Semi-Supervised Discriminative Common Vector Method for Computer Vision Applications

Hakan Cevikalp

Electrical and Electronics Engineering Department of Eskisehir Osmangazi University, Machine Learning and Computer Vision Laboratory, Meselik 26480 Eskisehir, Turkey.

Abstract

We introduce a new algorithm for distance metric learning which uses pairwise similarity (equivalence) and dissimilarity constraints. The method is adapted to the high-dimensional feature spaces that occur in many computer vision applications. It first projects the data onto the subspace orthogonal to the linear span of the difference vectors of the similar sample pairs. Similar samples thus have identical projections, i.e., the distance between the two elements of each similar sample pair becomes zero in the projected space. In the projected space we find a linear embedding that maximizes the scatter of the dissimilar sample pairs. This corresponds to a pseudo-metric characterized by a positive semi-definite matrix in the original input space. We also kernelize the method and show that this allows it to handle cases with low-dimensional input spaces and large numbers of similarity constraints. Despite the method's simplicity, experiments on synthetic problems and on real-world image retrieval, visual object classification, gender classification and image segmentation ones demonstrate its effectiveness, yielding significant improvements over existing distance metric learning methods.

Keywords: Semi-supervised distance learning, classification, clustering, image retrieval, similarity constraints, discriminative common vector.

1. Introduction

In a wide range of computer vision problems including object classification, segmentation and image and video retrieval, the performance depends critically on

Email address: hakan.cevikalp@gmail.com (Hakan Cevikalp)

the similarity metric used to compare examples, so it is important to develop effective methods for learning distance metrics for such applications [1, 2, 3, 4, 5]. Measuring distances is comparatively simple when the features used are hand-chosen to be independent and highly relevant, but in computer vision applications with modern feature sets there are typically a great many features, many of which are either highly correlated with other features or irrelevant for the task being considered. This happens because at present, despite their redundancy, large and comparatively generic modern feature sets typically give better performance than smaller hand-chosen ones, and because vision problems are often somewhat open with the most relevant features depending on the exact problem and dataset being considered. (*E.g.*, when organizing image collections, it is possible to group images in many different ways, based on objects that they contain, natural versus built, outdoors versus indoors, *etc*.) When there are irrelevant and/or correlated features, similarity judgements based on Euclidean feature space distances often give unacceptable results, so it is necessary to learn more discriminative distance metrics.

In the simplest forms of distance learning, explicit class labels are supplied for the training samples, thus establishing a global notion of the similarity that is to be learned. However there are many applications in which explicit labels are not available, either because the underlying problem does not involve classes or involves only poorly-defined ones, or owing to the high cost of supplying a full labeling. In such cases, side information in the form of categorical similar / dissimilar judgements linking pairs of examples may still be available at a reasonable cost. For example in surveillance applications such as [4], objects (*e.g.* faces) extracted at roughly the same location in successive video frames can be assumed to represent the same individual, whereas ones extracted at different locations in the same frame must represent different individuals. In some applications such as relevance-feedback based image retrieval [6] or interactive semi-supervised image segmentation [2], such similarity judgements are actually the most natural form of supervision.

This paper focuses on distance metric learning from similarity judgements of this kind. Our strategy is to handle the similarity constraints first by projecting the data to a lower-dimensional subspace in which each similar pair becomes an identical pair, and then to address the dissimilarity constraints by finding a linear embedding that maximizes the distances between the projected dissimilar pairs. There are several adavantages of this procedure: Projection onto the null space is the optimal linear projection in the sense that it preserves the variance along the orthogonal directions to the projection direction, hence the original dis-

tance measure is best preserved. Moreover, as the experiments show, the resulting method is particularly suitable for computer vision problems based on modern high-dimensional feature sets since one does not need to approximate complex distance model parameters.

2. Related Work

In recent years there has been a growing interest in methods for learning distance metrics due to their broad applications. Some of these approaches find the desired distance function directly, while others find embeddings in which the Euclidean distance serves as the new distance function. The two problems are equivalent and we will present them interchangeably here. We only discuss methods based on similarity judgments: ones that require explicit class labels [7, 8, 9, 10] will not be considered here. A more comprehensive survey of distance metric learning techniques can be found in [11].

Similarity judgement based distance learning methods modify their input distances to accommodate the given pairwise constraints, and at present most of them focus on learning linear Mahalanobis-like distances parameterized by positive-definite or semi-definite matrices. Xing et al. [12] used a convex programming formulation under equivalence constraints to learn a full-rank Mahalanobis metric. The metric is learned via an iterative procedure that involves projection and eigen-decomposition in each step. Tsang & Kwok [13] formulated the problem as a quadratic optimization one. They also extend their method to the nonlinear case using the kernel trick. Shalew-Shwartz et al. [14] proposed a sophisticated online distance metric learning algorithm that uses side information. The method incorporates the large margin concept, and the distance metric is modified based on two successive projections involving an eigen-decomposition. Davis et al. [15] proposed an information-theoretic approach to learn a Mahalanobis distance function. They formulated the metric learning problem as that of minimizing the differential entropy between two multivariate Gaussians under equivalence constraints on the distance function. Yang et al. [11] proposed a Bayesian framework that estimates a posterior distribution for the distance metric from the pairwise constraints. All of the above algorithms attempt to learn full-rank distance metrics. This makes them less suitable for high-dimensional computer vision problems, in which it is usually more effective to learn lower-rank distance metrics or embeddings. To this end, Cevikalp & Paredes [2] introduce a low-rank distance metric learning algorithm based on sigmoid functions. A similar weakly-supervised method was introduced in [16]. A semi-supervised low-rank Mahalanobis distance learning

algorithm for high-dimensional spaces using log-determinant matrix divergence was introduced in [17]. More recently, a sparse (low-rank) metric learning method using Nesterov's smooth optimization has been proposed for high-dimensional data [18]. Unlike other methods, the sparse metric learning algorithm uses relative comparisons (given in terms of triplets) instead of pair-wise equivalence constraints and the authors showed that it outperforms competing methods. However, they reported results on relatively small-dimensional data sets selected from UCI repository rather than on challenging high-dimensional real-world datasets. Another sparse metric learning method using alternating linearization optimization has been proposed in [19].

Graph-based methods that incorporate pairwise side information by modifying the weights of the graph have also been proposed [20, 21, 22, 23, 24]. Their major limitation is that they assume that local nearest-neighbor samples typically have the same class label (*c.f.* local neighborhood-based nonlinear dimensionality reduction methods in which each class is modeled as a manifold that is locally close to linear). This is only true if the classes are sampled densely relative to the inter-class spacing, which is hard to achieve with feasible training set sizes in high dimensional problems with difficult-to-distinguish classes. As a result, the graph-based approaches tend to perform poorly in practical vision problems because the constraints that they assume become too noisy. To alleviate this problem, some authors [25, 6] use multiple graphs which operate on different feature sets. Then, they learn more reliable distance metrics by fusing those graphs with different techniques.

A method that is more closely related to ours is Relevant Components Analysis (RCA) [26]. It searches for an embedding that assigns large weights to the most relevant dimensions and lower weights to less relevant ones, where relevance is estimated using the pairwise similarity constraints. RCA does not incorporate dissimilarity constraints and it is restricted to learning linear transformations in the original input space. Tsang et al. [27] improved RCA and kernelized it. Hoi et al. [5] proposed Discriminative Components Analysis (DCA), a method that allows dissimilarity constraints to be incorporated into RCA and Kernel RCA. Our approach is similar in spirit to DCA, but it overcomes a serious drawback of DCA (see section 2.3).

Finally, there are some hybrid methods that unify clustering and metric learning into a common framework based on side information [28, 29]. Among these, [29] is worth mentioning because it projects onto the null space of the similarity constraints as we do.

2.1. Metric Learning Under Side Constraints

Before presenting our method in its linear and kernelized forms, we summarize the setting for distance metric learning under side constraints and sketch the RCA and DCA approaches.

Let $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, N$ denote the samples of the training set. We are given a set of side constraints in the form of similar and dissimilar pairs and we aim to find a pseudo-metric that reflects the underlying relationships imposed by them. We focus on pseudo-metrics of the form

$$d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A} (\mathbf{x}_i - \mathbf{x}_j)}, \tag{1}$$

where $\mathbf{A} \geq \mathbf{0}$ is a symmetric positive semi-definite matrix. Equivalently, if $q = \mathrm{Rank}(\mathbf{A}) \leq d$, $\mathbf{A}$ can be written in the form $\mathbf{A} = \mathbf{W}\mathbf{W}^\top$ where $\mathbf{W}$ is a full-rank rectangular matrix of size $d \times q$, so that

$$\|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 = \|\mathbf{W}^\top \mathbf{x}_i - \mathbf{W}^\top \mathbf{x}_j\|^2, \tag{2}$$

i.e. distances between points under the metric $\mathbf{A}$ are equivalent to Euclidean distances on their linear projections by $\mathbf{W}^\top$.

2.2. Relevant Component Analysis (RCA)

The basic strategy of RCA is to identify irrelevant dimensions and reduce their effects by assigning lower weights to them. RCA does not exploit dissimilarity information. Similarity information is provided in the form of "chunklets": groups of two or more data samples that are considered "similar" (*e.g*. that belong to the same class). Assume that we are given C chunklets with chunklet c containing n_c patterns $\{\mathbf{x}_{c,1}, ..., \mathbf{x}_{c,n_c}\}$. RCA centers each chunklet then finds their combined covariance matrix:

$$\mathbf{S} \;=\; \frac{1}{n} \sum_{c=1}^{C} \sum_{i=1}^{n_c} (\mathbf{x}_{c,i} - \boldsymbol{\mu}_c)(\mathbf{x}_{c,i} - \boldsymbol{\mu}_c)^\top. \tag{3}$$

Here, $\boldsymbol{\mu}_c$ is the mean of chunklet c and $n = \sum_{c=1}^{C} n_c$. The linear embedding of RCA is then computed as $\mathbf{W} = \mathbf{S}^{-1/2}$, so that the Mahalanobis distance matrix becomes $\mathbf{A} = \mathbf{S}^{-1}$. Note that this is singular unless the covariance has full-rank, which certainly requires $n \geq d + C$ – something that is not possible in many high-dimensional problems. Details of the kernelization of RCA can be found in [27].

It should be noted RCA attempts to learn a full-rank distance matrix in the original input space. Therefore the number of parameters to be estimated is the square of the dimensionality and we have only limited number of similarity constraints in most of the real-world applications. As a result, learning an effective full-rank distance matrix in high-dimensional spaces is impracticable by RCA (The covariance matrix becomes singular and hence dimensionality reduction must be applied before RCA. But this may cause loss of important relevant information)

2.3. Discriminative Component Analysis (DCA)

DCA can be seen as a weakly-supervised variant of classical direct Linear Discriminant Analysis [30]. DCA assumes that dissimilarity constraints are also supplied between some of the chunklets. Specifically, each chunklet c has a dissimilarity set D_c whose elements are the chunklets (from the original C) that are flagged as being dissimilar to c. The within-chunklet and between-chunklet scatter matrices are then computed as follows

$$\begin{aligned} \mathbf{S}_b &= \frac{1}{m}\sum_{c=1}^{C}\sum_{i\in D_c}(\boldsymbol{\mu}_c-\boldsymbol{\mu}_i)(\boldsymbol{\mu}_c-\boldsymbol{\mu}_i)^\top, \\ \mathbf{S}_w &= \frac{1}{C}\sum_{c=1}^{C}\frac{1}{n_c}\sum_{i=1}^{n_c}(\mathbf{x}_{c,i}-\boldsymbol{\mu}_c)(\mathbf{x}_{c,i}-\boldsymbol{\mu}_c)^\top, \end{aligned} \tag{4}$$

where $m=\sum_{c=1}^{C}|D_c|$ is the total number of dissimilarity set entries. As in LDA, DCA finds a set of orthogonal linear projection directions $\mathbf{w}$ that maximize the ratio of the between-class and within-class scatters $J(\mathbf{w})=\frac{\mathbf{w}^\top\mathbf{S}_b\mathbf{w}}{\mathbf{w}^\top\mathbf{S}_w\mathbf{w}}$, assembling $\mathbf{W}$ column-wise from these. If $\mathbf{S}_w$ is nonsingular, the maximum eigenvectors of $\mathbf{S}_w^{-1/2}\mathbf{S}_b\mathbf{S}_w^{-1/2}$ provide the solution. Since the matrix is typically non-symmetric, its eigen-decomposition may be unstable. To circumvent this problem, the simultaneous diagonalization algorithm is often employed.

As with LDA and RCA, the above method can only be applied when the within-chunklet scatter matrix $\mathbf{S}_w$ has full-rank. Otherwise, the dimensionality must be reduced to avoid singularity, and as in RCA this can cause a considerable loss of discriminative power unless it is done properly. Unfortunately, DCA takes a very suboptimal route at this point. Instead of following the traditional LDA approach, it first projects the data onto the range space of $\mathbf{S}_b$ under the assumption that the null space of $\mathbf{S}_b$ does not contain any relevant information. This is seldom correct, as illustrated in Fig. 1. For the two dissimilar chunklets shown in

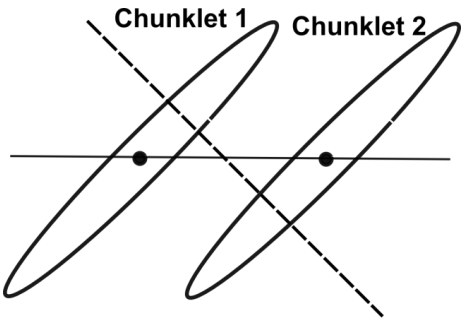

Figure 1: DCA chooses the line connecting the chunklet means as the optimal projection direction, whereas the best separating direction is the dashed line, which can be found by following the classical LDA method.

the figure, DCA chooses the line connecting their centers as the optimal projection vector: this is obviously a very suboptimal direction for separating them. See [31, 32] for more details on this.

3. Proposed Method: SS-DCV

DCA method aims to maximize the classical LDA function criterion, $J(\mathbf{w}) = \frac{\mathbf{w}^\top \mathbf{S}_b \mathbf{w}}{\mathbf{w}^\top \mathbf{S}_w \mathbf{w}}$. However, this criterion is not appropriate since the maximization does not have a unique solution when the dimensionality of the sample space is larger than the number of similar sample pairs (this yields a rank deficient within-chunklet matrix $\mathbf{S}_w$). In this case, every projection vector $\mathbf{w}$ (and projection matrix $\mathbf{W}$ whose columns includes these vectors) such that $\mathbf{w}^\top \mathbf{S}_w \mathbf{w} = 0$ and $\mathbf{w}^\top \mathbf{S}_b \mathbf{w} \neq 0$ maximizes this criterion. Note that if $\mathbf{S}_w$ is singular, which is typically the case for high-dimensional computer vision problems, one can create many projection matrices by using combinations of projection vectors coming from the null space of $\mathbf{S}_w$. But these are not necessarily the optimal projection directions. On the other hand, as we have shown in [31], the following *null space based LDA* criterion has a unique maximum for the projection vectors with unit length and it also maximizes the LDA criterion

$$J(\mathbf{W}) = \max_{|\mathbf{W}^\top \mathbf{S}_w \mathbf{W}|=0} \mathrm{Trace}(\mathbf{W}^\top \mathbf{S}_b \mathbf{W}) \tag{5}$$

Our proposed method, Semi-Supervised Discriminative Common Vectors (SS-DCV), works in the orthogonal complement of the span of the difference vectors of the similar pairs (*i.e.* the null space of the scatter matrix $\mathbf{S}_S$ of these

pairs), finding a linear embedding that maximizes the scatter $\mathbf{S}_D$ of the dissimilar pairs (*i.e*. the analog of LDA's between class scatter matrix) within this subspace. Specifically, we maximize the null spaced based LDA criterion $J(\mathbf{W}) = \mathrm{Trace}(\mathbf{W}^\top \mathbf{S}_D \mathbf{W})$ under the constraints that the columns of $\mathbf{W}$ are orthonormal and that $\mathbf{W}^\top \mathbf{S}_S \mathbf{W} = \mathbf{0}$ by using constraints rather than class labels. This can be done either by finding an orthonormal basis for the span of $\mathbf{S}_S$ and projecting $\mathbf{S}_D$ orthogonal to it, or by finding an orthonormal basis for the null space of $\mathbf{S}_S$ and projecting $\mathbf{S}_D$ onto it, followed by estimation of the leading eigenvectors of the resulting matrix. The method can also be viewed as a weakly-supervised variant of the well-known Discriminative Common Vector (DCV) method we introduced in [31, 33], which is known to give good results, *e.g*., in high-dimensional face recognition problems.

Specifically, if we are given sets of similar and dissimilar sample pairs, let $\mathbf{X}_S$ and $\mathbf{X}_D$ be matrices whose columns are the difference vectors of the (respectively) similar and dissimilar pairs

$$\begin{aligned} \mathbf{X}_S &= \left[\mathbf{x}_{s1,1} - \mathbf{x}_{s1,2},\ \mathbf{x}_{s2,1} - \mathbf{x}_{s2,2},\ \ldots\ \mathbf{x}_{sn,1} - \mathbf{x}_{sn,2}\right], \\ \mathbf{X}_D &= \left[\mathbf{x}_{d1,1} - \mathbf{x}_{d1,2},\ \mathbf{x}_{d2,1} - \mathbf{x}_{d2,2},\ \ldots\ \mathbf{x}_{dm,1} - \mathbf{x}_{dm,2}\right]. \end{aligned} \tag{6}$$

where $\mathbf{x}_{si,1}$ and $\mathbf{x}_{si,2}$ respectively denote the first and second samples of the i-th similar sample pair; $\mathbf{x}_{di,1}$ and $\mathbf{x}_{di,2}$ respectively denote the first and second samples of the i-th dissimilar sample pair. The corresponding scatter matrices are $\mathbf{S}_S = \mathbf{X}_S \mathbf{X}_S^\top$ and $\mathbf{S}_D = \mathbf{X}_D \mathbf{X}_D^\top$. Similarly, if we are given chunklets rather than pairs, $\mathbf{X}_S$ is constructed by subtracting the chunklet samples from their associated means, and $\mathbf{X}_D$ by subtracting the corresponding chunklet means as in (4).

Let $\mathbf{U}$ be a basis for the span of $\mathbf{X}_S$, *i.e*. a matrix whose columns are a minimal orthonormal basis for the columns of $\mathbf{X}_S$. $\mathbf{U}$ can be found by truncated SVD of $\mathbf{X}_S$ or eigendecomposition of $\mathbf{S}_S$ in the noisy case, or by QR or Gram-Schmidt decomposition of $\mathbf{X}_S$ in the noise-free case. Then $\mathbf{P} = \mathbf{I} - \mathbf{U}\mathbf{U}^\top$ (or equivalently $\mathbf{P} = \mathbf{I} - \mathbf{X}_S(\mathbf{X}_S^\top \mathbf{X}_S)^+ \mathbf{X}_S^\top$ where $()^+$ denotes the pseudo-inverse and $\mathbf{I}$ denotes the identity matrix) zeroes out components along the span of $\mathbf{X}_S$ and hence implicitly implements projection onto the orthogonal complement of the span. The samples $\mathbf{x}$ can be projected onto this subspace using $\mathbf{P}\mathbf{x} = \mathbf{x} - \mathbf{U}\mathbf{U}^\top \mathbf{x}$ before calculating $\mathbf{X}_D$ and hence $\mathbf{S}_D$ and its eigendecomposition. For each similar sample pair or chunklet, each point of it projects to a fixed point that depends only on the pair or chunklet (called the common vector in the supervised DCV method). Differences within similar pairs or chunklets are thus projected away. The process is similar to the method of metric learning by collapsing classes [7] in the sense that it naturally gives the optimal solution for this in high-dimensional spaces. Conversely,

it is complementary to RCA in the sense that it gives distances associated with the null space of the scatter of the similar sample pairs, whereas RCA gives distances associated with the range space. Moreover, projection onto the null space is the optimal linear projection in the sense that it preserves the variance along the orthogonal directions to the projection direction, hence the original distance measure is best preserved [29].

Given the projection $\mathbf{P}$, we estimate $\mathbf{W}$ by finding the leading singular values of the projected dissimilarity matrix $\widetilde{\mathbf{X}}_D = \mathbf{P}\mathbf{X}_D$, or equivalently the leading eigenvalues of the projected dissimilarity scatter matrix $\widetilde{\mathbf{S}}_D = \mathbf{P}\mathbf{X}_D\mathbf{X}_D^\top\mathbf{P}$. The number of eigenvectors l can be chosen based on the energy of the eigenvalues – below we choose l so that the sum of the retained eigenvalues is at least 98% of the sum of all of them, *i.e*. the trace of $\widetilde{\mathbf{S}}_D$. Given $\mathbf{W}$, the final distance metric is $\mathbf{A} = \mathbf{W}\mathbf{W}^\top$.

The full method can thus be summarized as follows:

Step 1: Compute $\mathbf{X}_S$ and find its orthonormal basis matrix $\mathbf{U}$.

Step 2: Project the dissimilar sample pairs to the null space of $\mathbf{X}_S$ using $\widetilde{\mathbf{X}}_D = (\mathbf{I} - \mathbf{U}\mathbf{U}^\top)\mathbf{X}_D$.

Step 3: Compute $\widetilde{\mathbf{S}}_D = \widetilde{\mathbf{X}}_D\widetilde{\mathbf{X}}_D^\top$, find its leading eigenvectors $\mathbf{W}$, and output the final distance metric $\mathbf{A} = \mathbf{W}\mathbf{W}^\top$.

Note that if the dimension of $\mathbf{U}$ is more than about half of the dimension of the feature space, it may be more efficient to find an explicit basis $\mathbf{V}$ for the null space of $\mathbf{X}_S$ and project $\mathbf{X}_D$ directly using this: $\widetilde{\mathbf{X}}_D = \mathbf{V}^\top\mathbf{X}_D$, returning $\mathbf{V}\mathbf{W}$ in place of $\mathbf{W}$.

3.1. Geometric Interpretation

The proposed method corresponds to approximating each pair or chunklet with an affine subspace spanned by all of the directions in $\mathbf{U}$, then finding projection directions that maximize the geometric distances (scatter) between these subspaces. In this way, each chunklet is enlarged by creating new points based on all possible variation cues coming from all similarity constraints. Because the same basis $\mathbf{U}$ is used for all of the subspaces, the subspaces themselves are quite large and they differ only in their locations within the orthogonal complement of $\mathbf{U}$. When the method is used to classify a new sample, the resulting output distance is effectively a reweighted form of the underlying sample-to-affine-subspace distances. In contrast, RCA method approximates each class with an hyperellipsoid

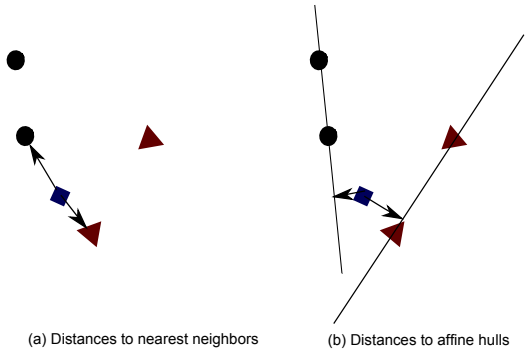

Figure 2: Comparisons of distances to the nearest neighbors and affine hulls. Affine hulls are the lines passing through two samples. The closest distance from a query (shown with blue diamond symbol) to an affine hull is the norm of displacement from the query to the closest point on the hull. Observe how the distances change by each method.

based on the variations coming from all similarity constraints as in the proposed method. But, an affine hull model is a much better approximation than an hyperellipsoid in high-dimensional spaces since the amount of geometric details that can be resolved usually decrease rapidly as the dimensionality increases and there is not enough sample to approximate the covariance matrices associated to the hyperellipsoids.

In image retrieval applications with similarity constraints, the proposed method builds a manifold (equivalent to an affine subspace) by using similar images to the query image. Then, the retrieved images are ranked based on the minimum Euclidean distances to this approximated manifold rather than distances to the unique query sample (see Fig. 2 for comparisons of distances to a nearest neighbor or to a nearest affine hull). In this context, the proposed method has close ties with the methods introduced in [34, 35]. As in our case, those methods also approximate samples with different type of manifolds for improving the classification performance of nearest-neighbor search. Especially, the tangent distance, which enlarges the training samples based on small spatial transformations and variations of the thickness of pen strokes, is still considered as state-of-the-arts for hand-written digit classification. Approximation of samples with different type of manifolds has been widely used for improving the classification performance of nearest-neighbor search in the literature, and in this set up, the proposed method is similar to these methods.

3.2. Kernelization of the Method

The linear SS-DCV method above breaks down when the orthogonal complement onto which the data is projected becomes too small, *i.e*. when the rank of $\mathbf{X}_S$ (the dimension of $\mathbf{U}$) approaches the effective dimension of the input data. In particular, this is likely to happen when the class covariances span the whole space and there are many more similarity pairs than input dimensions. In such cases RCA may be more effective. However kernelization can also extend the working range of SS-DCV in such cases, and more generally it allows problems with strong nonlinearities to be handled.

Let $\phi(.)$ be the implicit feature space embedding and $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ be the corresponding kernel function, where $\langle \cdot \rangle$ denotes the feature space inner product. The implicit matrices of difference vectors of similar and dissimilar pairs in the mapped feature space become

$$\begin{aligned} \mathbf{\Phi}_S &= [\phi(\mathbf{x}_{s1,1}) - \phi(\mathbf{x}_{s1,2}), \ \ldots \ \phi(\mathbf{x}_{sn,1}) - \phi(\mathbf{x}_{sn,2})], \\ \mathbf{\Phi}_D &= [\phi(\mathbf{x}_{d1,1}) - \phi(\mathbf{x}_{d1,2}), \ \ldots \ \phi(\mathbf{x}_{dm,1}) - \phi(\mathbf{x}_{dm,2})]. \end{aligned} \tag{7}$$

The orthogonal projection onto the null space of $\mathbf{\Phi}_S$ is then

$$\mathbf{P}_\Phi = \mathbf{I}_\Phi - \mathbf{\Phi}_S (\mathbf{\Phi}_S^\top \mathbf{\Phi}_S)^+ \mathbf{\Phi}_S^\top, \tag{8}$$

where $\mathbf{I}_\Phi$ is the identity matrix in Φ space. Note that $\mathbf{K}_S = \mathbf{\Phi}_S^\top \mathbf{\Phi}_S$ is an explicit matrix with entries $k(\mathbf{x}_{si,1}, \mathbf{x}_{sj,1}) - k(\mathbf{x}_{si,1}, \mathbf{x}_{sj,2}) - k(\mathbf{x}_{si,2}, \mathbf{x}_{sj,1}) + k(\mathbf{x}_{si,2}, \mathbf{x}_{sj,2})$, and that $\mathbf{K}_D = \mathbf{\Phi}_D^\top \mathbf{\Phi}_D$ and $\mathbf{K}_{SD} = \mathbf{\Phi}_S^\top \mathbf{\Phi}_D$ can be defined similarly. Given a new sample $\mathbf{x}$, the projection of $\phi(\mathbf{x})$ onto the null space is

$$\widetilde{\phi}(\mathbf{x}) = \mathbf{P}_\Phi \, \phi(\mathbf{x}) = \phi(\mathbf{x}) - \mathbf{\Phi}_S \mathbf{K}_S^+ \mathbf{\Phi}_S^\top \phi(\mathbf{x}). \tag{9}$$

Let $\widetilde{\mathbf{\Phi}}_D$ denote the projection of the matrix of dissimilar sample pairs onto this null space. The corresponding scatter matrix becomes

$$\widetilde{\mathbf{S}}_D = \widetilde{\mathbf{\Phi}}_D \widetilde{\mathbf{\Phi}}_D^\top = \mathbf{P}_\Phi \mathbf{\Phi}_D \mathbf{\Phi}_D^\top \mathbf{P}_\Phi. \tag{10}$$

We need to find the leading eigenvectors of this. All eigenvectors $\mathbf{v}$ corresponding to nonzero eigenvalues lie in the span of the dissimilar sample pairs, *i.e*. $\mathbf{v} = \widetilde{\mathbf{\Phi}}_D \boldsymbol{\alpha}$ for some $\boldsymbol{\alpha}$. The above equation can thus be written as

$$\lambda \, \widetilde{\mathbf{\Phi}}_D \, \boldsymbol{\alpha} = \widetilde{\mathbf{\Phi}}_D \, \widetilde{\mathbf{\Phi}}_D^\top \, \widetilde{\mathbf{\Phi}}_D \, \boldsymbol{\alpha}. \tag{11}$$

Multiplying by $\widetilde{\mathbf{\Phi}}_D^\top$ on left and denoting $\widetilde{\mathbf{\Phi}}_D^\top \widetilde{\mathbf{\Phi}}_D$ by $\widetilde{\mathbf{K}}_D$, we obtain

$$\lambda \widetilde{\mathbf{\Phi}}_D^\top \widetilde{\mathbf{\Phi}}_D \boldsymbol{\alpha} = \widetilde{\mathbf{\Phi}}_D^\top \widetilde{\mathbf{\Phi}}_D \widetilde{\mathbf{\Phi}}_D^\top \widetilde{\mathbf{\Phi}}_D \boldsymbol{\alpha} \quad \Rightarrow \quad \lambda \boldsymbol{\alpha} = \widetilde{\mathbf{K}}_D \alpha. \tag{12}$$

Moreover,

$$\begin{aligned} \lambda \boldsymbol{\alpha} &= \mathbf{\Phi}_D^\top \mathbf{P}_\Phi \mathbf{P}_\Phi \mathbf{\Phi}_D \boldsymbol{\alpha} = \mathbf{\Phi}_D^\top \mathbf{P}_\Phi \mathbf{\Phi}_D \boldsymbol{\alpha} \\ &= \mathbf{\Phi}_D^\top (\mathbf{I}_\Phi - \mathbf{\Phi}_S \mathbf{K}_S^+ \mathbf{\Phi}_S^\top) \mathbf{\Phi}_D \boldsymbol{\alpha} \\ &= (\mathbf{K}_D - \mathbf{K}_{DS} \mathbf{K}_S^+ \mathbf{K}_{SD}) \boldsymbol{\alpha} \end{aligned} \tag{13}$$

with the kernel matrices $\mathbf{K}_D$, $\mathbf{K}_{DS} = \mathbf{K}_{SD}^\top$ and $\mathbf{K}_S$ defined as above. There are at most m eigenvectors corresponding to nonzero eigenvalues, and by orthonormality the corresponding vectors $\boldsymbol{\alpha}_j$ must be normalized such that $\langle \mathbf{v}_j, \mathbf{v}_j \rangle = \boldsymbol{\alpha}_j^\top \widetilde{\mathbf{K}}_D \boldsymbol{\alpha}_j = 1$. As in the linear case, we choose the eigenvectors corresponding to the l largest eigenvalues using an energy criterion. Given these bases, a test sample $\phi(\mathbf{x}_{\text{test}})$ can be projected onto each eigenvector using

$$\Omega_j = \boldsymbol{\alpha}_j^\top \widetilde{\mathbf{\Phi}}_D^\top \phi(\mathbf{x}_{\text{test}}) = \boldsymbol{\alpha}_j^\top \mathbf{\Phi}_D^\top \mathbf{P}_\Phi \phi(\mathbf{x}_{\text{test}}) \tag{14}$$

$$\begin{aligned} &= \boldsymbol{\alpha}_j^\top \mathbf{\Phi}_D^\top (\phi(\mathbf{x}_{\text{test}}) - \mathbf{\Phi}_S \mathbf{K}_S^+ \mathbf{\Phi}_S^\top \phi(\mathbf{x}_{\text{test}})) \\ &= \boldsymbol{\alpha}_j^\top (\mathbf{k}_{D,\text{test}} - \mathbf{K}_{DS} \mathbf{K}_S^+ \mathbf{k}_{S,\text{test}}), \end{aligned} \tag{15}$$

where $\mathbf{k}_{D,\text{test}} = [k(\mathbf{x}_{\text{test}}, \mathbf{x}_{i,1}) - k(\mathbf{x}_{\text{test}}, \mathbf{x}_{i,2})]_{i=1,\ldots,m}$ is m-dimensional kernel vector against the dissimilar sample pairs, and $\mathbf{k}_{S,\text{test}} = [k(\mathbf{x}_{\text{test}}, \mathbf{x}_{i,1}) - k(\mathbf{x}_{\text{test}}, \mathbf{x}_{i,2})]_{i=1,\ldots,n}$ is the corresponding n-dimensional kernel vector against the similar sample pairs. The final embedded vector is $\mathbf{\Omega}_{\text{test}} = [\Omega_1, \ldots, \Omega_l]^\top$.

4. Experiments

We performed experiments on two synthetic datasets and on several challenging image retrieval, object classification, gender classification and image segmentation problems[1]. In each case, distance metrics obtained with the proposed Semi-Supervised Discriminative Common Vectors (SS-DCV) algorithm are compared to the Euclidean Distance (ED) metric and to distance metrics learned using Information-Theoretic Metric Learning (ITML), RCA, Kernel RCA, DCA and Kernel DCA. The kernelized methods use Gaussian kernels. To show how effective similarity and dissimilarity constraints are at discovering hidden groups within the data, we apply k-means clustering in the embedding spaces returned

[1] For software see *http://www2.ogu.edu.tr/∼mlcv/softwares.html*.

Gaussian Data	Clustering Rate	Classification Rate
ED	45.26±5.52	76.80±2.51
ITML	**92.77**±3.99	**96.06**±2.34
RCA	47.11±4.40	84.67±2.01
DCA	55.76±6.54	83.80±5.35
SS-DCV	88.01±3.73	91.33±4.22
Checkerboard Data		
ED	54.36±1.53	48.26±4.50
Kernel RCA	57.76±3.54	49.53±5.70
Kernel DCA	64.54±12.17	87.45±2.39
Kernel SS-DCV	**86.10**±11.83	**90.52**±1.40

Table 1: Clustering and classification rates (%) on the synthetic datasets.

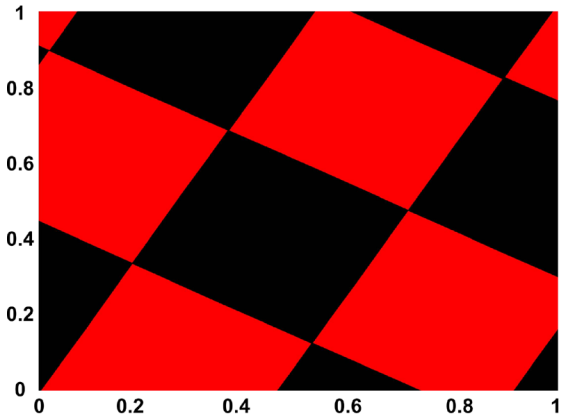

Figure 3: Rotated checkerboard data. The two classes are represented by different colors.

by the metric learning algorithms and quantify the effectiveness of the embedding by the percentage of same-class samples that are clustered together. We also report the classification accuracies of the tested embeddings using the nearest-class-mean classifier. For RCA, we include a regularizer $\delta\mathbf{I}$ to prevent singular covariance matrices, where δ is a small positive constant.

4.1. Synthetic Datasets

The first synthetic dataset contains 1000-dimensional feature vectors belonging to three classes. The first dimension is the only distinctive one, with the classes being normally distributed as respectively $N(-8, 1)$, $N(0, 1)$ and $N(8, 1)$. The remaining dimensions are irrelevant features distributed as $N(0, 4)$. We only tested linear methods as the data is linearly separable. 50 samples per class are used to define the equivalence constraints, with 100 samples per class for testing. In all, we included only 90 similarity and 60 dissimilarity constraints. The averages and standard deviations of the resulting clustering and classification accuracies over 10 independent runs are given in Table 1. Clearly, all of the distance learning algorithms tested give better results than the Euclidean distance. The best clustering and classification accuracies were obtained with ITML, followed by the proposed method, and both methods significantly outperformed all of the others tested. Moreover, note that the ITML results are anomalous in the sense that the dimensionality was too high to apply ITML directly, so we used PCA to reduce the dimensionality to 20 before applying it, whereas this was not necessary for the other methods tested.

For the second synthetic test we use a 2-D dataset in which the two classes cover respectively the red and the black squares of a rotated checkerboard as illustrated in Fig. 3. The points of each class fall into several distinct subclusters that are far from one another and intermixed with the subclusters of the other class, so it is hard to group them together. Linear separation is not possible so we used kernelized methods. We randomly sampled 300 points from the checkerboard for training, creating 150 similarity and 100 dissimilarity constraints using these, with 1000 additional points for testing. The clustering and classification accuracies over 10 random trials are shown in Table 1. The proposed method leads the table, failing to find the correct clusters only once in the 10 trials, whereas all of the other methods tested fail significantly more often, with ED and Kernel RCA giving results that are close to random.

4.2. Image Retrieval

For this experiment we use the dataset for image retrieval using side information defined by [36]. This contains 100 randomly selected images from the COREL CDs for each of 20 semantic classes (antelope, butterfly, cat, dog, mountain, roses, *etc*.). The images are represented by global 36-D feature vectors: 9 moments of HSV color histograms, 18 features from edge direction histograms, and 9 wavelet based texture features. We randomly selected $n = 50,\ 70,\ 90,\ 110$ similarity constraints and $m = 45,\ 60,\ 75,\ 90$ dissimilarity constraints from each

Constraints	n=50, m=45		n=70, m=60		n=90, m=75		n=110, m=90	
TOP	**10**	**20**	**10**	**20**	**10**	**20**	**10**	**20**
ED	45.7±0.4	38.9±0.4	45.8±0.4	38.9±0.4	45.8±0.4	38.8±0.4	45.5±0.4	38.6±0.4
ITML	28.8±2.8	24.3±2.7	27.8±1.9	23.6±2.1	27.7±3.2	23.3±3.0	18.8±5.0	16.0±4.0
RCA	48.4±0.7	40.9±0.6	48.3±0.6	40.8±0.5	48.2±0.5	40.7±0.4	48.6±0.7	41.0±0.7
Ker. RCA	35.4±0.7	28.4±0.6	33.0±0.9	26.4±0.9	31.8±0.8	25.6±0.5	33.0±0.7	27.1±0.4
Ker. DCA	37.7±0.7	32.2±0.7	38.0±0.6	32.3±0.5	38.0±0.5	32.3±0.6	38.0±0.4	32.4±0.5
Ker. SS-DCV	**52.1**±1.2	**43.4**±1.2	**58.2**±1.7	**50.1**±1.3	**61.5**±1.8	**55.1**±1.1	**65.6**±1.7	**59.0**±1.6

Table 2: Average precision (%) of the top ranked 10 or 20 images for image retrieval of twenty categories of 100 images chosen from the Corel dataset.

category. After learning the distance metric, each image in the whole dataset is tested as a query, in each case evaluating it using the learned embedding against the remaining images that were not used for creating constraints. The retrieval is evaluated based on the top-ranked 10 and 20 images. The experiment is repeated 10 times with different random samplings of the constraints. Table 2 reports the resulting average precisions (AP%) for the different numbers of constraints. The proposed method is significantly better than all of the others tested in all of the cases tested. For example, with $n = 110$, $m = 90$ constraints per category, it is about 17% better than the second best method RCA and 20% better than the Euclidean distance. ITML produces the worst results.

4.3. Visual Object Classification

For this experiment we sampled 600 images (150 per class) from the Caltech-4 dataset, which includes the four object categories airplanes, cars, faces and motorcycles as shown in Fig. 4. The images are too diverse to allow simple geometric alignment of their objects so we used a "bag of features" representation, computing SIFT descriptors over patches extracted at DoG interest points and generating a 500 word visual codebook by clustering them using k-means. Each image is thus represented by its 500-D visual word histogram. For each class we used 70 of the images to create the equivalence constraints, with the remaining 80 held back for testing. We randomly selected $n = 30,\ 50,\ 70,\ 90,\ 110$ similarity and $m = 15,\ 30,\ 45,\ 60,\ 75$ dissimilarity constraints for each class. As the χ^2 distance $CSD(\mathbf{u}, \mathbf{v}) = (1/2)\sum_{i=1}^{d}[(u_i - v_i)^2/(u_i + v_i)]$ is often a better metric for histogram comparison than the standard Euclidean Distance (ED), we report results

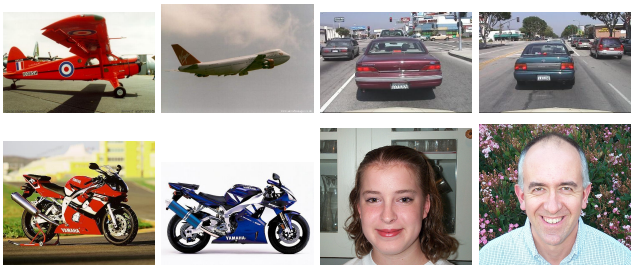

Figure 4: Some images from the Caltech-4 dataset.

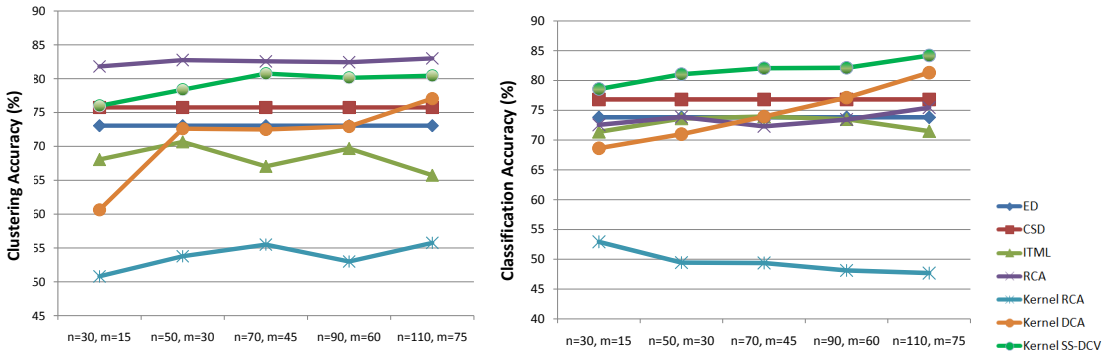

Figure 5: Clustering and classification accuracies on Caltech-4 (best viewed in color).

for both. For the kernelized CSD methods, we used the generalized-Gaussian CSD kernel $k(\mathbf{u}, \mathbf{v}) = \exp(-CSD(\mathbf{u}, \mathbf{v})/(2\sigma^2))$. Fig. 5 plots the resulting clustering and classification accuracies. As expected, CSD gives uniformly better results than ED. Overall, RCA is the best performer for clustering but a poor one for classification. The proposed method yields the second best results for clustering and the best ones for classification. Kernel DCA initially produces poor results as the number of dissimilarity constraints is low, but it becomes significantly better as this number is increased. Among the distance learning methods, only the proposed methods and Kernel DCA improve significantly as the number of constraints is increased. Kernel RCA produces very poor results as expected: RCA requires the inversion of a feature space scatter matrix and in high-dimensional spaces such as those given by Gaussian kernels, these matrices become severely rank deficient causing a large loss of accuracy. In contrast, the proposed method can make good use of the freedom provided by such high-dimensional feature spaces because it is based on the null space of the scatter matrix, not its range space as in RCA.

4.4. Experiments on Gender Database

Here we demonstrate how the proposed method can be used to organize image galleries in accordance to the personal preferences. For such applications, we determine a characteristic for distinction and group images based on this selection. In our case we group images by gender and use the gender recognition database used in [37]. This database consist of 1892 images (946 males and 946 females) coming from the following databases: AR, BANCA, Caltech Frontal face, Essex Collection of Facial Images, FERET, FRGC version 2, Georgia Tech and XM2VTS. Only the first frontal image of each individual was taken, however because all of the databases have more male subjects than females, the same number of images is taken for both male and female subjects. All images are cropped based on the eye coordinates and resized to 32×40 yielding a 1280-dimensional input space. Then, images are converted to gray-scale followed by histogram equalization. Some samples are shown in Fig. 6.

Figure 6: Some male and female samples from Gender database.

We used 50% of the images as training data and the remaining for testing. The dimensionality $d = 1280$ of the input space is too high, thus we reduced the dimensionality through PCA to 20 before application of ITML as before. We randomly selected $n = 100, 2000, 300, 400, 500$ similarity and $m = 50, 100, 150, 200,$-250 dissimilarity constraints. the experiment is repeated 5 times with different random samplings of the constraints. We used the Gaussian kernel function for all tested kernel methods. The resulting clustering and classification accuracies are given at Table 3 and Table 4, respectively. Our proposed method Kernel SS-DCV achieves the best clustering and classification rates for all cases. It is followed by Kernel DCA. For both methods, the accuracies increase as the number of constraints is increased as expected. The results saturate when $n = 400, m = 200$ constraints are used. Similar to the visual object classification problem, Kernel RCA produces very poor results. It should be noted that ITML yields very sim-

Constraints	m=100, n=50	m=200, n=100	m=3000, n=150	m=400, n=200	m=500, n=250
ED	53.81±2.4	55.13±1.1	55.38±2.5	55.31±1.5	54.94±1.8
ITML	66.34±3.0	69.60±1.5	70.86±2.6	68.10±2.6	69.28±2.2
RCA	68.11±2.0	67.60±4.3	66.38±2.9	61.08±4.2	60.47±2.4
DCA	65.10±2.9	71.23±1.1	74.79±1.3	76.76±2.4	76.58±1.6
SS-DCV	64.84±2.4	65.40±3.2	63.80±2.8	61.29±1.9	58.97±2.8
Kernel RCA	64.10±2.7	65.40±2.5	63.24±2.9	60.07±2.7	60.03±2.9
Kernel DCA	65.86±1.2	71.53±1.5	76.12±2.8	78.09±1.5	80.44±1.8
Kernel SS-DCV	**71.92**±3.0	**76.96**±1.1	**81.20**±0.9	**83.87**±1.2	**83.55**±0.8

Table 3: Clustering Accuracies for Different Number of Constraints on Gender Database.

Constraints	m=100, n=50	m=200, n=100	m=3000, n=150	m=400, n=200	m=500, n=250
ED	77.52±0.9	77.37±0.5	78.07±0.9	78.11±1.1	77.44±1.1
ITML	78.55±1.1	79.09±0.8	78.98±2.5	78.54±1.0	78.33±2.7
RCA	65.01±3.1	57.25±1.6	58.81±3.1	55.43±2.5	55.80±0.9
DCA	77.97±2.1	78.66±1.5	78.58±1.4	78.22±1.6	76.93±2.0
SS-DCV	71.63±2.3	66.15±1.0	64.51±2.2	62.58±2.2	61.56±2.1
Kernel RCA	62.31±4.5	54.42±2.1	52.55±2.3	51.63±1.6	51.86±1.8
Kernel DCA	77.70±1.8	78.31±1.2	80.02±2.1	80.11±1.7	80.85±0.9
Kernel SS-DCV	**78.40**±1.2	**81.76**±1.1	**82.39**±1.3	**84.13**±1.1	**84.14**±1.2

Table 4: Classification Accuracies for Different Number of Constraints on Gender Database.

ilar results for all cases even though the number of constraints is gradually increased. This shows that important discriminatory information is thrown away during dimensionality reduction step. The accuracy for the proposed linear SS-DCV method decreases as the number of constraints is increased. This is not surprising since as the number of similarity constraints are increased, the null space becomes smaller and smaller and it is hard to find discriminatory directions in such a small subspace. This does not apply to Kernel SS-DCV since the input space is mapped to an infinite-dimensional space.

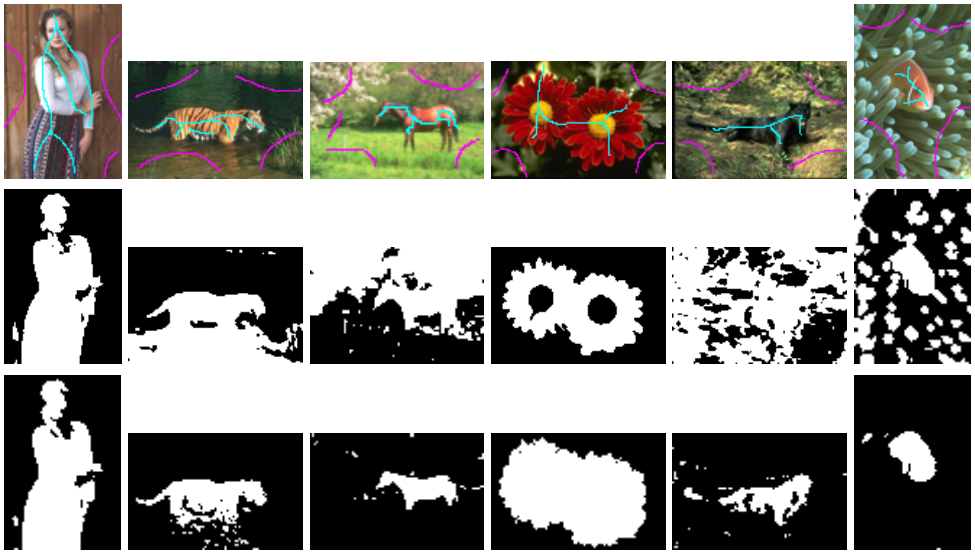

Figure 7: Image segmentation results (best viewed in color). First row: the pixels used for the equivalence constraints. Second row: the segmentation results without the constraints. Bottom row: the segmentation results incorporating the constraints. The results have not undergone morphological cleaning.

4.5. Image Segmentation

Finally, we tested the proposed method on an image segmentation problem. We selected images from the Berkeley Segmentation dataset[2]. For each pixel in each image, we extracted the 20×20 image patch centered at the pixel and computed its robust hue descriptor [38]. The resulting 36-D feature vectors are histograms over the hue values observed in the patch, with each value being weighted by its saturation. As a baseline we used heat kernel based Normalized Cut (NCut) image segmentation [39] with two clusters, one for the object of interest and one for the background. The second row of Fig. 7 shows these unsupervised segmentation results. Similarity and dissimilarity constraints were defined by sampling the pixels shown respectively in magenta and cyan in the first row of Fig. 7, and the proposed method was used to learn an embedding function from these. K-means clustering in the embedded space was then used to segment the image, giving the results shown in the third row of the figure. As can be seen, the addition of simple user defined (dis)similarity constraints significantly improves the segmentations.

[2]From http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench

For example, in the flower image there are essentially three well-separated color components: the green background, the red petals and the yellow flower centres. There are thus three reasonable segmentations, each separating one of the three components from the other two. It is not clear a priori which of these is desired by the user, but a few similarity constraints suffice to specify this.

5. Summary and Conclusions

This paper describes a method for learning distance metrics in high-dimensional feature spaces using pairwise similarity (equivalence) and dissimilarity (non - equivalence) constraints. The method finds a linear mapping that projects the input to a lower dimensional space in which similar point pairs coincide and the scatter of the dissimilar point pairs is as large as possible. It does this by first projecting the data onto the orthogonal complement of the directions spanned by the similar pairs, then optimizing a conventional scatter criterion in the projected space. The method works in both explicit and kernel-induced feature spaces and it can learn rectangular projection matrices that yield low-rank distance metrics. Experimental results on synthetic datasets and on several real computer vision problems show that the proposed method achieves significantly better clustering and classification accuracies than existing distance metric learning methods.

References

[1] B. Babenko, S. Branson, S. Belongie, Similarity metrics for categorization: from monolithic to category specific, in: ICCV, 2009.

[2] H. Cevikalp, R. Paredes, Semi-supervised distance metric learning for visual object classification, in: Inter. Conf. on Computer Vision Theory and Applications, 2009.

[3] A. Ghodsi, D. Wilkinson, F. Southey, Improving embeddings by flexible exploitation of side information, in: Inter. Joint Conf. on Artificial Intelligence, 2007.

[4] T. Hertz, N. Shental, A. Bar-Hillel, D. Weinshall, Enhancing image and video retrieval: Learning via equivalence constraints, in: CVPR, 2003.

[5] S. C. H. Hoi, W. Liu, M. R. Lyu, W. Y. Ma, Learning distance metrics with contextual constraints for image retrieval, in: CVPR, 2006.

[6] J. Yu, D. Tao, Y. Rui, J. Cheng, Pairwise constraints based multiview features fusion for scene classification, IEEE Transactions on on Image Processing 46 (2013) 483–496.

[7] A. Globerson, S. Roweis, Metric learning by collapsing classes, in: NIPS, 2005.

[8] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinow, Neighborhood component analysis, in: NIPS, 2004.

[9] J. Yu, M. Wang, D. Tao, Semisupervised multiview distance metrics learning for cartoon synthesis, IEEE Transactions on on Image Processing 21 (2012) 4636–4668.

[10] M. Wang, X.-S. Hua, J. Tang, R. Hong, Beyond distance measurement: Constructing neighborhood similarity for video annotation, IEEE Transactions on Multimedia 11 (2009) 465–476.

[11] L. Yang, R. Jin, Distance metric learning: A comprehensive survey, in: http://www.cse.msu.edu/∼yangliu1/ framesurveyv2.pdf, 2006.

[12] E. P. Xing, A. Y. Ng, M. Jordan, S. Russell, Distance metric learning with application to clustering with side-information, in: NIPS, 2003.

[13] I. W. Tsang, J. Kwok, Distance metric learning with kernels, in: Inter. Conf. on Artificial Neural Networks, 2003.

[14] S. Shalew-Shwartz, Y. Singer, A. Y. Ng, Online and batch learning of pseudo-metrics, in: ICML, 2004.

[15] J. V. Davis, B. Kullis, P. Jain, I. S. Dhillon, Information-theoretic metric learning, in: ICML, 2007.

[16] M. Guillaumin, J. Verbeek, C. Schmid, Is that you? metric learning approaches for face identification, in: ICCV, 2009.

[17] J. V. Davis, B. Kullis, P. Jain, I. S. Dhillon, Structured metric learning for high-dimensional problems, in: ACM SIGKDD Inter. Conf. on Knowledge Discovery and Data Mining, 2008.

[18] Y. Ying, K. Huang, C. Campbell, Sparse metric learning via smooth optimization, in: NIPS, 2009.

[19] W. Liu, S. Ma, D. Tao, J. Liu, P. Liu, Semi-supervised sparse metric learning using alternating linearization optimization, in: ACM SIGKDD Inter. Conf. on Knowledge Discovery and Data Mining, 2010.

[20] S. An, W. Liu, S. Venkatesh, Exploiting side information in locality preserving projection, in: CVPR, 2008.

[21] H. Cevikalp, J. Verbeek, F. Jurie, A. Klaser, Semi-supervised dimensionality reduction using pairwise equivalence constraints, in: Int. Conf. on Computer Vision Theory and Applications, 2008.

[22] H. Cevikalp, J. Verbeek, F. Jurie, A. Klaser, Semi-supervised dimensionality reduction using pairwise equivalence constraints, in: Inter. Conf. on Computer Vision Theory and Applications, 2008.

[23] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, X. Wu, Visual-textual joint relevance learning for tag-based social image search, IEEE Transactions on Image Processing 22 (2013) 363–376.

[24] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, 3-d object retrieval and recognition with hypergraph analysis, IEEE Transactions on Image Processing 21 (2012) 4290–4303.

[25] M. Wang, X.-S. Hua, R. Hong, J. Tang, G.-J. Qi, Y. Song, Unified video annotation via multigraph learning, IEEE Transactions on Circuits and Systems for Video Technology 19 (2009) 733–746.

[26] N. Shental, T. Hertz, D. Weinshall, M. Pavel, Adjustment learning and relevant component analysis, in: ECCV, 2002.

[27] I. W. Tsang, P.-M. Cheung, J. T. Kwok, Kernel relevant component analysis for distance metric learning, in: Inter. Joint Conf. on Neural Networks, 2005.

[28] M. Bilenko, S. Basu, R. J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, in: ICML, 2004.

[29] O. Tuzel, F. Porikli, P. Meer, Kernel methods for weakly supervised mean shift clustering, in: ICCV, 2009.

[30] H. Yu, J. Yang, A direct lda algorithm for high-dimensional data with application to face recognition, Pattern Recognition 34 (2001) 2067–2070.

[31] H. Cevikalp, M. Neamtu, M. Wilkes, Discriminative common vectors with kernels, IEEE Transactions on Neural Networks 17 (2006) 1550–1565.

[32] J. Ye, T. Xiong, Computational and theoretical analysis of null space and orthogonal linear discriminant analysis, Journal of Machine Learning Research 7 (2006) 1183–1204.

[33] H. Cevikalp, M. Wilkes, Face recognition by using disriminative common vectors, in: International Conference on Pattern Recognition, 2004.

[34] P. Simard, Y. L. Cun, J. Denker, B. Victorri, Transformation invariance in pattern recognitiontangent distance and tangent propagation, Lecture Notes in Computer Science 1524 (1998) 239–274.

[35] D. Decoste, B. Schölkopf, Training invariant support vector machines, Machine Learning 46 (2002) 161–190.

[36] S. C. Hoi, R. Jin, J. Zhu, M. R. Lyu, Semi-supervised svm batch mode active learning for image retrieval, in: CVPR, 2008.

[37] M. Villegas, R. Paredes, Simultaneous learning of a discriminative projection and prototype for nearest-neighbor classification, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2008.

[38] J. van de Weijer, C. Schmid, Coloring local feature extraction, in: ECCV, 2006.

[39] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on PAMI 22 (2000) 885–905.