Clustering Positive Definite Matrices by Learning Information Divergences

Panagiotis Stanitsas[1] Anoop Cherian[2] Vassilios Morellas [1] Nikolaos Papanikolopoulos[1]
[1]Department of Computer Science, University of Minnesota, Minneapolis
[2]Australian Centre for Robotic Vision, The Australian National University
{stani078, morellas, npapas}@umn.edu, anoop.cherian@anu.edu.au

Abstract

Data representations based on Symmetric Positive Definite (SPD) matrices are gaining popularity in visual learning applications. When comparing SPD matrices, measures based on non-linear geometries often yield beneficial results. However, a manual selection process is commonly used to identify the appropriate measure for a visual learning application. In this paper, we study the problem of clustering SPD matrices while automatically learning a suitable measure. We propose a novel formulation that jointly (i) clusters the input SPD matrices in a K-Means setup and (ii) learns a suitable non-linear measure for comparing SPD matrices. For (ii), we capitalize on the recently introduced $\alpha\beta$-logdet divergence, which generalizes a family of popular similarity measures on SPD matrices. Our formulation is cast in a Riemannian optimization framework and solved using a conjugate gradient scheme. We present experiments on five computer vision datasets and demonstrate state-of-the-art performance.

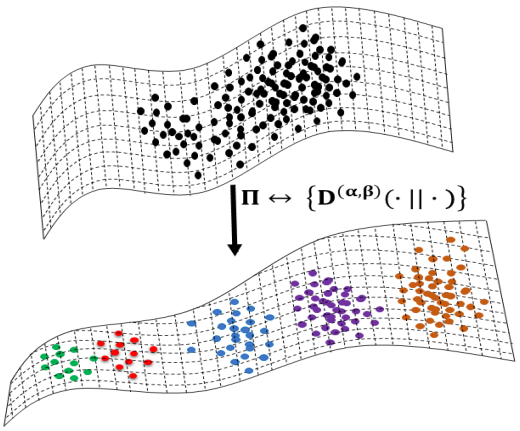


Figure 1. A schematic illustration of $\alpha\beta$-KMeans. Our scheme jointly learns a data partitioning (Π) and a distance measure for comparing SPD matrices based on the $\alpha\beta$-logdet divergence $D^{(\alpha,\beta)}(\cdot \parallel \cdot)$. Note that learning the divergence changes the structure of the data manifold, so that the partitioning (clustering) is more effective.

1. Introduction

Unsupervised clustering of data is a fundamental operation in computer vision applications. Clustering allows exploratory data analysis in the absence of data annotations and helps identify basic data patterns that are useful for higher-level semantic inference. In this paper, we investigate clustering algorithms for data that are in the form of SPD matrices. Such structured matrix-valued data descriptors are widely encountered in several computer vision problems and have been shown to provide significant performance advantages over other data descriptors. This is because of their ability to capture rich second-order data statistics, which are essential for recognition tasks (e.g., [7, 16, 20, 37, 38]).

When using SPD matrices in computer vision problems, one usually faces the difficulty pertinent to selecting an appropriate similarity measure for comparing the input matrices. This is because each element in an SPD matrix (usually generated as covariance on data features) encode the correlations implicit in data and thus is a very structured object. While one may ignore this structure and assume an SPD matrix to belong to a Euclidean geometry, it is often found that using non-linear (often Riemannian) geometries that use the spectral properties[1] of these matrices for similarity computations lead to significantly better application performance (e.g., [3, 31]). However, there are several ways that one could capture the spectral similarity between two such input matrices. A few popular measures are (i) the affine-invariant Riemannian metric (AIRM) using the natural Riemannian geometry [31], (ii) the Jensen-Bregman logdet divergence using information geometry [9], and (iii) Burg matrix divergence [22], among several others [11]. Choosing an appropriate measure for a given application is often based on empirically and is usually a manual process.

[1]That is, similarities that use the Eigen spectrum of the matrices.

Recently, Cichocki et al. [11] showed that all the above similarity measures on SPD matrices can in fact be written as different parameterizations of the so-called $\alpha\beta$-Logdet Divergences (ABLD). These divergences directly parameterize the generalized eigenspectrum of the input matrices. It is shown that by choosing different values for the parameters α and β, one can enumerate various popular divergences on SPD matrices (see Table 1). Interestingly, even the natural Riemannian metric (AIRM) can be written as an asymptotic limit of ABLD at the origin ($\alpha \to \beta \to 0$). This unifying result suggests that we could learn to choose an appropriate similarity measure in a data-driven way. Leveraging on this result, we explore schemes for K-Means clustering of SPD matrices, that also learns a suitable similarity measure by finding appropriate values of α and β. Conceptually, we illustrate an outline of the proposed methodology in Figure 1.

We furnish experiments on several computer vision datasets to evaluate the advantage of joint clustering and similarity learning (Section 7.4.3). Our results demonstrate that learning the measure is beneficial and lead to superior clustering performance. We also evaluate the quality of clusters on higher-level tasks such as nearest neighbor retrieval, and demonstrate better accuracy over alternatives.

2. Related Work

The $\alpha\beta$-divergence is an information divergence generalizing the α-family divergences [2] (that includes popular measures such as the KL-divergence, Jensen-Shannon divergence, and the chi-square divergence) and the β-family [4] (including the squared Euclidean distance and the Itakura Saito distance). In contrast to standard measures (such as the Euclidean distance), both α and β divergences are seen to provide more robust solutions in the presence of outliers and additive noise [24]. These divergences have also been used in machine learning tasks, including but not limited to non-negative matrix factorization [12, 21, 13], nearest neighbor embedding [18], and blind-source separation [26]. The $\alpha\beta$-logdet divergence (ABLD) is a matrix generalization of the scalar $\alpha\beta$-divergence and compares SPD matrix valued data points. In this work, we investigate approaches for clustering SPD matrices using ABLD in a Karcher means setup, however, we also propose to learn the suitable information divergences by learning appropriate values of α and β together with the clustering objective.

Several unsupervised schemes for clustering SPD matrices have been proposed in the relevant literature. Commonly used schemes capitalize on conventional clustering machinery after being modified towards abiding to the non-linear geometry of SPD matrices. In this direction, two extensions of the popular KMeans have been derived admitting the manifold of the SPD matrices. In the first variant of KMeans, centroids are computed using the Karcher means algorithm [6] and the affine-invariant Riemannian metric [31]. Substituting the similarity computation based on the AIRM by the log-Euclidean metric [3], yields a second variant of KMeans for SPD matrices termed LE-KMeans. Using the matrix logarithm operation, which entails a diffeomorphic mapping of an SPD matrix onto its tangent space, allows for distance computations in a Euclidean manner (as this tangent space is Euclidean). In that way, centroids are computed by averaging the samples' vectorial representations in the tangent space. Additional variants of KMeans can be derived by capitalizing on the different similarity measures for SPD matrices.

A second family of clustering schemes for SPD matrices takes advantage of Euclidean embeddings in the form of similarity matrices computed using suitable measures. In that direction, Spectral clustering schemes have been developed for SPD matrices by computing suitable Mercer kernels on the data using appropriate distances (e.g., LE). Sparse subspace clustering schemes have also been derived for SPD matrices via their embedding into a Reproducing Kernel Hilbert Space [39, 29]. Such schemes come at the expense of additional memory requirements involved with computing the eigen spectrum of the computed kernel.

In addition, non-parametric schemes for clustering SPD matrices have been derived in the form of dimensionality reduction on Riemannian manifolds [14], using Locally Linear Embeddings [32], or capitalizing on the Laplacian eigenmaps [5]. In the family of non-parametric clustering algorithms, a Bayesian framework for SPD matrices is formulated using the Dirichlet Process [8]. Finally, variants of the Mean shift clustering algorithm and Kernel Density Estimation for SPD matrices have also been derived in [36] and [30] respectively.

Moreover, metric learning schemes have also been proposed for SPD matrices. Learning a manifold to manifold embedding of large SPD matrices to small SPD spaces is proposed in [17]. Furthermore, embeddings capitalizing on the Log-Euclidean metric learning framework have been also proposed (e.g. [19, 33]). Such metric learning schemes require labeled data and thus do not share the same objective as this work, which is to provide inference in an unsupervised setup.

In contrast to all these methods, to the best of our knowledge, it is for the first time that a joint distance learning and clustering formulation is derived for SPD matrices. We note that recently, learning $\alpha\beta$-divergence in a discriminative setup is proposed in [10]. However, differently to that work, we study the problem of clustering in this paper and thus our objective is different.

In the sequel, we proceed by introducing the $\alpha\beta$-logdet divergence and explore its properties in the next section. Following that, we present our derivation of $\alpha\beta$-KMeans that achieves the aforementioned objectives.

(α, β)	ABLD	Divergence
$(\alpha, \beta) \to 0$	$\left\|\operatorname{Log} X^{-\frac{1}{2}} Y X^{-\frac{1}{2}}\right\|_F^2$	Squared Affine Invariant Riemannian Metric [31]
$\alpha = \beta = \pm\frac{1}{2}$	$4\left(\log\det\frac{X+Y}{2} - \frac{1}{2}\log\det XY\right)$	Jensen-Bregman Logdet Divergence [9]
$\alpha = \pm 1, \beta \to 0$	$\frac{1}{2}Tr\left(XY^{-1} + YX^{-1}\right) - d$	Jeffreys KL Divergence[2] [27]
$\alpha = 1, \beta = 1$	$Tr\left(XY^{-1}\right) - \log\det XY^{-1} - d$	Burg Matrix Divergence [22]

Table 1. ABLD and its connections to popular divergences used in computer vision applications.

Notations: Following standard notations, we use upper case for matrices (such as X), lower-bold case for vectors $\mathbf{x}$, and lower case for scalars x. Further, $\mathcal{S}_{++}^d$ is used to denote the cone of $d \times d$ SPD matrices. We use $\mathbf{C}$ to denote a 3D tensor each slice of which corresponds to an SPD centroid of size $d \times d$. Further, we use $\mathbf{I}_d$ to denote the $d \times d$ identity matrix, Log for the matrix logarithm, and diag for the diagonalization operator. Finally, we use $\mathbf{\Pi} = \{\pi_1, ..., \pi_k\}$ to denote a clustering of data into k partitions; π_i is the i-th partition and comprises a subset of the dataset assigned to this cluster.

3. Background

3.1. $\alpha\beta$-Log Determinant Divergence

Definition 1 (ABLD [11]) *For $X, Y \in \mathcal{S}_{++}^d$, the $\alpha\beta$-log-det divergence is defined as:*

$$D^{(\alpha,\beta)}(X\|Y) = \frac{1}{\alpha\beta}\log\det\left(\frac{\alpha(XY^{-1})^{\beta} + \beta(XY^{-1})^{-\alpha}}{\alpha+\beta}\right), \tag{1}$$

$$\alpha \neq 0,\ \beta \neq 0 \ \text{ and } \alpha + \beta \neq 0. \tag{2}$$

As can be easily verified, ABLD can be rewritten to use only the generalized eigenvalues of X and Y [11]. Let λ_i denote the i-th eigenvalue of XY^{-1}. Then, (1) in terms of λ_i is given by:

$$D^{(\alpha,\beta)}(X\|Y) = \frac{1}{\alpha\beta}\sum_{i=1}^{d}\log\left(\alpha\lambda_i^{\beta} + \beta\lambda_i^{-\alpha}\right) - d\log(\alpha+\beta). \tag{3}$$

As pointed out earlier, ABLD unifies several standard metrics and divergernces on SPD matrices. We explicitly list some of the popular ones in Table 1 along with the respective values of α and β.

For the sake of completeness of our presentation, we list below some important theoretical properties of ABLD that will come handy when deriving optimization algorithms for clustering in the sequel.

Degeneracy Solutions: As can be observed, for ABLD to generate non-negative real values (as is required by any distance metric), $\alpha\lambda_i^{\beta} + \beta\lambda_i^{-\alpha} > 0$ for all $i = 1, 2, \cdots, d$. As imposing such constraints make our optimization very expensive, in this paper we use a simplification by learning α and β of the same sign.

Dual Symmetry: This property allows expressions derived with respect to α to be used for β with some substitutions. Specifically,

$$D^{(\alpha,\beta)}(X \parallel Y) = D^{(\beta,\alpha)}(Y \parallel X). \tag{4}$$

Smoothness of ABLD: Assuming α, β have the same sign, ABLD is continuous everywhere, except at the origin. As noted above, when $\alpha = \beta \to 0$, ABLD is exactly the AIRM distance.

With this machinery, we present our $\alpha\beta$-KMeans formulation and optimization schemes.

4. $\alpha\beta$-Kmeans

Let $\mathcal{X}$ denote an SPD matrix-valued dataset. That is, $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N\}$, where each $\mathbf{X}_i \in \mathcal{S}_{++}^d$. Our goal is to cluster the data into k clusters, where k is assumed given. Let $\mathbf{\Pi} = \{\pi_1, \cdots, \pi_k\}$ denote a partitioning of $\mathcal{X}$ where π_i is the set of samples assigned to the i-th cluster and let $\mathbf{C}_i$ be the respective cluster centroid. We cast the joint information divergence learning and clustering problem as:

$$\min_{\mathbf{C},\mathbf{\Pi},\alpha,\beta>0} f(\mathbf{\Pi}, \mathcal{X}; \alpha, \beta) + \mathbf{\Omega}(\alpha, \beta), \tag{5}$$

where we envisage learning the cluster assignment of data points $\mathbf{\Pi}$, the cluster centroids $\mathbf{C}$, and the divergence scalar parameters α, β together. There are potentially two variants of α and β that we could learn, namely (i) by observing that most of the popular divergences in Table 1 are obtained when $\alpha = \beta$, we could assume this, thereby simplifying our objective, and (ii) we could also assume $\alpha \neq \beta$, which is more general, potentially incorporating (i). However, we postpone exploring these two possibilities until Section 7 and use the more general assumption (ii) for our theoretical derivations below. In (5), the function $\mathbf{\Omega}(\alpha, \beta) = \mu\left(\alpha^2 + \beta^2\right)$ is a regularization term on the parameters α and β, and μ is a regularization constant. The use of this term is seen to be unavoidable for numerical stability of our descent schemes, while the quadratic prior choice is seen to provide faster empirical convergence compared to the linear case. Substituting the standard KMeans formulation and using the $\alpha\beta$-divergence as the similarity measure, we have the following definition for f in (5):

$$f(\mathbf{\Pi}, \mathcal{X}; \alpha, \beta) = \sum_{\pi\in\mathbf{\Pi}}\sum_{i\in\pi}\left(D^{(\alpha,\beta)}(X_i \parallel \mathbf{C}_\pi)\right). \tag{6}$$

5. Efficient Optimization

In this section, we propose efficient ways to solve the $\alpha\beta$-KMeans objective described in (5). Following the standard Lloyd's algorithm for solving KMeans formulation, we propose to use a block-coordinate descent (BCD) scheme for optimization, in which each variable is updated alternately while fixing others. As depicted in Algorithm 1, our BCD has three main sub-problems, namely (i) solving for α and β, (ii) computing the centroids $\mathbf{C}$, and (iii) computing the data partition Π. Below, we detail each of these sub-problems and how we solve each of them.

Input: $\mathcal{X}$, k
$\mathbf{C} \leftarrow$ init$(\mathcal{X}, k)$, $(\alpha, \beta) \leftarrow$ init(lb, up);
repeat
  $(\alpha, \beta) \leftarrow$ update_$\alpha\beta(\mathcal{X}, \mathbf{\Pi}, \alpha, \beta, \mathbf{C}_z)$; // use (13)
  $\mathbf{\Pi} \leftarrow$ update_$\Pi(\mathcal{X}, \alpha, \beta, \mathbf{C})$; // use (7)
  for $z = 1$ **to** k **do**
    $\mathbf{C}_z \leftarrow$ update_C$(\mathcal{X}, \mathbf{\Pi}, \alpha, \beta, \mathbf{C}_z)$; // use (12)
  end
until *until convergence*;
return $\mathbf{C}, \mathbf{\Pi}, \alpha, \beta$

Algorithm 1: Overview of Block-Coordinate Descent for $\alpha\beta$-KMeans.

5.1. Updating Data Partitioning, Π

As is clear, updating Π is the easiest to attempt. That is, finding the cluster centroid $\mathbf{C}_\pi$ nearest to a given data point X_i, which we solve as the following argmin problem, by assuming the ABLD parameters are fixed at the current iterate, i.e., α_t and β_t at t-th BCD iteration. Formally, the data points in the cluster π_z are updated as $\pi_{z^*} \rightarrow \pi_{z^*} \cup \{X_i\}$, where

$$z^* = \underset{\forall z \in \{1,2,\cdots,k\}}{\arg\min} \; D^{(\alpha_t,\beta_t)}(X_i \parallel \mathbf{C}_z^t). \tag{7}$$

5.2. Updating Cluster Centroids, $\mathbf{C}$

The sub-problem of computing the centroid $\mathbf{C}_\pi$ for a cluster π is given by the following barycenter finding problem:

$$\mathbf{C}_\pi^t = \min_{\mathbf{C} \in \mathcal{S}_{++}^d} f(\Pi^t, \mathbf{C}; \alpha^t, \beta^t) := \sum_{X \in \pi} D^{(\alpha_t,\beta_t)}(X \parallel \mathbf{C}). \tag{8}$$

Given that this optimization is over all SPD matrices, we resort to casting it in a Riemannian optimization setup solving it using a conjugate gradient algorithm on the SPD manifold. With the recent advances in Riemannian optimization schemes [1], all we need to define to use a Riemannian conjugate gradient for solving (8) is to get the expressions for its Euclidean gradient, which we derive below:

$$\nabla_{\mathbf{C}} f := \nabla_{\mathbf{C}} \left(D^{(\alpha_t,\beta_t)}(X_i \parallel \mathbf{C}) \right). \tag{9}$$

Substituting ABLD in (9) and rearranging the terms, we have:

$$\nabla_{\mathbf{C}} f = \frac{1}{\alpha_t \beta_t} \nabla_{\mathbf{C}} \log\det \left[\frac{\alpha_t}{\beta_t} \left(X_i^{-1} \mathbf{C} \right)^{\alpha_t + \beta_t} + \mathbf{I}_d \right] - \frac{1}{\beta_t} \mathbf{C}^{-1}. \tag{10}$$

Let $\theta = \alpha + \beta$ and $r = \frac{\alpha}{\beta}$. Further, let $Z_i = X_i^{-1}$. Then, the term inside the gradient in (10) simplifies to:

$$g(\mathbf{C}; Z, r, \theta) = \log\det \left[r \left(Z\mathbf{C}_z \right)^\theta + \mathbf{I}_d \right], \tag{11}$$

and its gradient is given by:

$$\nabla_{\mathbf{C}} g = r\theta \mathbf{C}^{-1} Z_i^{-\frac{1}{2}} \left(Z_i^{\frac{1}{2}} \mathbf{C} Z_i^{\frac{1}{2}} \right)^\theta \left(\mathbf{I}_d + r \left(Z_i^{\frac{1}{2}} \mathbf{C} Z_i^{\frac{1}{2}} \right)^\theta \right)^{-1} Z_i^{\frac{1}{2}}. \tag{12}$$

Substituting (12) in (10) gives the gradient of f with respect to $\mathbf{C}$.

5.3. Updating the Information Divergence, α and β

For gradients with respect to α, we will use the form of ABLD given in (3), where λ_{ijz} is assumed to be the j-th generalized eigenvalue of X_i and centroid $\mathbf{C}_\pi$ such that $X_i \in \pi$, we get:

$$\begin{aligned} \nabla_\alpha f &= \sum_{j=1}^{d} \nabla_\alpha \left[\frac{1}{\alpha\beta} \log \frac{\alpha \lambda_{ijz}^\beta + \beta \lambda_{ijz}^{-\alpha}}{\alpha + \beta} \right] \\ &= \frac{1}{\alpha^2 \beta} \sum_{j=1}^{d} \left\{ \frac{\alpha \lambda_{ijz}^\beta - \alpha\beta \lambda_{ijz}^{-\alpha} \log \lambda_{ijz}}{\alpha \lambda_{ijz}^\beta + \beta \lambda_{ijz}^{-\alpha}} \right. \\ &\qquad \left. - \frac{\alpha}{\alpha + \beta} - \log \frac{\alpha \lambda_{ijz}^\beta + \beta \lambda_{ijz}^{-\alpha}}{\alpha + \beta} \right\}. \end{aligned} \tag{13}$$

Using dual symmetry of ABLD allows computing the expressions for gradients wrt β directly from (13).

6. Computational Complexity

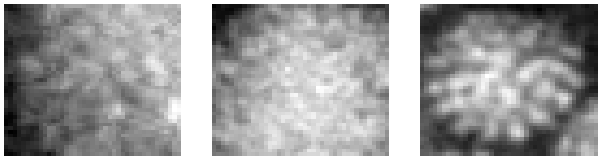
Using Schur decomposition (instead of the expensive eigenvalue decomposition), gradient computation for each $\mathbf{C}$ takes $\mathcal{O}(Nd^3)$ flops. Using the gradient formulation in (13) for α and β, we need $\mathcal{O}(Ndk + Nd^3)$ flops, similar in complexity to a Karcher mean algorithm using AIRM as the similarity measure.

7. Experiments

In this section, we provide an evaluation of the proposed $\alpha\beta$-KMeans on five recognition datasets. In that direction, we present two sets of experiments evaluating $\alpha\beta$-KMeans against different KMeans variants for SPD matrices. The first experiment, targets a pure clustering setup, while the second one uses clustering as a pre-processing step in a Bag-of-Words setup. The following datasets are used, namely (i) the KTH-TIPS2 dataset [25], (ii) Brodatz textures [28], (iii) the Virus dataset [23], (iv) the Myometrium cancer dataset [34, 35], and (v) the Prostate cancer dataset [34, 35]. Below, we provide details of all these datasets and the way SPD descriptors are obtained on them. Furthermore, we present a sensitivity analysis with respect to the number of dimensions and number of clusters for the proposed scheme as well as an empirical convergence analysis. Note that we explore two variants of ABLD, namely (i) when $\alpha = \beta$ and when $\alpha \neq \beta$.

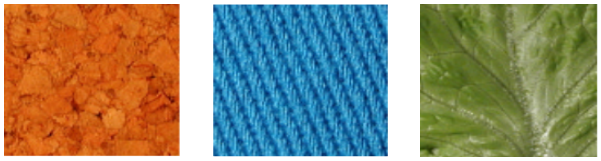
7.1. Datasets

KTH-TIPS2 dataset and Brodatz Textures: The KTH-TIPS-2 dataset [25], is a popular material recognition database consisting of 4,752 samples depicting 11 materials; TIPS standing for 'Textures under varying Illumination, Pose and Scale'. Three samples from this database are presented in Figure 3. Region Covariance Descriptors of size 23×23 are computed on the features proposed in Harandi et al. [15]. As for the Brodatz dataset, we use the relative pixel coordinates, image intensity, and image gradients to form 5×5 region covariance descriptors from 100 texture classes. Our dataset consists of 31000 SPD matrices.



(a) (b) (c)

Figure 2. Samples of the VIRUS dataset for classes (a) Ebola, (b) Influenza, and (c) Orf.



(a) (b) (c)

Figure 3. Samples of the KTH-TIPS2 dataset for classes (a) Cork, (b) Cotton and (c) Lettuce.

Virus Dataset: The VIRUS dataset is a collection of 1500, Transmission Electron Microscopy (TEM) images belonging to 15 different virus types, which are automatically segmented based on [23]. Three sample images are presented in Figure 2. The selected descriptors are Region Covariances of size 29×29, computed on the features proposed as suggested in Harandi et al. [15].

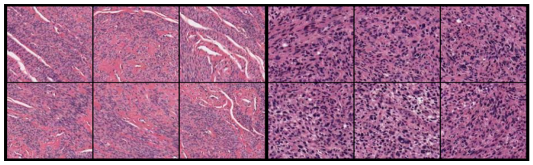


Figure 4. Myometrium tissue H&E stained samples. Columns 1-3 correspond to patches depicting benign cases while columns 4-6 correspond to patches depicting malignant cases.

Cancer Datasets. Apart from these standard SPD datasets, we also report performances on two cancer recognition datasets from [34] kindly shared with us by the authors. We use images from two types of cancers, namely (i) Myometrium cancer, consisting of binary classes (tissue is either cancerous or not) consisting of about 3320 samples, and (ii) Prostate cancer, consisting of 3315 samples; we use covariance-kernel descriptors as described in [34] which are of size 8×8. For Myometrium cancer we present a collection of malignant and benign samples in Figure 4.

7.2. Experimental Setup

We present experiments on the aforementioned benchmarks and compare against two popular variants of KMeans for SPD data. In that direction, we establish comparisons against the Log-Euclidean KMeans, as well as the Karcher means using AIRM. To evaluate the quality of the derived partitions of data in clusters, we use the F1-Score (14) that is a weighted average of precision and recall and ranges between 0 and 1, with 1 corresponding to the optimal partition of the data. Furthermore, we present comparisons between the aforementioned clustering schemes in a Bag-of-Words recognition setup in terms of F1-Score, as well as accuracy. The F_1 score is defined as:

$$F_1 = 2\,\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{14}$$

7.3. Parameter Initialization

In all our experiments, we initialize the centroids using LE-KMeans. In that direction, we first compute the matrix logarithm of the given samples, then concatenate the columns of each sample and compute a partitioning of the vectorial data using Euclidean KMeans. The resulting centroids are then reshaped to their original structure and utilizing the exponential map are projected back to the SPD

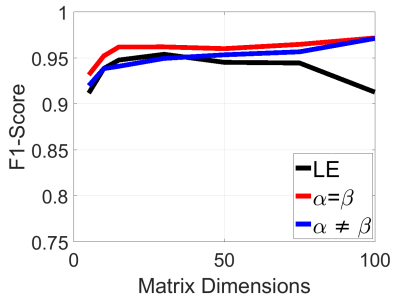
cone. Furthermore, parameters α and β are selected at random within the range of zero and ten. Finally, the regularization parameter μ is set to 1.

7.4. Sensitivity Analysis

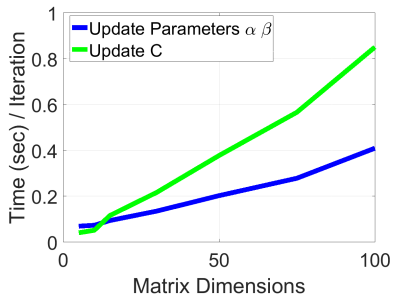
In this section, we present the sensitivity of our algorithm to different attributes of the input dataset. We provide plots illustrating the responsiveness of the proposed scheme against the following, namely (i) dimensionality of the input matrices and (ii) number of clusters and (iii) elapsed for the relevant updates. The synthetic datasets are generated using the code from [8]. This code generates Wishart SPD matrix clusters for k arbitrarily parameterized Wishart distributions.

7.4.1 Dimensionality

Towards assessing the performance of $\alpha\beta$-KMeans for SPD matrices of varying dimensionality, we generate synthetic SPD datasets of dimensionality d, where $d \in \{5, 15, 30, 50, 75, 100\}$ corresponding to $k = 15$ clusters and using fifty samples per class. Figure 5 (a) summarizes the computed F1-scores averaged across ten runs. We can clearly see that $\alpha\beta$-KMeans is not impacted by the increasing dimensions of the input matrices, while both variants consistently outperform the baseline of LE-KMeans. Figure 5 (b) present the time takes for a single iteration of each optimization component of $\alpha\beta$-KMeans.



(a)



(b)

Figure 5. Sensitivity of $\alpha\beta$-KMeans against an increasing dimensionality in the range $[5, 100]$. (a) The blue and red lines correspond to $\alpha\beta$-KMeans with $\alpha = \beta$ and $\alpha \neq \beta$ respectively, while the black line corresponds to LE-KMeans. (b) Time required for each iteration of updating parameters $\alpha\beta$ (blue line) and centroids (green line).

7.4.2 Number of Clusters

Similarly, in this section we extract valuable conclusions regarding the performance of $\alpha\beta$-KMeans against an increasing number of clusters in a simulated dataset. We test the robustness of the $\alpha\beta$-KMeans algorithm for a cluster number k, such that $k \in \{2, 5, 10, 20, 50, 100\}$ keeping the dimension of the SPD matrices fixed to $d = 10$ and using twenty five samples per class. Figure 6 (a) summarizes the F1-Score of $\alpha\beta$-KMeans averaged across ten runs for an increasing number of clusters. We can infer that both variants are negatively affected by large increases in the number of clusters in the dataset, nevertheless, the performance is consistently higher than that of the LE-KMeans baseline. In addition, there is an increasing overall trend in the time required for all components of $\alpha\beta$-KMeans, while the results of having to iterate through the different clusters as depicted in Figure 6 (b).

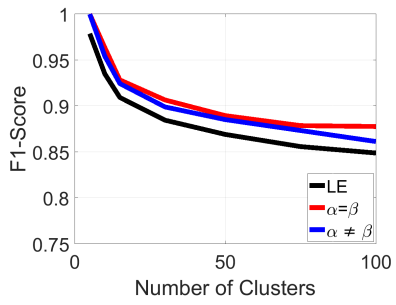
7.4.3 Empirical Convergence Analysis

In this section we empirically study the convergence of $\alpha\beta$-KMeans. We select to present this analysis on the Myometrium cancer dataset nevertheless, the results remain consistent among the different datasets. Figure 7 illustrates the convergence of the BCD scheme discussed in Section 4 for $\alpha\beta$-KMeans with $\alpha = \beta$. Even though the objective is non-convex, it is apparent that the empirical convergence is satisfactory. We run the scheme until more than 99.9% of the clustering assignments remain unchanged between two successive clustering steps.

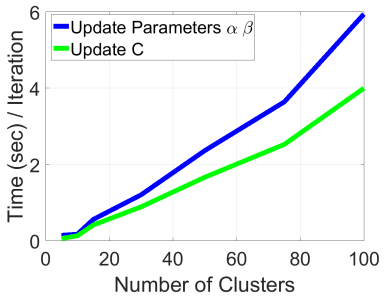
7.5. Experiments on Real Data

7.5.1 Comparisons to Variants of KMeans

Comparisons are first established against two popular variants of KMeans for SPD matrices; LE-KMeans and Karcher Means. Table 2 summarizes the experiments evaluating the performance of the $\alpha\beta$-KMeans in a pure clustering setup. The first and second columns correspond to the F1-Score achieved by LE-KMeans and Karcher Means respectively. The two proposed variants of $\alpha\beta$-KMeans are depicted in columns three ($\alpha = \beta$) and four ($\alpha \neq \beta$). For each dataset, we average our results across ten different runs to alleviate the effect of initializations. We can clearly see that the two variants of $\alpha\beta$-KMeans consistently outperform the competing schemes underlying the merits of learning the measure while clustering the data.



(a)



(b)

Figure 6. Sensitivity of $\alpha\beta$-KMeans against an increasing number of clusters in the range $[5, 100]$. The blue and red lines correspond to $\alpha\beta$-KMeans with $\alpha = \beta$ and $\alpha \neq \beta$ respectively, while the black line corresponds to LE-KMeans. (b) Time required for each iteration of updating parameters $\alpha\beta$ (blue line) and centroids (green line).

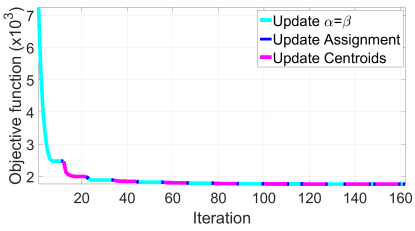


Figure 7. Convergence plot of the objective function 5 for the myometrium cancer dataset and $\alpha = \beta$. Cyan line segments correspond to iterations of updating the divergence parameters, blue segments correspond to updating the clustering assignment and magenta segments correspond to iterations of updating the centroids.

7.5.2 Bag-of-Words Nearest Neighbor Retrieval

An additional set of experiments was conducted towards evaluating the quality of the proposed clustering scheme in a Bag-of-Words (BoW), one-nearest-neighbor retrieval setup. In particular, we compute an over-partitioning of the sample space given each of the variants of KMeans such that the number of clusters is 5-10X the real number of clusters. We provide our results in a 5-fold validation format, of which 4 folds are used to compute the centroids of the over-partitioned space. Each centroid is assigned the most frequent label of the samples assigned to its cluster. For each sample in the unseen fold, we identify its nearest centroid in terms of the selected (or computed) measure and assign its label to the sample.

Dataset — Method	LE	Karcher	$\alpha\beta$-E	$\alpha\beta$-NE
VIRUS	0.248	0.254	0.252	**0.257**
BRODATZ	0.353	0.366	0.378	**0.381**
KTH TIPS	0.379	0.400	**0.429**	0.419
Prostate Cancer	0.578	0.594	**0.679**	0.660
Myometrium Cancer	0.737	0.661	0.778	**0.779**

Table 2. F1-Score based comparisons against different KMeans variants.

Table 3 aggregates our results for the 5 considered datasets averaged across the 5 folds. For each scheme, we provide an evaluation both in terms of F1-Score (14) and accuracy (ACC). We can see that for all datasets the two $\alpha\beta$-KMeans variants exceeded the performance of commonly used alternatives, once again stressing the merits of learning the appropriate measure in tandem with the clustering objective.

	LE		Karcher		$\alpha\beta$-E		$\alpha\beta$-NE	
Dataset	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**
VIRUS	55.87	0.378	56.40	0.388	**58.40**	**0.406**	53.80	0.357
BRODATZ	67.70	0.536	68.39	0.547	68.57	0.553	**68.97**	**0.555**
KTH TIPS	77.76	0.649	78.66	0.657	**81.02**	**0.693**	80.89	0.683
Prostate	79.61	0.679	80.78	0.691	**81.12**	**0.698**	80.94	0.695
Myometrium	87.92	0.792	88.10	0.795	88.31	0.796	**89.10**	**0.811**

Table 3. F1-Score and accuracy based comparisons for the BoW experimental setup.

8. Discussion

In this section, we discuss some of our observations when optimizing our objective. We found that it is essential to use a regularizer on α and β; in the absence of which, the optimization was seen to diverge, the parameters taking very large values leading to irrecoverable numerical deficiencies. As noted earlier, we found quadratic regularizers on α, β yielded good results. Exploring other forms, such as polynomials on α and β, or robust priors such as the Huber loss, is left as future work.

An analysis of the variations in similarity using $\alpha\beta$-logdet divergence is investigated for the Virus and texture datasets in [10]. A similar observation was made on the clustering objective. For our experiments on real data, we found beneficial small additive perturbations on the diagonal of the SPD matrices. On all our datasets, we found each block of updates using RCG converged in a about 5-10 steps. Surprisingly, the proposed BCD scheme is seen

to converge much faster for the $\alpha \neq \beta$-case in comparison to $\alpha = \beta$, when centroids are initialized using the LE-KMeans rather than randomly selecting samples from each dataset. This faster convergence is perhaps because of the more degrees of parameter freedom and the conditioning of the matrices.

9. Conclusions

In this work, we proposed a clustering formulation that amalgamates the problems of clustering SPD matrices and divergence learning. To achieve this, we derived and efficiently solved a clustering algorithm, termed $\alpha\beta$-KMeans, which is tasked with learning of $\alpha\beta$-logdet divergences while clustering the data. We devised an optimization scheme for efficiently solving the formulated objective, using Riemannian optimization. Finally, a diverse set of experiments was conducted on five recognition benchmarks underlining the advantages of the proposed method. Our experiments clearly demonstrated that learning the information divergence and clustering jointly led to superior accuracy in comparison to using a standard divergence on SPD matrices.

10. Acknowledgments

This material is based upon work supported by the National Science Foundation through grants #CNS-0934327, #CNS-1039741, #SMA-1028076, #CNS-1338042, #CNS-1439728, #OISE-1551059, and #CNS-1514626. AC is funded by the Australian Research Council Centre of Excellence for Robotic Vision (#CE140100016).

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009. 4

[2] S.-i. Amari and H. Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2007. 2

[3] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic resonance in medicine*, 56(2):411–421, 2006. 1, 2

[4] A. Basu, I. R. Harris, N. L. Hjort, and M. Jones. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 1998. 2

[5] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, 2002. 2

[6] D. A. Bini and B. Iannazzo. Computing the karcher mean of symmetric positive definite matrices. *Linear Algebra and its Applications*, 438(4):1700–1710, 2013. 2

[7] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 1

[8] A. Cherian, V. Morellas, and N. Papanikolopoulos. Bayesian nonparametric clustering for positive definite matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(5):862–874, 2016. 2, 6

[9] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos. Jensen-bregman logdet divergence with application to efficient similarity search for covariance matrices. *PAMI*, 35(9):2161–2174, 2013. 1, 3

[10] A. Cherian, P. Stanitsas, M. Harandi, V. Morellas, and N. Papanikolopoulos. Learning discriminative $\alpha\beta$-divergences for positive definite matrices. In *ICCV*, 2017. 2, 7

[11] A. Cichocki, S. Cruces, and S.-i. Amari. Log-determinant divergences revisited: Alpha-beta and gamma log-det divergences. *Entropy*, 17(5):2988–3034, 2015. 1, 2, 3

[12] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009. 2

[13] I. S. Dhillon and S. Sra. Generalized nonnegative matrix approximations with bregman divergences. In *NIPS*, 2005. 2

[14] A. Goh and R. Vidal. Clustering and dimensionality reduction on riemannian manifolds. In *CVPR*, 2008. 2

[15] M. Harandi and M. S. F. Porikli. Bregman divergences for infinite dimensional covariance matrices. In *CVPR*, 2014. 5

[16] M. Harandi and M. Salzmann. Riemannian coding and dictionary learning: Kernels to the rescue. In *CVPR*, 2015. 1

[17] M. T. Harandi, M. Salzmann, and R. Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices. In *ECCV*, 2014. 2

[18] G. Hinton and S. Roweis. Stochastic neighbor embedding. In *NIPS*, 2002. 2

[19] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *ICML*, 2015. 2

[20] C. Ionescu, O. Vantzos, and C. Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *ICCV*, 2015. 1

[21] R. Kompass. A generalized divergence measure for nonnegative matrix factorization. *Neural computation*, 19(3):780–791, 2007. 2

[22] B. Kulis, M. Sustik, and I. Dhillon. Learning low-rank kernel matrices. In *ICML*, 2006. 1, 3

[23] G. Kylberg, M. Uppström, K. Hedlund, G. Borgefors, and I. Sintorn. Segmentation of virus particle candidates in transmission electron microscopy images. *Journal of microscopy*, 245(2):140–147, 2012. 5

[24] J. Lafferty. Additive models, boosting, and inference for generalized divergences. In *Computational learning theory*, 1999. 2

[25] P. Mallikarjuna, A. T. Targhi, M. Fritz, E. Hayman, B. Caputo, and J.-O. Eklundh. The KTH-TIPS2 database, 2006. 5

[26] M. Mihoko and S. Eguchi. Robust blind source separation by beta divergence. *Neural computation*, 14(8):1859–1886, 2002. 2

[27] M. Moakher and P. G. Batchelor. Symmetric positive-definite matrices: From geometry to applications and visualization. In *Visualization and Processing of Tensor Fields*, pages 285–298. Springer, 2006. 3

[28] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996. 5

[29] V. M. Patel and R. Vidal. Kernel sparse subspace clustering. In *ICIP*, 2014. 2

[30] B. Pelletier. Kernel density estimation on Riemannian manifolds. *Statistics & probability letters*, 73(3):297–304, 2005. 2

[31] X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006. 1, 2, 3

[32] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000. 2

[33] R. Sivalingam, V. Morellas, D. Boley, and N. Papanikolopoulos. Metric learning for semi-supervised clustering of region covariance descriptors. In *ICDSC*, 2009. 2

[34] P. Stanitsas, A. Cherian, X. Li, A. Truskinovsky, V. Morellas, and N. Papanikolopoulos. Evaluation of feature descriptors for cancerous tissue recognition. In *ICPR*, 2016. 5

[35] P. Stanitsas, A. Cherian, A. Truskinovsky, V. Morellas, and N. Papanikolopoulos. Active convolutional neural networks for cancerous tissue recognition. In *ICIP*, 2017. 5

[36] R. Subbarao and P. Meer. Nonlinear mean shift over riemannian manifolds. *International Journal of Computer Vision*, 84(1):1–20, 2009. 2

[37] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV*, 2006. 1

[38] L. Wang, J. Zhang, L. Zhou, C. Tang, and W. Li. Beyond covariance: Feature representation with nonlinear kernel matrices. In *ICCV*, 2015. 1

[39] M. Yin, Y. Guo, J. Gao, Z. He, and S. Xie. Kernel sparse subspace clustering on symmetric positive definite manifolds. In *CVPR*, 2016. 2