are described in Table 2.

Mode	Feature Set
1	$[\ x\ y\ I\ \lvert I_x\rvert\ \lvert I_y\rvert\ \lvert I_{xx}\rvert\ \lvert I_{yy}\rvert\ ]$
2	$\left[\ x\ y\ \lvert I_x\rvert\ \lvert I_y\rvert\ \lvert I_{xx}\rvert\ \lvert I_{yy}\rvert\ \arctan\frac{\lvert I_y\rvert}{\lvert I_x\rvert}\ \right]$
3	$[\ x\ y\ \lvert I_x\rvert\ \lvert I_y\rvert\ \lvert I_{xx}\rvert\ \lvert I_{yy}\rvert\ ]$
4	$\left[\ x\ y\ I\ \lvert I_x\rvert\ \lvert I_y\rvert\ \lvert I_{xx}\rvert\ \lvert I_{yy}\rvert\ \arctan\frac{\lvert I_y\rvert}{\lvert I_x\rvert}\ \right]$
5	$[\ x\ y\ g_{00}\ g_{01}\ \cdots\ g_{7v_{\max}}\ ]$
6	$[\ x\ y\ I\ g_{00}\ g_{01}\ \cdots\ g_{7v_{\max}}\ ]$
7	$[\ g_{00}\ g_{01}\ \cdots\ g_{7v_{\max}}\ ]$

TABLE 2

Features used in construction of region covariances for face recognition on the FERET face dataset. Feature sets 5–7 consist of 5 subsets each (a)–(e), where the number of octaves is varied from $v_{\max} = 0, \ldots, 4$.

We compute the region covariance descriptor over the entire face only, and not subsections of each face image as was done in [26]. At each iteration of the experiment, 4 out of 7 images from each subject are taken for training, and the remaining 3 are used as test images, yielding a total of $\binom{7}{3} = 35$ different train-test splits.

The face recognition is performed using the reconstruction error-based approach. In addition to the combined dictionary approach explained before, we also classify the signal by sparse coding it with each class dictionary $\mathcal{A}_m$ independently to obtain the coefficient vector $\mathbf{x}_m$, and predicting the label m^* as:

$$m^* = \arg\min_m \ D_{\mathrm{ld}}\left(\hat{S}_m, S\right).$$

.

TABLE 3
Mean classification accuracy for the *FERET* face recognition dataset. Results are averaged are over 35 trials, and standard deviations are provided in parenthesis.

Mode (n)	Covariances				Precisions				Geo-KNN (%)	Geo-SVM (%)
	Separate (%)		Combined (%)		Separate (%)		Combined (%)			
	1-cone	2-cone	1-cone	2-cone	1-cone	2-cone	1-cone	2-cone	$K = 1$	$\sigma = 20.0$
1 (7)	**85.81** (9.57)	**85.81** (10.40)	83.24 (10.40)	79.14 (11.14)	76.48 (10.26)	76.67 (11.63)	40.86 (6.14)	43.81 (5.17)	77.62 (9.55)	66.95 (7.23)
2 (7)	69.24 (12.75)	64.76 (11.93)	**71.33** (11.82)	64.19 (13.20)	53.71 (10.59)	54.95 (11.94)	20.95 (7.28)	24.76 (6.49)	62.67 (9.62)	49.62 (8.08)
3 (6)	65.24 (11.96)	64.48 (14.14)	**71.43** (13.12)	65.33 (13.62)	53.24 (10.00)	52.57 (10.78)	16.48 (6.90)	17.43 (6.81)	61.33 (10.76)	49.33 (7.51)
4 (8)	**86.76** (9.27)	84.19 (10.55)	84.76 (10.46)	76.95 (9.87)	77.33 (10.47)	79.05 (11.00)	44.10 (3.57)	48.57 (4.67)	78.48 (10.28)	67.71 (7.84)
5a (10)	**83.52** (10.69)	73.05 (11.69)	**83.52** (12.39)	75.62 (11.43)	38.67 (8.33)	38.29 (9.74)	18.38 (6.87)	18.48 (6.96)	79.62 (12.47)	70.57 (8.38)
5b (18)	93.24 (4.81)	80.00 (8.69)	**94.10** (4.79)	79.81 (8.35)	47.43 (10.14)	53.43 (9.68)	20.19 (7.47)	23.71 (6.51)	86.10 (7.83)	83.62 (7.62)
5c (26)	**93.81** (4.79)	76.19 (7.35)	91.43 (4.80)	72.95 (6.98)	72.57 (10.51)	71.81 (11.91)	50.38 (9.94)	56.10 (8.49)	90.57 (6.78)	88.86 (6.76)
5d (34)	**95.81** (3.77)	74.29 (9.55)	92.48 (4.80)	67.52 (9.41)	81.52 (10.15)	67.14 (10.99)	58.38 (8.52)	63.52 (9.39)	91.81 (6.34)	91.71 (6.29)
5e (42)	94.76 (5.36)	70.00 (10.54)	90.10 (7.01)	64.10 (8.73)	91.62 (6.49)	69.52 (9.92)	76.29 (8.57)	63.62 (10.37)	92.48 (5.54)	**94.95** (4.67)
6a (11)	89.24 (7.81)	80.95 (11.73)	**89.33** (7.92)	81.33 (10.87)	48.76 (10.24)	48.67 (10.05)	20.10 (9.61)	20.19 (9.79)	85.81 (9.10)	79.14 (7.57)
6b (19)	94.10 (4.79)	83.90 (8.22)	**95.33** (4.52)	83.05 (7.53)	54.38 (12.03)	61.81 (8.45)	22.00 (7.61)	27.90 (6.62)	89.81 (5.91)	88.19 (7.53)
6c (27)	**95.62** (3.63)	79.43 (6.64)	92.86 (4.00)	76.00 (8.20)	74.48 (10.98)	76.10 (12.28)	51.43 (9.80)	58.86 (10.30)	93.14 (6.07)	91.14 (6.37)
6d (35)	**96.48** (3.73)	75.14 (9.13)	94.29 (4.33)	70.38 (8.76)	84.19 (8.99)	68.67 (11.77)	61.43 (8.02)	65.33 (9.77)	92.76 (6.09)	92.57 (5.41)
6e (43)	**95.52** (4.91)	71.05 (11.35)	91.24 (6.67)	65.24 (10.09)	93.24 (5.77)	70.10 (9.14)	78.10 (7.86)	64.57 (10.11)	92.76 (5.49)	**95.52** (4.29)
7a (8)	78.76 (9.30)	73.24 (11.03)	**79.52** (9.96)	73.24 (9.81)	38.86 (9.01)	39.24 (8.84)	25.71 (6.20)	25.81 (6.14)	70.95 (12.74)	63.43 (10.88)
7b (16)	**92.19** (5.80)	77.81 (8.24)	91.71 (5.93)	77.52 (7.74)	46.29 (10.26)	50.10 (7.99)	20.67 (6.70)	21.90 (5.82)	83.62 (9.84)	84.29 (7.02)
7c (24)	**92.10** (5.86)	75.43 (5.35)	87.14 (5.92)	71.90 (8.41)	69.62 (9.01)	65.71 (11.42)	48.86 (9.15)	53.14 (8.35)	86.10 (7.62)	86.29 (7.47)
7d (32)	**93.05** (4.94)	72.29 (9.49)	90.48 (5.86)	65.24 (9.16)	78.67 (8.88)	62.29 (11.15)	53.05 (9.51)	57.81 (8.76)	89.14 (7.14)	89.43 (6.50)
7e (40)	**93.05** (5.77)	68.86 (11.27)	88.29 (6.34)	61.90 (12.35)	84.95 (7.36)	68.86 (9.01)	72.67 (8.00)	60.67 (9.75)	89.71 (6.44)	92.95 (5.39)
Mean	**88.86** %	75.31 %	87.50 %	72.18 %	66.63 %	61.84 %	42.11 %	42.96 %	**83.92** %	**80.33** %

We refer to this method as the *separate dictionary approach*.

The dictionaries are composed of the covariance descriptors from the training images. This is compared to the recognition performance using geodesic KNN and geodesic SVM.

Since the inverse of a positive definite matrix is also positive definite, we repeat the same experiment with the inverse covariances (or precision matrices). Since the geodesic distance between two matrices A and B is identical to that between A^{-1} and B^{-1},

$$D_{\text{geo}}(A, B) = D_{\text{geo}}\left(A^{-1}, B^{-1}\right),$$

the KNN and SVM classifiers do not differ in performance between covariance and precision matrices.

Further, we show the recognition performance when the upper cone constraint is relaxed ("1-cone") and compare it to the case where it is retained ("2-cone").

The mean classification accuracy over 35 trials is presented in Table 3 for each covariance feature mode. The best performance is obtained when using feature set $6d$ - the (x, y) location, image intensity, and 4 octaves of Gabor filter responses. Approaches based on vector sparse coding show an inferior performance compared to those based on tensor sparse coding, and are omitted here due to lack of space.

7.3 Texture Classification

In this section we present experimental results on texture classification with the Brodatz dataset [56]. We use the training images from the dataset which form the five 5-class, two 10-class, two 16-class, and three 2-class texture mosaics. Each texture class corresponds to one training image of 256×256 pixels, which is broken down into non-overlapping blocks of 32×32 pixels. A 5×5 covariance descriptor is then computed from each of these blocks, using the grayscale intensities and absolute values of the first- and second-order spatial derivatives, $\{I, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|\}$.

There are 64 covariance descriptors from each texture class, of which 8 descriptors from each class are chosen for training, and the remaining are used for testing. The classification results are averaged over 20 random train-test splits, and are shown in Table 4.

Similar to the previous section, we also repeat the same experiments with the inverse covariances descriptors, and by relaxing the extra cone constraint. The best sparse coding-based approach performs competitively with the baseline KNN and SVM approaches.

Note that the KNN and SVM approaches have had their respective parameters optimized for best performance through cross-validation. Their accuracy varies

TABLE 4
Mean classification accuracy for the *Brodatz* mosaic dataset. Results are averaged are over 20 trials, and standard deviations are provided in parenthesis.

Mode (n)	Covariances				Precisions				Geo-KNN (%)	Geo-SVM (%)
	Separate (%)		Combined (%)		Separate (%)		Combined (%)			
	1-cone	2-cone	1-cone	2-cone	1-cone	2-cone	1-cone	2-cone	$K=1$	$\sigma=0.6$
1 (5)	99.43 (0.63)	99.00 (0.62)	99.29 (0.41)	98.79 (0.74)	99.41 (0.62)	**99.45** (0.47)	99.18 (0.41)	98.84 (0.54)	98.88 (0.63)	99.14 (0.72)
2 (5)	93.13 (2.75)	91.66 (2.97)	86.09 (2.36)	84.89 (2.31)	**93.20** (2.61)	92.32 (2.87)	87.98 (2.47)	86.86 (2.50)	92.00 (2.27)	91.04 (2.31)
3 (5)	89.25 (2.47)	87.95 (2.55)	81.93 (2.59)	80.00 (3.04)	**89.32** (2.20)	87.86 (2.89)	82.54 (2.87)	82.11 (2.61)	87.21 (2.19)	88.79 (2.19)
4 (5)	85.36 (3.28)	84.05 (3.14)	83.41 (2.59)	82.05 (2.55)	85.64 (2.84)	84.05 (2.82)	83.66 (1.97)	82.39 (2.57)	92.55 (1.47)	**94.79** (1.38)
5 (5)	86.52 (1.83)	84.21 (2.48)	76.91 (3.26)	74.39 (3.38)	87.02 (1.50)	86.93 (1.99)	75.34 (2.51)	73.89 (2.61)	92.84 (1.48)	**94.55** (0.98)
6 (16)	**85.59** (1.02)	84.19 (0.84)	80.02 (1.15)	78.90 (1.05)	85.56 (1.08)	84.50 (1.36)	79.47 (0.97)	78.33 (1.11)	83.91 (0.98)	82.04 (1.98)
7 (16)	78.95 (1.52)	76.57 (1.30)	70.11 (0.99)	68.47 (1.35)	79.15 (1.35)	77.58 (1.67)	71.73 (1.32)	70.08 (1.47)	76.57 (1.34)	**80.18** (1.07)
8 (10)	87.71 (1.65)	86.13 (2.15)	84.81 (2.20)	83.79 (2.03)	87.48 (1.48)	86.59 (1.77)	84.40 (2.04)	83.46 (2.06)	**87.84** (1.48)	86.83 (3.94)
9 (10)	80.19 (1.88)	78.26 (1.69)	71.63 (1.84)	70.29 (2.84)	81.06 (1.83)	79.78 (1.97)	71.80 (2.19)	71.50 (2.15)	80.45 (2.08)	**82.21** (4.01)
10 (2)	99.87 (0.32)	99.87 (0.32)	99.91 (0.27)	99.82 (0.36)	99.96 (0.19)	**100.00** (0.00)	99.87 (0.32)	99.78 (0.56)	99.15 (0.82)	99.82 (0.36)
11 (2)	99.20 (1.23)	98.84 (1.41)	98.79 (1.17)	97.99 (1.46)	99.42 (1.07)	99.33 (1.26)	98.53 (1.50)	98.93 (1.40)	99.82 (0.36)	**100.00** (0.00)
12 (2)	98.30 (1.49)	96.43 (2.02)	96.34 (2.49)	94.33 (2.51)	98.62 (1.48)	99.06 (0.96)	98.13 (1.54)	98.79 (1.30)	**100.00** (0.00)	**100.00** (0.00)
Mean	90.29 %	88.93 %	85.77 %	84.48 %	**90.49** %	89.79 %	86.05 %	85.41 %	**90.94** %	**91.62** %

quite drastically for different parameter choices. On the other hand, our method's classification performance does not vary substantially with λ. In fact, for a wide variation in the values of λ, the final classification performance does not change drastically (although the individual coefficients of sparse coding do). While increasing λ results in a poorer reconstruction $\hat{S}$, we are comparing the effect of different class dictionaries - the quality of approximation is decreased ($D_{\mathrm{ld}}(\hat{S}_m, S)$ increases) for all classes $m = 1, \ldots, M$, leading to similar classification accuracies. This shows a certain robustness in our method with respect to the choice of parameter. Figure 6 shows how the accuracy varies with parameter choice for our method against the geodesic SVM for texture 12.

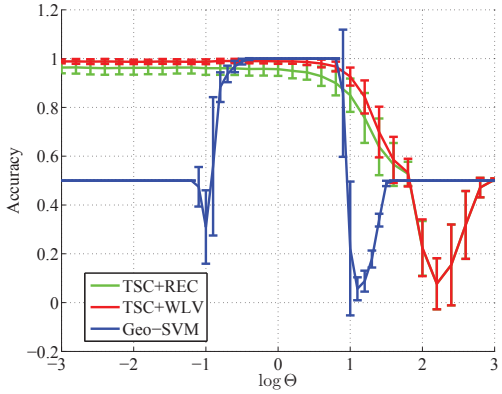


Fig. 6. Variation in classification accuracy (texture 12) with parameter choice for tensor sparse coding and SVM approaches. The parameter $\log_{10}\Theta$ is varied along the x-axis, and $\Theta = \lambda$ for the tensor sparse coding approach and $\Theta = \sigma$ for the geodesic SVM classifier. The former approach shows a largely consistent high performance. 1σ standard deviation bars are also shown (Best viewed in color).

8 Relation between D_{geo} and D_{ld}

In this section we derive an interesting connection between the Riemannian geodesic distance and the Logdet divergence.

Let $\lambda \sim \lambda(A, B)$ be the generalized eigenvalues of (A, B). The Riemannian geodesic distance between A and B is given by

$$D_{\mathrm{geo}}(A,B) = \left\| \log\left(B^{-1/2} A B^{-1/2}\right) \right\|_F .$$

In terms of the generalized eigenvalues, the geodesic distance

$$D_{\mathrm{geo}}(A,B) = \|\log\lambda\|_2 = \left\| \log\left(\frac{1}{\lambda}\right) \right\|_2 .$$

The general form of a Bregman divergence for matrix arguments is given by [57]

$$D_\varphi(X,Y) = \varphi(X) - \varphi(Y) - \langle \nabla\varphi(Y), (X - Y) \rangle,$$

where $\varphi(\cdot)$ is a strictly convex function over a convex set $\mathcal{S}$, and is differentiable in $\mathrm{relint}(\mathcal{S})$ (relative interior). The last term denotes the matrix inner product $\langle A, B \rangle = \mathrm{tr}(AB^T)$.

The Logdet divergence is derived from $\varphi(X) = -\log\det X$ and is given by:

$$\begin{aligned} D_{\mathrm{ld}}(A,B) &= \log\det A^{-1} - \log\det B^{-1} - \langle -B^{-1}, A - B \rangle \\ &\qquad \text{since } \nabla(-\log\det X) = -X^{-1} \\ &= -\log\det\left(B^{-1}A\right) + \mathrm{tr}\left(B^{-1}A - B^{-1}B\right). \end{aligned}$$

$$\therefore D_{\mathrm{ld}}(A,B) = \mathrm{tr}\left(B^{-1}A\right) - \log\det\left(B^{-1}A\right) - n. \quad (18)$$

The second term in the above equation can be written in terms of λ as:

$$-\log\det\left(B^{-1}A\right) = \operatorname{tr}\left(\log\left(B^{-1}A\right)^{-1}\right) = \sum_{i=1}^{n}\log\left(\frac{1}{\lambda_i}\right).$$

In our sparse coding formulation, we require that the approximation $\hat{S} \preceq S$, the original signal. If $B = S$ and $A = \hat{S}$, then $A \preceq B$, or $B^{-1}A \preceq I_n$. Therefore, for $i = 1, \ldots, n$,

$$\lambda_i \leq 1 \Longrightarrow \frac{1}{\lambda_i} \geq 1 \Longrightarrow \log\left(\frac{1}{\lambda_i}\right) \geq 0.$$

Since the elements in the sum are all non-negative,

$$\begin{aligned}-\log\det\left(AB^{-1}\right) &= \sum_{i=1}^{n}\log\left(\frac{1}{\lambda_i}\right) = \sum_{i=1}^{n}\left|\log\left(\frac{1}{\lambda_i}\right)\right| \\ &= \left\|\log\left(\frac{1}{\lambda}\right)\right\|_1.\end{aligned}$$

Plugging back into Equation (18), we have

$$D_{\mathrm{ld}}(A,B) = \left\|\log\left(\frac{1}{\lambda}\right)\right\|_1 + \langle B^{-1}, A-B\rangle, \qquad (19)$$

which is a combination of

1) an ℓ_1-norm term of reciprocal generalized eigenvalues of (A, B), denoted by $D_{\mathrm{L1}}(A, B)$), and
2) the component of the difference between A and B in the direction of the tangent of $\varphi(\cdot) = -\log\det(\cdot)$ evaluated at B.

When λ is very close to 1, or $|1-\lambda| \ll 1$, setting $x = \lambda - 1$ and using the Taylor's approximation $\log(1+x) \approx x$ when $|x| \ll 1$, the geodesic distance can be rewritten as follows:

$$\begin{aligned}D_{\mathrm{geo}}(A,B) &= \|\log(\lambda)\|_2 \approx \|\lambda - 1\|_2 \\ &= \left\|B^{-1}A - I_n\right\|_F \\ &= \left\|B^{-1}(A-B)\right\|_F \\ D^2_{\mathrm{geo}}(A,B) &\approx \operatorname{tr}\left\{\left(B^{-1}(A-B)\right)^2\right\} \quad \text{when } \lambda \approx 1.\end{aligned}$$

Similarly, rewriting the second term in Equation (19), we get

$$D_{\mathrm{ld}}(A,B) = \left\|\log\left(\frac{1}{\lambda}\right)\right\|_1 + \operatorname{tr}\left\{B^{-1}(A-B)\right\}.$$

It is interesting that the second term of the Logdet divergence forms a different ℓ_1-ℓ_2 type similarity with the approximate geodesic distance when $\lambda \approx 1$. Thus there is a two-fold connection between the Riemannian geodesic distance and the Logdet divergence.

Therefore, in our framework, specifically under the condition that $\hat{S} \preceq S$,

$$\begin{aligned}D_{\mathrm{geo}}(A,B) &= \left\|\log\left(\frac{1}{\lambda}\right)\right\|_2 \\ D_{\mathrm{L1}}(A,B) &= \left\|\log\left(\frac{1}{\lambda}\right)\right\|_1\end{aligned}$$

$$\begin{aligned}D_{\mathrm{ld}}(A,B) &= D_{\mathrm{L1}}(A,B) + \operatorname{tr}\left\{B^{-1}(A-B)\right\} \\ D^2_{\mathrm{geo}}(A,B) &\approx \operatorname{tr}\left\{\left(B^{-1}(A-B)\right)^2\right\} \quad \text{when } \lambda \approx 1.\end{aligned}$$

This clearly illustrates an analogy of the geodesic distance and the Logdet divergence to the ℓ_2 and ℓ_1 distances in more than one way.

This supports the use of the Logdet divergence in our model, and also intuitively explains the similarity in the trend of the geodesic distance and Logdet divergence across varying approximations in the sparse coding decompositions. Further, since the ℓ_1 norm tends to push most of the components to zero, the ℓ_1 term on the log-reciprocal generalized eigenvalues pushes most of the generalized eigenvalues to 1, thus giving us a closer approximation $\hat{S}$ to S, and a semidefinite residual $E = S - \hat{S}$.

The three dissimilarity measures can be compared for the simple case of 2×2 SPD matrices, as the eigenvalues (λ_1, λ_2) are varied in $[0, 1]$, the domain of our problem. In Figure 7, we show the slice of this surface at $\lambda_1 = \lambda_2$.

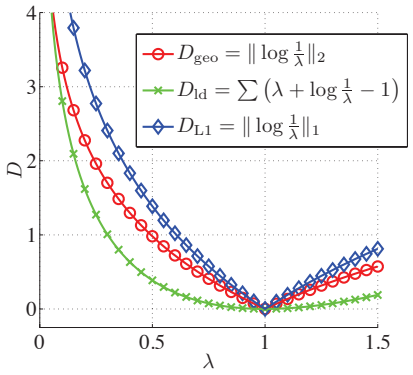


Fig. 7. Comparison of dissimilarity measures in the 2×2 case: Slice at $\lambda_1 = \lambda_2 = \lambda$. Clearly all three distance functions have their minimum at $\lambda_1 = \lambda_2 = 1$. In terms of how 'strong' the objective function is in pushing the λ_i's to 1, $D_{\mathrm{ld}} < D_{\mathrm{geo}} < D_{\mathrm{L1}}$.

9 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

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. `YALMIP` [58] and `SDPT3` [59] are used for the tensor sparse coding implementation, and `LIBSVM` [60] for the SVM implementation.

REFERENCES

[1] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society (Series B)*, vol. 58, pp. 267–288, 1996.

[2] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–451, 2004.

[3] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[4] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[5] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[6] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.

[7] T. Hazan, S. Polak, and A. Shashua, "Sparse image coding using a 3d non-negative tensor factorization," in 10^{th} *IEEE Intl. Conf. on Computer Vision, 2005.*, vol. 1, Oct. 2005, pp. 50–57.

[8] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[9] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *ECCV 2006*, 2006, pp. 589–600.

[10] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on Lie algebra," in *IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition 2006*, vol. 1, Jun. 2006, pp. 728–735.

[11] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on Riemannian manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1713–1727, Oct. 2008.

[12] S. Smith, "Covariance, subspace, and intrinsic Cramer-Rao bounds," *IEEE Trans. Signal Process.*, vol. 53, no. 5, pp. 1610–1630, 2005.

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

[14] X. Pennec, "Statistical computing on manifolds: From Riemannian geometry to computational anatomy," in *Emerging Trends in Visual Computing*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, vol. 5416, pp. 347–386.

[15] 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, Apr. 1998.

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

[17] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-euclidean metrics for fast and simple calculus on diffusion tensors," *Magnetic Resonance in Medicine*, vol. 56, no. 2, pp. 411–421, Aug. 2006.

[18] R. Bhatia, *Positive Definite Matrices*. Princeton, NJ, USA: Princeton University Press, 2007.

[19] I. L. Dryden, A. Koloydenko, and D. Zhou, "Non-euclidean statistics for covariance matrices, with applications to diusion tensor imaging," *The Annals of Statistics*, vol. 3, no. 3, pp. 1102–1123, 2009.

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

[21] H. Palaio and J. Batista, "Multi-object tracking using an adaptive transition model particle filter with region covariance data association," in *19th Intl. Conf. on Pattern Recognition, 2008.*, Dec. 2008, pp. 1–4.

[22] H. Wildenauer, B. Mičušík, and M. Vincze, "Efficient texture representation using multi-scale regions," in *Proc. 8th Asian Conf. on Computer vision - Volume Part I*, ser. ACCV'07, 2007, pp. 65–74.

[23] J. Y. Tou, Y. H. Tay, and P. Y. Lau, *Gabor Filters as Feature Images for Covariance Matrix on Texture Classification Problem*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 745–751.

[24] 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.*, Nov. 2006, pp. 107–107.

[25] A. Ruta, F. Porikli, S. Watanabe, and Y. Li, "In-vehicle camera traffic sign detection and recognition," *Machine Vision and Applications*, pp. 1–17, 2009.

[26] Y. Pang, Y. Yuan, and X. Li, "Gabor-based region covariance matrices for face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 989–993, Jul. 2008.

[27] 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 and Signal Processing 2010*, Mar. 2010, pp. 1110–1113.

[28] P. C. Cargill, C. U. Rius, D. M. Quiroz, and A. Soto, "Performance evaluation of the covariance descriptor for target detection," in *Intl. Conf. of the Chilean Computer Science Society 2009*, Nov. 2009, pp. 133–141.

[29] S. Zheng, H. Qiao, B. Zhang, and P. Zhang, "The application of intrinsic variable preserving manifold learning method to tracking multiple people with occlusion reasoning," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems 2009.*, Oct. 2009, pp. 2993–2998.

[30] J. Wang and Y. Wu, "Visual tracking via incremental covariance model learning," in *Second Intl. Conf. on Computer Modeling and Simulation 2010*, vol. 1, Jan. 2010, pp. 277–280.

[31] F. Porikli, "Learning on manifolds," in *Proc. 2010 joint IAPR Intl. Conf. on Structural, Syntactic, and Statistical Pattern Recognition*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 20–39.

[32] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Intl. Conf. on Machine Learning*. New York, NY, USA: ACM, 2004, p. 6.

[33] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Fast kernel learning using sequential minimal optimization," EECS Department, University of California, Berkeley, Tech. Rep. UCB/CSD-04-1307, Feb. 2004.

[34] Y. Ying, K. Huang, and C. Campbell, "Enhanced protein fold recognition through a novel data integration approach," *BMC Bioinformatics*, vol. 10, no. 1, p. 267, 2009.

[35] K. Tsuda, G. Rätsch, and M. K. Warmuth, "Matrix exponentiated gradient updates for on-line learning and Bregman projection," *J. Mach. Learn. Res.*, vol. 6, pp. 995–1018, Dec. 2005.

[36] G. Meyer, S. Bonnabel, and R. Sepulchre, "Regression on fixed-rank positive semidefinite matrices: A Riemannian approach," *J. Mach. Learn. Res.*, vol. 12, pp. 593–625, Feb. 2011.

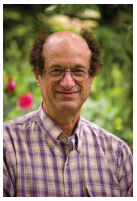
[37] Y. E. Nesterov and M. Todd, "On the Riemannian geometry defined by self-concordant barriers and interior-point methods," *Foundations of Computational Mathematics*, vol. 2, no. 4, pp. 333–361, 2002.

[38] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *24th Intl. Conf. on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 209–216.

[39] K. Guo, P. Ishwar, and J. Konrad, "Action recognition using sparse representation on covariance manifolds of optical flow," in *Seventh*

IEEE Intl. Conf. on Advanced Video and Signal Based Surveillance 2010, Sep. 2010, pp. 188–195.
[40] Z. Wang and B. Vemuri, "DTI segmentation using an information theoretic tensor dissimilarity measure," *IEEE Trans. Med. Imag.*, vol. 24, no. 10, pp. 1267–1277, Oct. 2005.
[41] S. Sra and A. Cherian, "Generalized dictionary learning for symmetric positive definite matrices with application to nearest neighbor retrieval," in *Proc. 2011 European Conf. on Machine Learning and Knowledge Discovery in Databases - Volume Part III*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 318–332.
[42] G. Pfander, H. Rauhut, and J. Tanner, "Identification of matrices having a sparse representation," *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5376–5388, Nov. 2008.
[43] H. Wang, A. Banerjee, and D. Boley, "Modeling time varying covariance matrices in low dimensions," Dept. of Computer Science and Engineering, University of Minnesota, Technical Report TR-10-017, Aug. 2010.
[44] B. Kulis, M. Sustik, and I. Dhillon, "Learning low-rank kernel matrices," in *Proc. 23rd Intl. Conf. on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 505–512.
[45] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 3, pp. 200–217, 1967.
[46] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *J. Mach. Learn. Res.*, vol. 6, pp. 1705–1749, Dec. 2005.
[47] J. Wishart, "The generalized product moment distribution in samples from a normal multivariate population," *Biometrika*, vol. 20A, no. 1-2, pp. 32–52, 1928.
[48] G. Wang, Y. Liu, and H. Shi, "Covariance tracking via geometric particle filtering," in *2nd Intl. Conf. on Intelligent Computation Technology & Automation, 2009*, vol. 1, Oct. 2009, pp. 250–254.
[49] A. Gelman, J. B. Carlin, H. S. Stern, and D. S. Rubin, *Bayesian Data Analysis, Second Edition*. Boca Raton: Chapman & Hall/CRC, 2003.
[50] J. Tropp, "Just relax: convex programming methods for identifying sparse signals in noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1030–1051, Mar. 2006.
[51] D. L. Donoho and J. Tanner, "Sparse nonnegative solution of underdetermined linear equations bylinear programming," *National Academy of Sciences of the United States of America*, vol. 102, no. 27, pp. 9446–9451, 2005.
[52] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *Advances in Neural Information Processing Systems 16*. Cambridge, MA: MIT Press, 2004.
[53] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems 16*, T. Leen, T. Dietterich, and V. Tresp, Eds. Cambridge, MA: MIT Press, 2000, vol. 13, pp. 556–562.
[54] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
[55] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.
[56] T. Randen and J. H. Husøy, "Filtering for texture classification: A comparative study," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 21, pp. 291–310, Apr. 1999.
[57] B. Kulis, M. A. Sustik, and I. S. Dhillon, "Low-rank kernel learning with Bregman matrix divergences," *J. Mach. Learn. Res.*, vol. 10, pp. 341–376, Jun. 2009.
[58] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *Proc. CACSD Conf.*, Taipei, Taiwan, 2004. [Online]. Available: http://users.isy.liu.se/johanl/yalmip
[59] R. H. Tutuncu, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3." *Math. Program.*, vol. 95, no. 2, pp. 189–217, 2003.
[60] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Ravishankar Sivalingam received his Bachelors in Electronics and Communication Engineering from Anna University, India in 2006. He received his M.S in Computer Science in 2009 and M.S. in Electrical Engineering in 2010 both from the University of Minnesota. He is currently a PhD candidate in Electrical Engineering at the University of Minnesota. His primary interests lie in the domains of computer vision, pattern recognition, and machine learning. His current and past projects include aerial image processing (image registration, mosaicing and region annotation), people detection, tracking and crowd counting, generic object detection and application of sparsity and dictionary learning techniques.



Daniel Boley received his A.B. degree Summa Cum Laude in Mathematics and with Distinction in All Subjects from Cornell University in 1974, and his M.S. and Ph.D. degrees in Computer Science from Stanford University in 1976 and 1981, respectively. Since 1981, he has been on the faculty of the Department of Computer Science and Engineering at the University of Minnesota, where he is now a full professor. He has had extended visiting positions at the Los Alamos Scientific Laboratory, the IBM Research Center in Zurich (Switzerland), the Australian National University in Canberra, Stanford University, and the University of Salerno (Italy). Dr. Boley is known for his past work on numerical linear algebra methods for control problems, parallel algorithms, iterative methods for matrix eigenproblems, error correction for floating point computations, inverse problems in linear algebra, as well as his more recent work on computational methods in statistical machine learning and unsupervised document categorization in data mining and bioinformatics. He is an associate editor for the SIAM Journal of Matrix Analysis and has chaired several technical symposia at major conferences. His current interests involve scalable methods for data mining with applications in bioinformatics, computational biology, large collections of text documents (most recently e-mail for the study of social networks), etc.

Vassilios Morellas received his Diploma of Engineering in Mechanical Engineering, from the National Technical University of Athens in 1983. He received his M.S. in Mechanical Engineering from Columbia University in 1988, and PhD in Mechanical Engineering from the University of Minnesota in 1995. Vassilios Morellas research interests are in the area of geometric image processing, machine learning, robotics and sensor integration to enhance automation of electromechanical systems. He is the Program Director in the department of Computer Science and Engineering and Executive Director of the NSF Center for Safety Security and Rescue. Prior to his current position he was a Senior Principal Research Scientist at Honeywell Laboratories where he developed technologies in the general areas of access control, security and surveillance and biometrics with emphasis on the problem of tracking of people and vehicles across non overlapping cameras. Past research experience also includes work on Intelligent Transportation Systems where he developed innovative technologies to reduce run-off-the-road accidents.

Nikolaos Papanikolopoulos received his Diploma of Engineering in Electrical and Computer Engineering, from the National Technical University of Athens in 1987. He received his M.S. in 1988 and PhD in 1992 in Electrical and Computer Engineering from Carnegie Mellon University. Professor Papanikolopoulos specializes in robotics, computer vision and sensors for transportation uses. His research interests include robotics, sensors for transportation applications, computer vision, and control systems. As the director of the Center for Distributed Robotics and a faculty member of the Artificial Intelligence and Robotic Vision Laboratory, his transportation research has included projects involving vision-based sensing and classification of vehicles, and the recognition of human activity patterns in public areas and while driving.