Metric Learning for Semi-Supervised Clustering of Region Covariance Descriptors

Ravishankar Sivalingam Vassilios Morellas Daniel Boley Nikolaos Papanikolopoulos
University of Minnesota, Twin Cities
Minneapolis, Minnesota, USA
{ravi, morellas, boley, npapas}@cs.umn.edu

***Abstract*—In this paper we extend distance metric learning to a new class of descriptors known as Region Covariance Descriptors. Region covariances are becoming increasingly popular as features for object detection and classification over the past few years. Given a set of pairwise constraints by the user, we want to perform semi–supervised clustering of these descriptors aided by metric learning approaches. The covariance descriptors belong to the special class of symmetric positive definite (SPD) tensors, and current algorithms cannot deal with them directly without violating their positive definiteness. In our framework, the distance metric on the manifold of SPD matrices is represented as an L_2 distance in a vector space, and a Mahalanobis–type distance metric is learnt in the new space, in order to improve the performance of semi–supervised clustering of region covariances. We present results from clustering of covariance descriptors representing different human images, from single and multiple camera views. This transformation from a set of positive definite tensors to a Euclidean space paves the way for the application of many other vector-space methods to this class of descriptors.**

***Index Terms*—distance metric learning; semi–supervised clustering; region covariance descriptors; appearance clustering.**

I. Introduction

In surveillance and tracking scenarios, there is usually a supervisor manning a station of monitors displaying the feeds from multiple cameras. The system depends on the supervisor to a certain extent to perform flawless detection and tracking of different people in the image streams. It is neither feasible for the supervisor to identify and track each and every person viewed on the screen, nor is it practical to expect the system to be ideal, *i.e.* fully unsupervised with perfect accuracy, but semi–supervision is a possible option. This is especially attractive when the dataset is huge. Semi–supervised input in terms of a few labeled instances is again not always feasible, since the supervisor cannot remember every person and provide them unique labels or IDs.

The provision of pairwise constraints between data points [1], [2] is a possible way of semi–supervision, where the supervisor can specify whether or not two human images represent the same person or different people. This method of pairwise comparisons is much easier and can be done by a user even without domain knowledge. A skilled supervisor can do this very rapidly, with the exception of situations when there are heavy occlusions or illumination changes. Further, even if unique label IDs are available for a small set of images, we can always convert these instance–level constraints to pairwise constraints. Such constraints can also be generated automatically using domain–specific information.

Region covariance descriptors [3] are becoming increasingly popular as features for object detection over the past few years. Methods for fast computation of region covariances using *integral images* [4] enable the use of these compact features for many practical applications that demand real–time performance. Hu *et al.* [5] use covariance descriptors for probabilistic tracking, using particle filtering. Palaio and Batista [6] also perform multi–object tracking using region covariances and particle filters. Other local features have also been used in human detection and tracking, and [7] provides an experimental evaluation of different features, including region covariances, on benchmark datasets. In [8], Paisitkriangkrai *et al.* boost the covariance features to improve the classification accuracy. Pang *et al.* [9] use Gabor–based region covariance descriptors for face recognition. Our aim is to learn a distance metric on these region covariance descriptors to improve clustering accuracy.

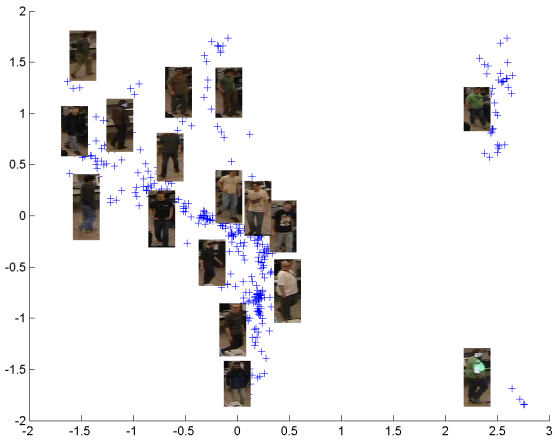


Fig. 1. ISOMAP embedding of a dataset of 367 region covariances representing 16 different people. Constructed from 40–nearest–neighborhoods. The representative image of each person is shown near the embeddings of their corresponding covariance descriptors.

There have been many distance metric learning approaches to improve the performance of nearest–neighbor classifiers [10] and clustering [11], [12]. In [13] Yang and Jin present

a Bayesian framework for distance metric learning. Schultz and Joachims [14] develop an algorithm based on the SVM training approach to learn a distance metric from pairwise comparisons. More towards computer vision, [15], [16] learn distance functions for image retrieval purposes, and [17] uses kernel–based distance metrics for the same problem. Davis *et al.* [18] provide an information–theoretic framework for distance metric learning, and also present an online algorithm with provable regret bounds. We use this metric learning algorithm in our problem domain to learn a distance metric efficiently.

Fig. 1 depicts the problem at hand. The region covariances obtained from a large dataset are embedded using ISOMAP [19] for visualization. Representative images of each person are also shown. The distribution of the descriptors is not uniform, and lie on some convoluted manifold themselves, as is often the case for human image descriptors. The motivation for this work is therefore to learn a metric that clearly "understands" the data manifold structure better, while also respecting the positive definiteness of the descriptor used.

The main contribution of this paper is the development of a framework for metric learning in the space of positive definite tensors. We extend vector–based metric learning methods to positive definite tensors, by representing the tensors as vectors in a manner such that it respects the positive definiteness of the tensors, and the learnt metric can be used as a valid transformation in the original tensor space. Further, the robust clustering of covariances from multiple cameras, based on the learnt metric, enables the efficient use of these descriptors for multi–camera detection and tracking applications. The rest of the paper is organized as follows: In the remaining part of Section I, we describe the region covariance descriptors. Section II describes the problem statement and the goal of the approach. Section III elaborates the various stages in our algorithm. Section IV presents experimental results, and Section V wraps up with the conclusions and future research directions.

A. Region Covariance Descriptors

Let each pixel in an image $\mathcal{I}$ be represented by an n–dimensional feature vector z. A given image region R is represented by a $n \times n$ covariance matrix C of the feature vectors $\{z_i\}_{i=1}^{|R|}$ of the pixels in region R. The feature vector z usually consists of color information (in some preferred color–space) and information about the first and higher order derivatives of the image intensity along the x and y directions, depending on the application intended. Although covariance matrices are positive semi–definite in general, in practice 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 $PD(n)$, the space of $n \times n$ positive definite matrices forming a connected Riemannian manifold. Given two covariance matrices C_i and C_j, the Riemannian distance metric $d(C_i, C_j)$ gives the length of the geodesic connecting these two points on this manifold [20],

$$d_{AI}(C_i, C_j) = \left\| \log\left(C_i^{-1/2} C_j C_i^{-1/2} \right) \right\|_F \qquad (1)$$

where the $\log$ represents the matrix logarithm and $\|\cdot\|_F$ is the Frobenius norm. The subscript AI represents the fact that the Riemannian metric is affine–invariant in the sense that any congruence transformation to a covariance matrix C of the form $C' = XCX^T$ by a non–singular matrix X will not affect the distance under this metric [20].

II. Problem Description

Let $\mathcal{C} = \{C_i\}_{i=1}^N, C_i \in PD(n)$ be the set of region covariance descriptors given. These may be obtained from images captured by a single camera or a collection of cameras. The user (supervisor) provides a set of pair–wise constraints in the form of *must–links*, or similarity constraints $\mathcal{S}$, and *cannot–links*, or dissimilarity constraints $\mathcal{D}$ ([1], [2]). In other words, $(C_i, C_j) \in \mathcal{S}$ implies that C_i and C_j belong to the same cluster (person), and $(C_i, C_j) \in \mathcal{D}$ implies that C_i and C_j belong to the different clusters (persons). We would like to obtain a partitioning of $\mathcal{C}$ into k clusters, $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$, which respect these constraints. Towards this end, it is desirable to learn a distance metric, from the given data and constraints, which would enhance the clustering performance over data points which are not involved in any constraint, or held–out test points.

Current distance metric learning algorithms work primarily with vector–valued data, and any tensor data is first arranged in vector form (by row–major scanning) as a pre–processing step. Vectors formed from positive definite matrices in this manner will lie on a connected cone in high–dimensional space, and any Euclidean algebra is not valid on this manifold. Our goal is to learn a distance metric in a framework which respects the positive definite property of the descriptors.

III. Approach

Our approach involves a method of vectorizing the covariance matrices so that the vectors can lie anywhere (unconstrained) in the new space. Therefore, Euclidean approaches can be applied here without any restrictions. We learn a Mahalanobis–type metric in this space and perform constrained clustering under this new metric. Since a metric in some embedding space $\phi(x)$ of data points is also a metric for the original data points x, the distance function learnt in our approach is also a well–defined metric. Fig. 2 shows the different steps in our approach. The system finally outputs the clusters of appearances, where each cluster represents separate individuals.

A. Modifying the Metric : Mapping to Euclidean Space

Our approach requires a single uniform embedding into a Euclidean space and hence, instead of using the Riemannian metric, we use another related metric in its place.

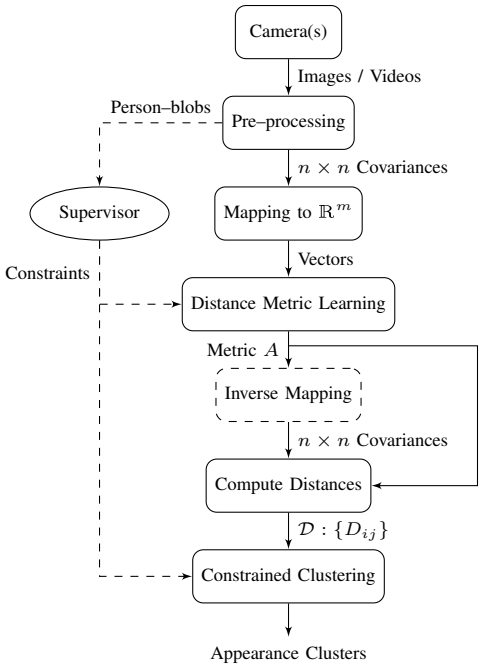


Fig. 2. Flowchart describing the steps in our approach. The user–provided constraints are used for both distance metric learning and constrained clustering. The inverse mapping from the vectors back to the $n \times n$ covariance matrices is an optional step.

The matrix logarithm is an embedding into Euclidean space which induces the Log–Euclidean distance metric [21], [22] between two positive definite matrices C_i and C_j, given by

$$d_{LE}(C_i, C_j) = \|\log C_i - \log C_j\|_F \,. \tag{2}$$

According to [23], the Log–Euclidean metric is a lower bound to the geodesic distance on the manifold, and this bound is exact when the two matrices C_i and C_j commute.

$$d_{AI}(C_i, C_j) \geq \|\log C_i - \log C_j\|_F \,. \tag{3}$$

Noting here that the Frobenius norm of a matrix is simply the L_2 norm of the vector composed of all the components of this matrix, and that the squared Mahalanobis distance $(x-y)^T A(x-y)$ is a modification of the squared L_2 distance $(x-y)^T(x-y)$, it is clear that a similar modification can be applied to the Frobenius norm. However, it is important to maintain the property of positive definiteness of the covariance matrices under the modified distance metric, *i.e.* the modified distance should also be a metric on positive definite matrices. Hence instead of simply vectorizing the elements of the $n \times n$ matrix into a vector lying in some constrained region in $\mathbb{R}^{n^2}$, we adopt a mapping to $\mathbb{R}^{n(n+1)/2}$ where the vectors are unconstrained and can occupy any region in this space. This is explained in the remainder of this section.

The matrix logarithm $L = \log C$, of a positive definite matrix C, is just a symmetric matrix. If we denote the spectrum of C as $\{\lambda_i\}_{i=1}^n$ with $\lambda_i > 0$ for $1 \leq i \leq n$ (since C is positive definite), then the spectrum of L is $\{\log \lambda_i\}_{i=1}^n$, and therefore it can have zero and even negative eigenvalues. Since this is now just a symmetric matrix with no other constraints, we can convert the upper triangular part of L into a vector $\mathbf{c}$, with appropriate scaling of the off–diagonal elements, so that $\|L\|_F = \|\mathbf{c}\|_2$ ([22]).

For example, we can convert the following 3×3 symmetric matrix L into a vector $\mathbf{c}$ as:

$$L = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{12} & L_{22} & L_{23} \\ L_{13} & L_{23} & L_{33} \end{bmatrix} \Rightarrow \mathbf{c} = \begin{bmatrix} L_{11} \\ \sqrt{2}L_{12} \\ \sqrt{2}L_{13} \\ L_{22} \\ \sqrt{2}L_{23} \\ L_{33} \end{bmatrix} . \tag{4}$$

Here it is evident that $\|L\|_F = \|\mathbf{c}\|_2$. Thus we can convert $L = \log C$ into a vector $\mathbf{c}$, containing the main diagonal elements (unscaled) and all elements above this diagonal (scaled by $\sqrt{2}$). By this conversion, we can reduce the Log–Euclidean metric of Eq. 2 to be the Euclidean distance between two vectors. If $\mathbf{a}$ and $\mathbf{b}$ are obtained from $\log A$ and $\log B$ respectively in the above manner, then the metric can be rewritten as

$$d_{LE}(A, B) = \|\mathbf{a} - \mathbf{b}\|_2 \,. \tag{5}$$

If A and B are $n \times n$ positive definite matrices, then $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$ where $m = \frac{n(n+1)}{2}$. It is important to note here that the vectors $\mathbf{a}$ and $\mathbf{b}$ are completely unconstrained, *i.e.* they can lie anywhere in $\mathbb{R}^m$, since the matrices $\log A$ and $\log B$ are just symmetric matrices with no restrictions on their eigenvalues. Thus the covariance descriptors are taken from the manifold $PD(n)$ to $\mathbb{R}^m$. Note that this is a reversible transformation.

B. Metric Learning

In the previous section, the transformation from the Riemannian manifold of positive definite matrices to an unconstrained Euclidean space was explained, which enabled us to convert the tensor covariance descriptors into vectors in $\mathbb{R}^m$. Now we will learn a distance metric on these vectors, that respects the constraints imposed by the user. Following the information–theoretic approach of Davis *et al.* [18], we learn a Mahalanobis distance function on this new set of vectors $\mathcal{C}' = \{\mathbf{c}_i\}_{i=1}^N$ using the set of similarity $\mathcal{S}$ and dissimilarity $\mathcal{D}$ constraints.

The (squared) Mahalanobis distance is parameterized by a positive definite matrix $A \in \mathbb{R}^{m \times m}$, and is given by

$$d_A^2(\mathbf{c}_i, \mathbf{c}_j) = (\mathbf{c}_i - \mathbf{c}_j)^T A (\mathbf{c}_i - \mathbf{c}_j) \,. \tag{6}$$

If $(C_i, C_j) \in \mathcal{S}$, then we would like the Mahalanobis distance between them $d_A(\mathbf{c}_i, \mathbf{c}_j) \leq u$, where u is some upper bound. If $(C_i, C_j) \in \mathcal{D}$, then we would like the Mahalanobis distance between them $d_A(\mathbf{c}_i, \mathbf{c}_j) \geq l$, where l is some lower bound. Learning this Mahalanobis metric A is formulated as a problem of minimizing the differential relative entropy between two zero–mean Gaussians, under the above–mentioned similarity and dissimilarity constraints, and involves a Bregman optimization where the LogDet divergence is minimized subject to linear constraints. Unlike previous methods for metric learning[11], this is fast, scalable, can

incorporate a prior distance function, and does not need any eigenvalue decomposition or semi–definite programming.

If A and A_0 parameterize two different Mahalanobis distance functions, then their corresponding multivariate Gaussians are

$$p(x;A) = \frac{1}{Z_A}\exp^{\left(-\frac{1}{2}d^2_A(x,\mu)\right)} \quad (7)$$

$$p(x;A_0) = \frac{1}{Z_{A_0}}\exp^{\left(-\frac{1}{2}d^2_{A_0}(x,\mu)\right)} \quad (8)$$

where Z_A, Z_{A_0} are the normalizers, and A^{-1} and A_0^{-1} denote the corresponding covariances. Given pairs of similar points $\mathcal{S}$ and dissimilar points $\mathcal{D}$, the distance metric learning problem is to minimize the KL–divergence between the two Gaussians.

$$\begin{aligned} \min_A \quad & KL(p(x;A_0)\|p(x;A)) \quad (9)\\ \text{subject to} \quad & d^2_A(x_i,x_j) \le u \qquad (x_i,x_j)\in\mathcal{S} \\ & d^2_A(x_i,x_j) \ge l \qquad (x_i,x_j)\in\mathcal{D} \end{aligned}$$

If we assume the means of the Gaussians to be the same,

$$\begin{aligned} KL(p(x;A_0)\|p(x;A)) &= \frac{1}{2}D_{ld}(A_0^{-1},A^{-1}) \\ &= \frac{1}{2}D_{ld}(A,A_0) \quad (10) \end{aligned}$$

where $D_{ld}(A,A_0)$ is the LogDet divergence [18]

$$D_{ld}(A,A_0) = tr(AA_0^{-1}) - \log\det(AA_0^{-1}) - m. \quad (11)$$

The learning is then reformulated in terms of a Bregman optimization problem, where LogDet divergence is minimized over all positive semi–definite matrices, under linear constraints.

$$\begin{aligned} \min_{A\succeq 0} \quad & D_{ld}(A_0,A) \quad (12)\\ \text{s.t.} \quad & tr(A(x_i-x_j)^T(x_i-x_j)) \le u \quad (x_i,x_j)\in\mathcal{S} \\ & tr(A(x_i-x_j)^T(x_i-x_j)) \ge l \quad (x_i,x_j)\in\mathcal{D} \end{aligned}$$

Davis *et al.* [18] efficiently solve this optimization problem, allowing slack variables ξ_{ij} in case there is no feasible solution that satisfies all the constraints. The interested reader is referred to [18] for the complete details of the algorithm used and choice of parameters. In our experiments A_0 was set to the identity matrix, and u and l to the 5^{th} and 95^{th} percentile values of the sample data distances, respectively. The advantage of this method is that it is easily extended to an online setting, and [18] provides a theoretical guarantee about the regret bounds of the online algorithm as compared to an optimal batch version of the same algorithm. Thus in our framework, the system can learn the distance metric in an online fashion, when the dataset grows and new constraints are provided by the supervisor.

As mentioned at the beginning of this section, since this Mahalanobis metric is a well–defined metric in the Euclidean space, which can be thought of as an embedding space $\phi(\cdot)$ for the covariance matrices, the forward and inverse transformations along with the Mahalanobis metric together define a proper and well–defined metric in the original space of covariance matrices as well.

Once the new Mahalanobis metric A is learnt according to the above algorithm, we compute the distance matrix $D_A^{LE}(i,j) = d^2_A(x_i,x_j) \quad \forall i,j$. This is the modified Log–Euclidean distance matrix. It is also possible to view the Mahalanobis distance as the Euclidean (L_2) distance in a transformed space, where the transformation is represented by the matrix square root of the Mahalanobis matrix, $A^{1/2}$, *i.e.* $d(x'_i,x'_j) = d(A^{1/2}x_i, A^{1/2}x_j) = d_A(x_i,x_j)$. If $\mathbf{c}$ was obtained from C following the procedure in Section III-A, this can be reversed to get back another positive definite matrix C' from $L\mathbf{c}$. From the modified set of covariance matrices, $\mathcal{C}' = \{C'_i\}_{i=1}^N$, we compute the Riemannian affine–invariant metric given by Eq. 1, to get the distance matrix D_A^{AI}. We also have the initially computed D^{AI} and D^{LE} matrices from the original affine–invariant and Log–Euclidean metrics.

C. Constrained Clustering

In the previous part, we learnt a distance metric from the given points and constraints, which is now used along with the constraints, to cluster the data points. We use the PC-KMeans (pairwise–constrained k–means) approach of Basu *et al.* [12], where the usual k–means objective function is modified to include the penalties for violating any of the given pairwise constraints. When the data points are vectors $x \in \mathbb{R}^n$, the PC-Kmeans objective function is written as

$$\begin{aligned} \mathcal{J}_{pckm} = \; & \textstyle\sum_{i=1}^N \|x_i - \mu_{l_i}\|^2 + \sum_{(x_i,x_j)\in\mathcal{S}} w_{ij}\mathbb{1}[l_i \ne l_j] \\ & + \textstyle\sum_{(x_i,x_j)\in\mathcal{D}} \bar{w}_{ij}\mathbb{1}[l_i = l_j] \quad (13) \end{aligned}$$

where l_i is the label of x_i, μ_{l_i} is the mean of all the points having label l_i, $\mathbb{1}$ is the indicator function ($\mathbb{1}[true] = 1, \mathbb{1}[false] = 0$), and w_{ij} and $\bar{w}_{ij}$ are the respective penalties for violating a *must–link* or *cannot–link* constraint between a pair (x_i,x_j).

A kernelized version of this algorithm is used here. Let the (original or modified) distance matrix D obtained from the previous step represent the squared distance in some unknown implicit embedding $\phi(x)$. This may be the one of the modified distance matrices D_A^{AI} or D_A^{LE}, or the original distances D^{AI} or D^{LE}. The kernel matrix K for this embedding can be derived from the distance matrix as

$$\begin{aligned} D_{ij} &= \{\phi(x_i) - \phi(x_j)\}\cdot\{\phi(x_i) - \phi(x_j)\} \\ &= \phi(x_i)\cdot\phi(x_i) + \phi(x_j)\cdot\phi(x_j) - 2\phi(x_i)\cdot\phi(x_j) \\ &= K_{ii} + K_{jj} - 2K_{ij}. \\ K_{ij} &= \frac{1}{2}\left(K_{ii} + K_{jj} - D_{ij}\right). \quad (14) \end{aligned}$$

The PC–Kmeans can be directly performed as a coordinate–free procedure on this kernel matrix, similar to kernel k–means [24]. The extra penalty terms in the objective function $\mathcal{J}_{pckm}$ for violation of constraints can be computed using the assigned labels. We select $w_{ij} = \bar{w}_{ij} = 1$. The number of clusters k is assumed to be known.

The initial seeding is a crucial part of any k–means algorithm, due to the fact that the procedure can get stuck at local minima. A careful choice of seeding is important to obtain good clustering results. Approaches such as [25] are specifically designed for semi–supervised clustering problems. However, we follow the *k–means++* algorithm of Arthur and Vassilvitskii [26] in order to choose the initial centers, since it performs much better in practice and comes with the theoretical guarantee that it is $\Theta(\log k)$ competitive with the optimal clustering (where k is the number of clusters). The pre–computation of distances required for seeding in this method is not an overhead in our case, since we already have the distance matrix computed beforehand.

Thus the learnt distance metric A along with the user–provided constraints are used to cluster the different covariance descriptors, and these appearance clusters correspond to distinct individuals from the camera images.

IV. Experimental Results:

The performance evaluation is presented in terms of two measures: the *Corrected Rand Index* and the pairwise *F–measure*.

The *Rand Index* [1], [11], [27] is a common measure of accuracy in the clustering literature. Each clustering of the dataset of N points is considered as a collection of $\binom{N}{2}$ pairwise decisions, as to whether a pair of points belong to the same cluster or different clusters. The Rand Index gives the agreement between the predicted labels and the ground truth.

$$RandIndex = \frac{\#\text{ correct decisions}}{\#\text{ total decisions}}. \tag{15}$$

However, the correctness of decisions for the pairs directly involved in any constraints is ensured. Therefore, following [28], [29], we instead adopt the *Corrected Rand Index*, which computes the agreement in the free decisions, *i.e.*, which are not involved in any constraints.

$$CorrectedRandIndex = \frac{\#\text{ correct free decisions}}{\#\text{ total free decisions}}. \tag{16}$$

The pairwise *F–measure*, which is based on traditional information retrieval measures such as precision and recall, is also used [12], [2], [30] to evaluate clustering accuracy.

$$Precision = \frac{\#\ PairsCorrectlyPredictedInSameCluster}{\#\ TotalPairsPredictedInSameCluster}.$$

$$Recall = \frac{\#\ PairsCorrectlyPredictedInSameCluster}{\#\ TotalPairsActuallyInSameCluster}.$$

$$\text{F-measure} = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{17}$$

We evaluate the performance of the algorithm on five different datasets, *Cam1* to *Cam5*, collected at our laboratory, varying the number of constraints provided by the user. These consist of 5×5 covariance matrices obtained from the R,G,B color channels and $x-$ and $y-$ gradients. The constraints for the experiments are obtained by random selection of pairs and determining their similarity or dissimilarity based on the ground truth. The transitive closure of the constraints [1], over both the *must–links* $\mathcal{S}$ and the *cannot–links* $\mathcal{D}$, is performed, and the resulting constraint set is used as input to the algorithm for distance metric learning and constrained clustering.

The number of clusters and number of points in each cluster is specified in Fig. 3. The *Cam4* dataset is a subset of *Cam5* containing half the number of clusters. These two datasets were collected using two camera views of the same area, and hence demonstrate the performance of the method in a multi–camera setting. The embedding of Fig. 1 is formed from a subset of descriptors from the *Cam5* dataset. The constrained clustering performed directly using the Riemannian metric is referred to as 'AI', while that performed directly using the Log–Euclidean metric is referred to as 'LE'. The same methods but with the learnt Mahalanobis distance A are referred to as 'AI+A' and 'LE+A' respectively. The recovery of the covariance matrices after the transformation by $A^{1/2}$ enables us to use the Riemannian metric on the transformed covariances as well. Fig. 4 shows the plots of Corrected Rand Index and F–measure for the *Cam1* dataset. Similarly Figs. 5 and 6 show these plots for the *Cam2* and *Cam3* datasets, respectively. Figs. 8 and 9 correspond to the datasets *Cam4* and *Cam5*.

The plots show that the distance metric learning greatly improves the clustering performance, even under a small number of user constraints. The improvement becomes more significant as more and more constraints are added, and the learnt metric clearly 'understands' the underlying data manifold structure better, as was mentioned as our motivation.

The number of constraints provided may be misleading, in relation to the number of data points, but it is important to note that these are pairwise constraints. For a dataset of N points, there are $\binom{N}{2} = \frac{N(N-1)}{2}$ possible pairs. Hence, in the *Cam1* dataset containing 94 data points, we provide results for up to 100 constraints. This number expressed as a fraction of the total possible pairs, is $\frac{100}{\binom{94}{2}} = \frac{100}{4371} = 2.28\%$. For the *Cam2* dataset, we provide a maximum of around 200 constraints, which is 6.17% of all possible pairs, and for *Cam3*, around 400 constraints forming approximately 3%. *Cam4* and *Cam5* are provided with a maximum of 3.5% and 0.93% of all pairwise constraints respectively.

In the *Cam2* dataset, the distance metric learning boosts

Dataset	K	# Data points
Cam1	2	94 $(18, 76)$
Cam2	3	81 $(17, 27, 37)$
Cam3	4	166 $(71, 21, 31, 43)$
Cam4	9	241
Cam5	18	415

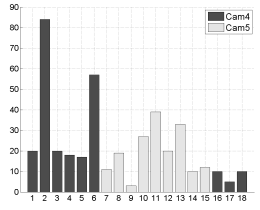
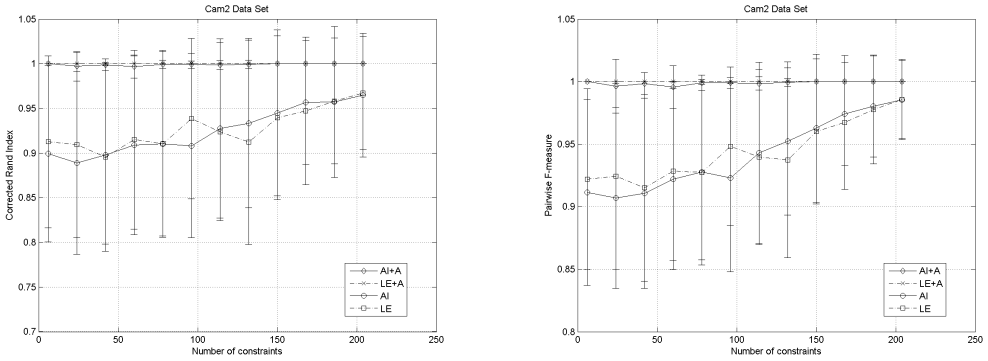
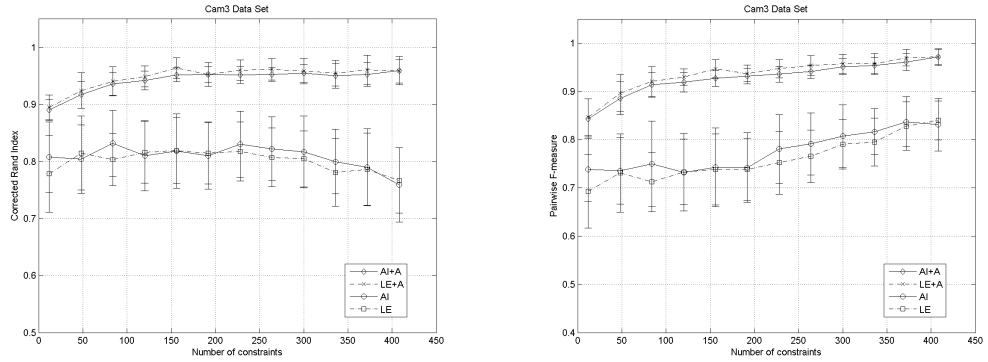


Fig. 3. Left: Datasets used where K is the number of clusters. Right: Histogram showing the distribution of points in each cluster in the *Cam5* dataset. *Cam4* is a subset of *Cam5* and is depicted in a darker shade.



(a) (b)

Fig. 5. Plot of the *Corrected Rand Index* (a) and pairwise *F–measure* (b) for the *Cam2* dataset, for different techniques. Results are averaged over 50 trials, and 1–standard–deviation bars are displayed.



(a) (b)

Fig. 6. Plot of the *Corrected Rand Index* (a) and pairwise *F–measure* (b) for the *Cam3* dataset, for different techniques. Results are averaged over 50 trials, and 1–standard–deviation bars are displayed.

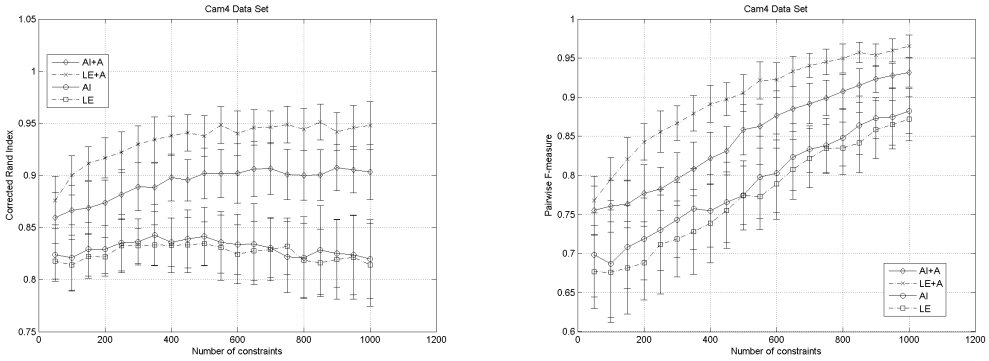
the classification accuracy to near perfect as soon as a few constraints are introduced. However, there is high variability in the constrained K–means procedure without metric learning. This can be explained as follows: Fig. 7 shows an ISOMAP [19] embedding of the data points from the *Cam2* dataset. The metric learning clearly induces a transformation on the data points where similar points are brought closer together, while dissimilar points are pushed away from each other. This embedding shows how the distance metric learning improves the separation between dissimilar points even when the number of constraints is small. Since some of the points from clusters 1 and 2 in the original space are very close, with barely any noticeable boundary, initialization by k-means$++$ is not always helpful. Even a slight offset of the initial centers will result in the merging of both those clusters into one. This results in the high variability of performance for the constrained k–means procedure without any metric learning for this dataset.

When the metric learning approach is introduced, even with a few constraints, the dissimilarity between the points in these clusters become evident, and the clusters are pushed away from each other. Hence our method produces consistent and almost perfect results. Due to the fact that there were few *must–link* constraints between the data points in cluster 2, the points in this cluster have not been pulled together sufficiently. However the *cannot–link* constraints from the other clusters, 1 and 3, towards cluster 2 ensured the clear separation of the points from different clusters.

Results from the *Cam4* and *Cam5* datasets emphasize that the framework is useful in clustering descriptors obtained from different camera views, despite any hardware–specific variability. The distance metric implicitly learns these variations and enables efficient detection and tracking of individuals across different cameras.

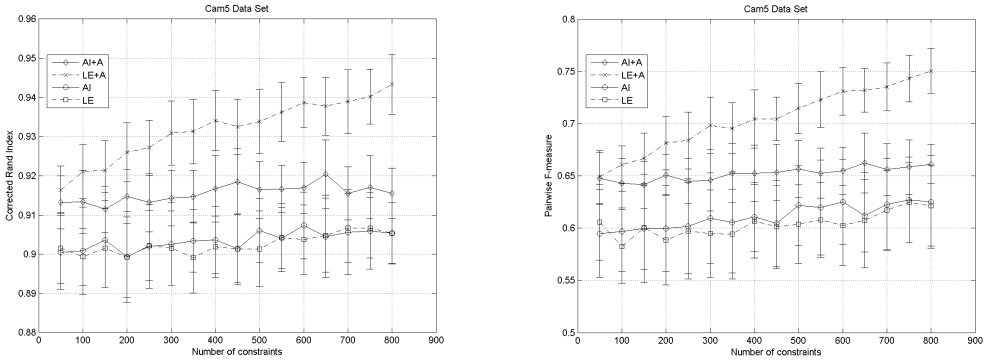
V. Conclusions and Future Work

Thus we have presented an approach to extend distance metric learning to a special class of feature descriptors known as region covariance descriptors. The covariance matrices are converted into a vector in $\mathbb{R}^m$ where the vectors are unconstrained and allowed to occupy any region in that space. This enabled the extension of vector–based distance metric learning techniques, and a global Mahalanobis distance metric is learnt in the embedding space $\mathbb{R}^m$ from the pairwise



(a) (b)

Fig. 8. Plot of the *Corrected Rand Index* (a) and pairwise *F–measure* (b) for the *Cam4* dataset, for different techniques. Results are averaged over 50 trials, and 1–standard–deviation bars are displayed.



(a) (b)

Fig. 9. Plot of the *Corrected Rand Index* (a) and pairwise *F–measure* (b) for the *Cam5* dataset, for different techniques. Results are averaged over 50 trials, and 1–standard–deviation bars are displayed.

constraints provided by the user. Constrained clustering based on the modified distance metric clearly shows significant improvement over constrained clustering using the original Riemannian metric on the data points. In a multi–camera setting, the disparities across different camera views can be learnt in a unified setting based on this framework, to aid multi–view detection and tracking applications.

This work has implications for diffusion tensor imaging (DTI), since each voxel in DTI is represented by a 3×3 positive definite tensor, which provides information about water diffusion across the voxel. Semi–supervision or constraints based on domain–specific knowledge can help in segmenting the DTI data efficiently. Further, we are currently working on learning a local distance metric in the tangent space of each data point of the manifold, based on the Riemannian affine–invariant metric. A nonlinear extension to our approach is also possible with kernel–based learning, and the online version of the metric learning algorithm can be used for incremental learning of the distance metric as more data and constraints arrive.

ACKNOWLEDGEMENTS

We are thankful to Prof. Arindam Banerjee for his thoughtful input. This work was supported by the U.S. ARMY (ARO) through contract #W911NF-08-1-0463 (Proposal 55111-CI), Johnson Controls Inc., the National Science Foundation through Grants #IIS-0219863, #IIP-0443945 and #0534286, and the Minnesota Department of Transportation.

REFERENCES

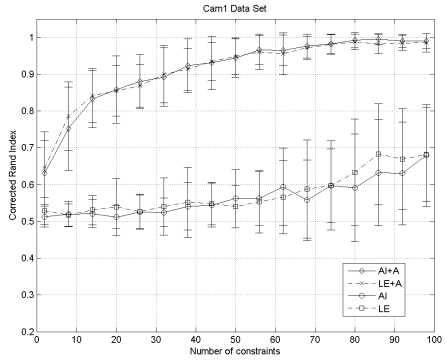
[1] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained K-means Clustering with Background Knowledge," in *Proc. of the 18th Intl. Conf. on Machine Learning (ICML '01)*, pp. 577–584, 2001.

[2] S. Basu, A. Banerjee, and R. J. Mooney, "Active Semi-supervision for Pairwise Constrained Clustering," in *Proc. of the 2004 SIAM Intl. Conf. on Data Mining (SDM '04)*, pp. 333–344, 2004.

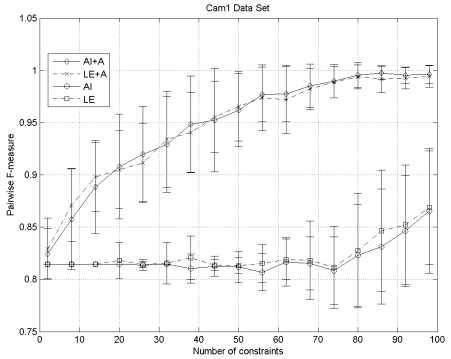
[3] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian Detection via Classification on Riemannian Manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 1713–1727, October 2008.

[4] F. Porikli and O. Tuzel, "Fast Construction of Covariance Matrices for Arbitrary Size Image Windows," in *Proc. IEEE Intl. Conf. on Image Processing*, pp. 1581–1584, October 2006.

[5] H. Hu, J. Qin, Y. Lin, and Y. Xu, "Region Covariance–based Probabilistic Tracking," in *7th World Congress on Intelligent Control and Automation (WCICA '08)*, pp. 575–580, June 2008.



(a)



(b)

Fig. 4. Plot of the *Corrected Rand Index* (a) and pairwise *F–measure* (b) for the *Cam1* dataset, for different techniques. Clearly the learnt metric A improves the clustering performance, even when the number of user–provided constraints are small. Results are averaged over 50 trials, and 1–standard–deviation bars are displayed.

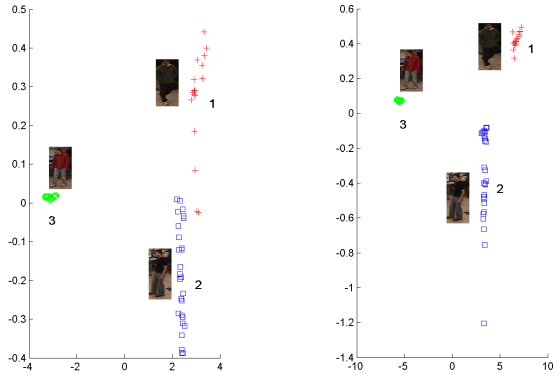


Fig. 7. ISOMAP embedding of the data points from the *Cam2* dataset. The embeddings based on the Riemannian metric (left) and based on the newly learnt distance metric (right) are shown. These were constructed using the 40–nearest–neighbors to construct a single 2–dimensional embedding of all the points. The distance metric was learnt using just 24 constraints.

[6] H. Palaio and J. Batista, "Multi-Object Tracking Using an Adaptive Transition Model Particle Filter with Region Covariance Data Association," in *Proc. 19th Intl. Conf. on Pattern Recognition (ICPR '08)*, pp. 1–4, 2008.

[7] S. Paisitkriangkrai, C. Shen, and J. Zhang, "Performance Evaluation of Local Features in Human Classification and Detection," *IET Computer Vision*, vol. 2, pp. 236–246, December 2008.

[8] S. Paisitkriangkrai, C. Shen, and J. Zhang, "Fast Pedestrian Detection Using a Cascade of Boosted Covariance Features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, pp. 1140–1151, August 2008.

[9] Y. Pang, Y. Yuan, and X. Li, "Gabor-Based Region Covariance Matrices for Face Recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, pp. 989–993, July 2008.

[10] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," in *Adv. in Neural Inf. Proc. Sys. (NIPS)*, pp. 1473–1480, MIT Press, 2006.

[11] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell, "Distance Metric Learning with Application to Clustering with Side-Information," in *Adv. in Neural Inf. Proc. Sys. (NIPS)*, pp. 505–512, MIT Press, 2002.

[12] S. Basu, M. Bilenko, and R. J. Mooney, "Comparing and Unifying Search-Based and Similarity-Based Approaches to Semi-Supervised Clustering," in *Proc. of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pp. 42–49, August 2003.

[13] L. Yang, R. Jin, and R. Sukthankar, "Bayesian Active Distance Metric Learning," in *23rd Conf. on Uncertainty in Artificial Intelligence (UAI '07)*, July 2007.

[14] M. Schultz and T. Joachims, "Learning a Distance Metric from Relative Comparisons," in *Adv. in Neural Inf. Proc. Sys. (NIPS)*, MIT Press, 2003.

[15] T. Hertz, A. Bar-Hillel, and D. Weinshall, "Learning Distance Functions for Image Retrieval," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. 570–577, June-2 July 2004.

[16] A. Frome, F. Sha, Y. Singer, and J. Malik, "Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification," in *Proc. 11th IEEE Intl. Conf. Computer Vision (ICCV '07)*, pp. 1–8, October 2007.

[17] H. Chang and D.-Y. Yeung, "Kernel-based Distance Metric Learning for Content-Based Image Retrieval," *Image Vision Comput.*, vol. 25, no. 5, pp. 695–703, 2007.

[18] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-Theoretic Metric Learning," in *Proc. of the 24th Intl. Conf. on Machine Learning (ICML '07)*, pp. 209–216, 2007.

[19] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[20] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian Framework for Tensor Computing," *Int'l J. Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006.

[21] A. Goh and R. Vidal, "Segmenting Fiber Bundles in Diffusion Tensor Images," in *Proc. of the 10th European Conf. on Computer Vision (ECCV '08)*, pp. 238–250, 2008.

[22] 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, pp. 411–421, August 2006.

[23] R. Bhatia, *Positive Definite Matrices*. Princeton Series in Applied Mathematics, Princeton, NJ, USA: Princeton University Press, 2007.

[24] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel K-means: Spectral Clustering and Normalized Cuts," in *Proc. of the 10th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD '04)*, pp. 551–556, 2004.

[25] S. Basu, A. Banerjee, and R. J. Mooney, "Semi-Supervised Clustering by Seeding," in *Proc. of the 19th Intl. Conf. on Machine Learning (ICML '02)*, pp. 27–34, 2002.

[26] D. Arthur and S. Vassilvitskii, "K-means++: The Advantages of Careful Seeding," in *Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '07)*, pp. 1027–1035, 2007.

[27] W. M. Rand, "Objective Criteria for the Evaluation of Clustering Methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[28] K. Wagstaff and C. Cardie, "Clustering with Instance-Level Constraints," in *Proc. of the 17th Intl. Conf. on Machine Learning (ICML '00)*, pp. 1103–1110, 2000.

[29] D. Klein, S. D. Kamvar, and C. D. Manning, "From Instance-Level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering," in *Proc. of the 19th Intl. Conf. on Machine Learning (ICML '02)*, 2002.

[30] S. Basu, "A Comparison of Inference Techniques for Semi-supervised Clustering with Hidden Markov Random Fields," in *Proc. of the ICML 2004 Workshop on Statistical Relational Learning and its Connections to Other Fields (SRL '04)*, 2004.