

Classification of Weakly-Labeled Data with Partial Equivalence Relations

Sanjiv Kumar
Google Research
New York, NY 10011
sanjivk@google.com

Henry A. Rowley
Google Research
Mountain View, CA 94043
har@google.com

Abstract

In many vision problems, instead of having fully labeled training data, it is easier to obtain the input in small groups, where the data in each group is constrained to be from the same class but the actual class label is not known. Such constraints give rise to partial equivalence relations. The absence of class labels prevents the use of standard discriminative methods in this scenario. On the other hand, the state-of-the-art techniques that use partial equivalence relations, e.g., Relevant Component Analysis, learn projections that are optimal for data representation, but not discrimination. We show that this leads to poor performance in several real-world applications, especially those with high-dimensional data. In this paper, we present a novel discriminative technique for the classification of weakly-labeled data which exploits the null-space of data scatter matrices to achieve good classification accuracy. We demonstrate the superior performance of both linear and nonlinear versions of our approach on face recognition, clustering, and image retrieval tasks. Results are reported on standard datasets as well as real-world images and videos from the web.

1. Introduction

Fully supervised classification requires the class labels to be known while training, which usually involves tedious manual labeling. For many applications, it is much easier to obtain training data which is only ‘weakly’ labeled. For example, for face recognition in video, one can use a coarse face tracker to detect small video clips such that each clip contains the images of the same individual (Figure 1, top), but the identity of the individual is not known. Similarly, for image retrieval, one can construct a training set by grouping images into small groups such that images within each group are ‘similar’, but the actual class of the group is not known (Figure 1, bottom). Preparing such a set is much simpler than labeling each image with a distinct class label.

In this paper, we address the problem of learning classifiers using weakly-labeled training data, in which the train-



Figure 1. An illustration of the form of weakly-labeled training data with partial equivalence relations for face recognition (top row) and image retrieval (bottom row). The input data consists of groups (shown in ellipses), each containing data from the same class but the actual class label is not known. Note that different groups may share the same class label as shown in columns (A) and (D), which contain images of the same person (top row) and of the ‘flower’ class (bottom row).

ing input is given in the form of small groups. The data in each group comes from the same class but the actual class labels are not known. Such a scenario yields pairwise relations, *i.e.*, ‘X is similar to Y’ for all the points X and Y that belong to the same group. However, it is important to note that it does *not* give relations of type ‘X is dissimilar to Z’ if X and Z come from different groups. This is because two different groups may have the same (unknown) class label, as shown in Figure 1 for groups (A) and (D). We call the relations in this scenario *partial equivalence relations*. We use the term ‘partial’ to emphasize that we do not have access to any pairs that contain points from different classes.

In the presence of partial equivalence relations, standard discriminative techniques cannot be used due to the absence of class labels in the training data. Moreover, the lack of access to both ‘similar’ and ‘dissimilar’ pairs in the training data precludes the use of many promising techniques that

work with equivalence relations [14][17][6]. These techniques discriminatively learn a distance metric for clustering or classification when both *similar* and *dissimilar* pairs are given. In the absence of *dissimilar* pairs, Xing *et al.* [17] suggest treating the data pairs that are not *similar* as the *dissimilar* pairs. But, such heuristics have been shown to give lower accuracy in comparison to the techniques that use only *similar* pairs [1].

In vision, Shental *et al.* [11] introduced learning with partial equivalence relations under a paradigm named Adjustment Learning. They proposed a promising approach called Relevant Component Analysis (RCA), which finds a linear transformation of the data such that irrelevant variability in the data is reduced. A nonlinear extension of RCA called kernel RCA was proposed by Tsang *et al.* [12]. Even though RCA has been shown to perform well in several applications, it suffers from two main problems. First, similar to Principal Component Analysis (PCA), RCA finds projections that are good for data representation or compression. These projections may not be good for class discrimination (see Section 3 for more details). Second, in the case of high-dimensional data with insufficient samples, linear RCA becomes ill-posed. In fact, the issue of scatter matrices becoming singular is not restricted to small-sample-size problems. Kernel RCA always suffers from the singularity problem no matter how much training data is available because the size of the kernel matrix grows in proportion to the number of training samples [12].

In vision applications, data often lies in a high-dimensional space. For instance, in face recognition, it is common to represent faces as vectors of pixel intensities or Gabor jets, yielding $O(10^3)$ to $O(10^5)$ dimensions. The same is true in image retrieval, where images are typically represented using color, texture and shape features in high dimensions. The method suggested in [11] to overcome the singularity problem is based on intermediate PCA-based dimensionality reduction step. This yields inferior results as we show later in Section 5 and Section 6. In [12], the authors suggest regularization of scatter matrices which requires picking the ‘right’ value of the regularization coefficient for good performance. This is usually done by hand-tuning the coefficient. Recently, Gaussian Mixture Model (GMM) based formulations have been developed for partial equivalence relations [10] but they become impractical for high-dimensional data due to the singularity problem.

In this work we take a different approach and argue that instead of getting rid of the singularities, these can be exploited to overcome both the shortcomings of RCA. In fact, we show that the null-space based techniques proposed in vision literature for fully supervised classification problems [3][7] can be naturally extended to address the problem of classification with weakly-labeled data. The key advantage of our technique is that it results in projections that are good

for class discrimination unlike RCA. In addition, our formulation does not involve hand-tuned free parameters such as the number of principal components in RCA, or the regularization coefficient in kernel RCA.

In this paper, we focus on discriminatively learning linear or nonlinear transformations using weakly-labeled training data such that a Nearest Neighbor (NN) classifier in the transformed space gives good classification accuracy. One important family of algorithms that learns a transformation or embedding of unlabeled data includes Locally Linear Embedding (LLE) [9] and Semi-Definite Embedding (SDE) [16]. However, it is not clear how to modify these unsupervised methods to make use of the partial equivalence relationships given in the input. RCA (along with its kernel extensions [12]) is the most popular technique to learn projections using partial equivalence relations, but it does not learn discriminative projections as discussed above.

To summarize, our paper (i) *proposes an algorithm to discriminatively learn a linear or nonlinear transformation using partial equivalence relations*, (ii) *extends null-space based analysis to weakly-labeled data*, and (iii) *demonstrates superior performance on standard as well as real-world web data for recognition, clustering and retrieval*.

After defining the problem formally in the next section, we briefly discuss the RCA technique in Section 3. Our null-space based technique is described in Section 4. Sections 5 and 6 show the experiments on standard and real-world web datasets, respectively.

2. Problem formulation

Suppose input data is given by $X = \{x_i\}_{i=1}^N$ where $x_i \in \mathbb{R}^d$, and each point takes one of the C class labels. Let H_c be the set of all the data with class label c such that, $\bigcup_c H_c = X$. We are not given the class labels during training. Instead, we are provided with groups such that data in a group share the same class label. Let there be R such groups, $\{X_r\}_{r=1}^R$, such that $\bigcup_r X_r = X$, and $R \geq C$. Let N_r be the number of points in group r . Our goal is to find a transformation of X such that a NN classifier gives good classification accuracy in the transformed space.

Since our technique builds upon traditional Fisher’s Linear Discriminant Analysis (LDA), we start with a brief review of LDA to set the notation. When all the class labels are available during training, LDA proposes to find the projection matrix \hat{W} that minimizes the within-class scatter, S_w , and maximizes the between-class scatter, S_b , by maximizing the following ratio [4],

$$\hat{W} = \arg \max_W \frac{|W^T S_b W|}{|W^T S_w W|}, \quad (1)$$

where $|\cdot|$ denotes the matrix determinant and,

$$S_w = \sum_{c=1}^C \sum_{x \in H_c} (x - \mu_c)(x - \mu_c)^T, \quad (2)$$

$$S_b = \sum_{c=1}^C N_c (\mu_c - \mu)(\mu_c - \mu)^T. \quad (3)$$

Here μ is the mean of the total input X , μ_c is the mean and N_c is the number of points in class c , and C is the number of classes. If S_w^{-1} exists, the columns of \hat{W} are simply the eigenvectors of $S_w^{-1}S_b$. If data from different classes is distributed normally with equal covariances, it is easy to show that LDA returns the optimal directions that maximize the class separation [4]. Even when data does not follow this generative model, the LDA criterion returns useful discriminative projections. In our case, since we do not have access to the class labels, LDA cannot be applied directly to learn the desired projections.

3. Relevant Component Analysis (RCA)

RCA assumes the data from each class to be distributed as Gaussian, *i.e.* $X_c \sim \mathcal{N}(\mu_c, \Sigma_c) \forall c = 1, \dots, C$. Further, the class covariances are assumed to be equal, *i.e.* $\Sigma_c = \Sigma \forall c$, and Σ is estimated as the weighted average of empirical class covariances resulting in $\Sigma \approx \frac{1}{N}S_w$. Since the class labels are not known, RCA approximates S_w with the within-group scatter matrix, S_g , defined using the group data $\{X_r\}_{r=1}^R$ as,

$$S_g = \sum_{r=1}^R \sum_{x \in X_r} (x - \mu_r)(x - \mu_r)^T, \quad (4)$$

where μ_r is the mean of the set X_r . In RCA, the optimal projection matrix is given by the whitening transformation, *i.e.*, $\hat{W}^T = \Sigma^{-1/2} = \Lambda^{-1/2}V^T$, where matrices V and Λ contain the eigenvectors and the eigenvalues of S_g . Classification is performed by using a NN classifier in the transformed space with Euclidean distance.

The above process is very similar to the idea of data whitening usually performed with PCA except that in PCA one uses total scatter of the data instead of within-group scatter used in RCA. The projections of RCA are hence optimized for representation or compression of data in a group. Another interpretation of RCA is that it maximizes the constrained mutual information between the input vector and its projection as explained in [1]. However, this argument also points to the fact that RCA is optimal for data compression, not for class discrimination.

3.1. RCA in high dimensions

In the previous discussion on RCA, the matrix S_g was assumed to be full-rank. When this is not true, RCA proposes to project the data on the top m eigenvectors of the total scatter matrix, where m is the effective rank of S_g . This is equivalent to using PCA for dimensionality reduction before computing the whitening transformation, \hat{W} . However, a potential problem in doing so is that the intermediate PCA step may discard dimensions that contain important

discriminative information. This can also be seen from the form of the whitening transform estimated by RCA given in the previous section. Ideally, the dimensions for which the eigenvalues of S_g are zero (*i.e.*, the null space of S_g) should be given infinite weight while computing the transformed data. However, PCA gets rid of those dimensions. Our null-space method, instead, exploits the null space of S_g to compute discriminative projections.

4. Null-Space Projections with Partial Equivalence Relations (NP-PER)

Our work on null-space projections is inspired by the work of Huang et al. [7] in the context of finding the best LDA projections given fully labeled data. However, since we do not have access to the class labels, we make the same assumption as made in RCA and approximate within-class scatter, S_w , with S_g defined in (4). Similar to LDA, we would like to find those transformations that minimize the within-group scatter (S_g) after projection since the data in each group belongs to the same class. However, since any two input groups may contain data from the same class, the between-group scatter, $S_{\bar{g}}$, defined as,

$$S_{\bar{g}} = \sum_{r=1}^R N_r (\mu_r - \mu)(\mu_r - \mu)^T, \quad (5)$$

is not a good approximation of the between class scatter, S_b . Hence, unlike LDA, it is incorrect to look for the directions that maximize $S_{\bar{g}}$ after projection. Nevertheless, it is obvious that after projection, $S_{\bar{g}}$ should not collapse to zero. In such a case, the projected means of *all* the groups would coincide, and no discrimination would be possible between different classes. Hence, in this work, we look for the projections W that optimize the following function:

$$\begin{aligned} \hat{W} &= \arg \min_W |W^T S_g W| & (6) \\ \text{Subject to} & \quad |W^T S_{\bar{g}} W| > 0, \text{ and} \\ \|w_j\|^2 &= 1 \text{ for } j = 1, \dots, m \end{aligned}$$

where w_j 's are the columns of the projection matrix W . The unit norm constraints on w_j 's are imposed as scale is arbitrary.

4.1. Nonlinear extension

For many real-world problems, a linear solution given by (6) may not be powerful enough. To allow more expressive nonlinear transformations, we use the kernel methods similar to [8][15]. Let Φ be a nonlinear mapping of the original feature vector x into a high-dimensional space \mathcal{F} . Let S_g^Φ and $S_{\bar{g}}^\Phi$ be the new scatter matrices in \mathcal{F} , obtained by simply replacing x with $\Phi(x)$ in (4) and (5), respectively. Then, the optimal linear discriminant, \hat{W}^Φ , in \mathcal{F} can be computed by replacing S_g and $S_{\bar{g}}$ in (6) by S_g^Φ and $S_{\bar{g}}^\Phi$. Note that the

linear discriminant in \mathcal{F} induces a nonlinear discriminant in the original input space.

Generally, the mapping Φ induces a very high (possibly infinite) dimensional space and directly solving (6) in \mathcal{F} is not feasible. Instead, we exploit popular kernel methods to solve (6) by using a Mercer kernel $k(\cdot, \cdot)$, for which, $k(x_1, x_2) = \Phi(x_1) \cdot \Phi(x_2)$. From the theory of Reproducing Kernel Hilbert Space (RKHS), the optimal solution of (6) in \mathcal{F} lies in the span of all the training samples, *i.e.*,

$$w_j^\Phi = \sum_{i=1}^N \alpha_{ij} \Phi(x_i) \quad \text{for } j = 1, \dots, m. \quad (7)$$

Let us define a new feature vector for each input, x_i , as $y_i = [k(x_1, x_i), k(x_2, x_i), \dots, k(x_N, x_i)]^T$. Also, let S_g^y and S_g^Φ be the new scatter matrices computed using features y_i in (4) and (5). Then, using the dot product property of Mercer kernels, one can show that $W^{\Phi T} S_g^\Phi W^\Phi = \alpha^T S_g^y \alpha$, and $W^{\Phi T} S_g^\Phi W^\Phi = \alpha^T S_g^y \alpha$ [8]. Hence, the optimization problem in (6) can be expressed in the kernel space \mathcal{F} as,

$$\hat{\alpha} = \arg \min_{\alpha} |\alpha^T S_g^y \alpha| \quad (8)$$

Subject to $|\alpha^T S_g^y \alpha| > 0$, $\|\alpha_j\|^2 = 1$ for $j = 1, \dots, m$.

4.2. Optimization

To compute $\hat{\alpha}$ in (8), the first thing to note is that, since S_g is positive semidefinite, the objective function satisfies $|\alpha^T S_g^y \alpha| \geq 0$. The constraints given in (8) eliminate the possibility of the trivial solution, $\hat{\alpha} = 0$. However, the objective function can still attain the absolute minimum of zero if $\hat{\alpha}$ lies in the null space of S_g^y , provided such a null-space exists. To check this, we first analyze the ranks of the two scatter matrices S_g^y and S_g^Φ . From the definition of S_g^y and S_g^Φ , it is easy to show that

$$\text{rank}(S_g^y) \leq N - R \quad \text{and} \quad \text{rank}(S_g^\Phi) \leq R - 1, \quad (9)$$

where N is the number of input samples, and R is the number of data groups. Since both of these matrices are of size $N \times N$, it is clear that they are rank-deficient. It is important to note that in the linear version of NP-PER, S_g is rank-deficient only if $d > N - R$. But in the kernel version, it will always be rank-deficient no matter how much training data is provided. This is because the size of S_g^y also increases as the number of training samples is increased.

Thus, the objective in (8) is minimized (*i.e.* takes value 0) if $\alpha_j \in \text{Null}(S_g^y)$, $\forall j$, where $\text{Null}(A)$ is the null-space of matrix A . But, to satisfy the constraints in (8), it is required that $\alpha_j \notin \text{Null}(S_g^y)$, $\forall j$. In other words, the eigenvectors that are in $\text{Null}(S_g^y) \cap \text{Null}(S_g^\Phi)$ should be discarded. Let us define the total scatter of the features, $\{y_i\}_{i=1}^N$, as:

$$\begin{aligned} S_t^y &= \sum_{i=1}^N (y_i - \mu^y)(y_i - \mu^y)^T, \\ &= S_g^y + S_g^\Phi, \end{aligned} \quad (10)$$

where $\mu^y = (1/N) \sum_i y_i$. Following the reasoning in [7], since all the matrices in (10) are positive semidefinite, $\text{Null}(S_t^y) = \text{Null}(S_g^y) \cap \text{Null}(S_g^\Phi)$. From (10), S_t^y has at most $N-1$ nonzero eigenvalues because $\text{rank}(S_t^y) \leq N-1$. Let P_t^y be the matrix containing the eigenvectors corresponding to nonzero eigenvalues of S_t^y . Thus, by projecting S_g^y onto P_t^y , one can get rid of the null space of S_g^y . Let Q_g^y be the matrix that contains the eigenvectors corresponding to zero eigenvalues of the projected within-group scatter matrix, $P_t^{y T} S_g^y P_t^y$. In other words, Q_g^y spans the null-space of $P_t^{y T} S_g^y P_t^y$. Then, the desired projections are given by the columns of Q_g^y in the projected space, *i.e.*, $\hat{\alpha} = P_t^y Q_g^y$.

Interpreted geometrically, NP-PER finds the projection directions $\hat{\alpha}$ such that the data in each group collapses to the group mean, while ensuring that the means of different groups do not overlap. The former property is useful for a NN classifier because the data in each group comes from a single class. The latter property, which results from the constraints in (8), makes sure that class discrimination is possible by keeping the group means separated.

4.3. Optimal number of projection vectors

What should be the optimal number of projections, m ? For the high-dimensional case in RCA, Shental et al. [11] recommend taking the effective rank of S_g , *i.e.*, the number of singular values that are ‘significantly’ larger than zero, as m . The accuracy of RCA is quite susceptible to the choice of m as we will show later in the experiments in Section 5.1. The kernel extension of RCA also suffers from a similar problem. On the contrary, m is fixed for NP-PER (and its nonlinear extension) for a given input.

To see this, first note that $\text{rank}(P_t^{y T} S_g^y P_t^y) = \text{rank}(S_g^y)$, because P_t spans the range of S_t^y , and the null space of S_t^y is the common null space of S_g^y and S_g^Φ [18]. The number of optimal projection vectors, m , is given by the dimension of $\text{Null}(P_t^{y T} S_g^y P_t^y)$. Hence,

$$m = \text{rank}(S_t^y) - \text{rank}(S_g^y). \quad (11)$$

When all the input samples are linearly independent, the inequalities in (9) become strict equalities. In this case, the optimal m is given as, $m = (N - 1) - (N - R) = R - 1$. It is interesting to note that m is independent of the original dimension, d , and the number of input samples, N .

4.4. Effect of group size

In the NP-PER analysis, we have approximated the within-class scatter, S_w , with within-group scatter, S_g , similar to [11]. The same holds for kernel NP-PER, except the inputs $\{x_i\}$ are replaced by their kernel mapped versions $\{y_i\}$ as described in Section 4.1. Comparing (2) and (4), the error in approximation is:

$$S_w - S_g = \sum_{i=1}^N (\mu_{c_i} - \mu_{r_i})(\mu_{c_i} - \mu_{r_i})^T, \quad (12)$$

where c_i is the actual class label of input x_i , and r_i is the group to which data x_i belongs. Clearly, the error depends on how accurately a group mean approximates its true (unknown) class mean. As the group size, N_r , increases, the error in approximation goes down.

When $N_r \rightarrow \infty$, $\forall r$, one can show using Chebyshev's inequality for the weak law of large numbers for two different sample means that $P(|\mu_{c_i} - \mu_{r_i}| > \epsilon) \rightarrow 0$ for any $\epsilon > 0$. Thus, in the infinite data case, NP-PER will be equivalent to fully supervised improved null-space LDA described in [18] even when explicit class labels are unknown.

4.5. Computational issues

The computation of the optimal projection matrix, $\hat{\alpha}$, in kernel NP-PER involves three steps. First, computing the kernel mapping y_i for each input x_i , which is $O(N^2d)$. Second, computing the vectors P_t^y spanning the range of the total scatter S_t^y . Since S_t^y is of size $N \times N$, P_t^y can be obtained by doing eigen-analysis or SVD, which is $O(N^3)$. In fact, if the inputs are known to be linearly independent, this step can be skipped. This is because, in that case, the rank of S_t^y is $(N-1)$ and P_t^y can be obtained by simply discarding the last column of S_t^y [15]. The final step involves computing the null space of $P_t^y S_g^y P_t^y$, whose complexity is upper bounded by $O((N-1)^3)$.

The above computations are affordable for moderate N . For large N , one may adopt iterative procedures for solving SVD as described in [5]. Note that the computational efforts needed for kernel NP-PER are equivalent to those for kernel RCA, since kernel RCA also needs to compute the kernel mapping, the top m eigenvectors of S_t^y , and the SVD of the projected S_g^y . Linear RCA and linear NP-PER also have equivalent computational complexities.

5. Experiments on standard datasets

5.1. Classification

We use three standard face datasets (Yale, FERET and ORL) to compare the face recognition performance of four linear techniques: Eigenfaces [13], Fisherfaces [2], RCA [11] and our null-space method, NP-PER, and two nonlinear techniques: kernel RCA (kRCA) and kernel NP-PER (kNP-PER). The comparisons with Eigenfaces and Fisherfaces are meant to demonstrate how the performance of linear projection methods vary when data is unlabeled (Eigenfaces) to when data is fully labeled (Fisherfaces). Since both RCA and NP-PER use weakly labeled data, their performance is expected to fall between these two extremes. The RCA and NP-PER methods are compared by varying the number of groups, and the number of images in each group. During testing, for each projected test image, its nearest neighbor (in the Euclidean sense) from the projected training set is used for classification.

The Yale database [2] consists of 156 face images of 15 individuals. These images are randomly split into a training set containing 105 images (7 images per class) and a test set containing the remaining 51 images. From each image, a tightly-cropped face is extracted and warped to a standard size of 48×48 pixels, yielding a vector of dimension $d = 2304$. To deal with illumination variations, the output image is further normalized such that each pixel has zero mean and unit variance within a local 5×5 pixel window.

We compared RCA, NP-PER and their kernel versions for three different group sizes: $N_r = 3, 5, 7$. For each size, we assigned the maximum possible data from the training set to groups. The groups were populated by randomly picking data from each class. For each of the above training sets, we first computed the optimal linear transformation, \hat{W} , for RCA and NP-PER as described in Sections 3 and 4, respectively. Note that \hat{W} for linear NP-PER is obtained by simply setting the transformed features $y_i = x_i \forall i$. The optimal number of projections, m , for NP-PER is fixed and computed using (11). For RCA, we chose the m that gave minimum error on the training set. The \hat{W} for Eigenfaces is learned only once for the whole training set, as it is independent of sampling variations in the groups. Also, Fisherfaces, being a fully supervised technique, is applied only when all the data from each class is contained within one group, *i.e.* when $N_r = 7$.

The classification performance of different linear methods on the Yale set is given in the left plot in Figure 2. It shows the mean error rate and one standard deviation error bars obtained by repeating the experiments with 100 random group realizations. Since Eigenfaces and Fisherfaces do not depend on group selection, their results do not show any error bars. As expected, Eigenfaces (PCA) gives the poorest error rate, as it does not make use of the partial equivalence relations given in the data. For each value of N_r , it is clear that the proposed NP-PER algorithm yields a lower error rate than RCA, verifying that the null space of the within-group scatter matrix contains useful discriminative information. As the size of each group increases from 3 to 5, the errors for both RCA and NP-PER are expected to decrease. However, as shown in the plot, the RCA error increases. This may be due to the fact that for $N_r = 5$, only about 71% of input data could be used for training while $N_r = 3$ allows the use of about 86% of the data.

When $N_r = 7$, since all the training data from a class is assigned to a single group, both RCA and NP-PER give better results than for any other N_r . In this case, NP-PER is equivalent to the improved null-space LDA [18], which performed similar to Fisherfaces and better than RCA. Fisherfaces performs better than RCA because Fisherfaces explicitly looks for those directions that also maximize between-class scatter, S_b , unlike RCA, which ignores it.

The right plot in Figure 2 shows the dependency of RCA

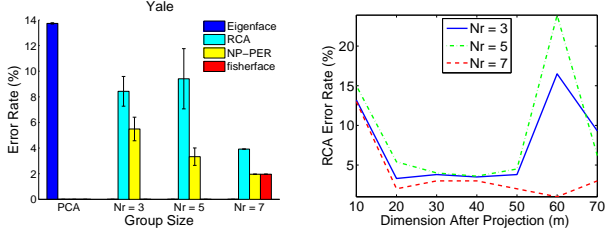


Figure 2. Results on Yale dataset. Left: Mean error rates on test set as a function of group size. Error bars show standard deviation for 100 random experiments repeated with different group samples. Note that for maximum N_r , there is no variability in the results as all the data from a class is assigned to a single group. Right: Plot showing strong dependence of mean training error rates for RCA on the dimension, m , of the projected vector.

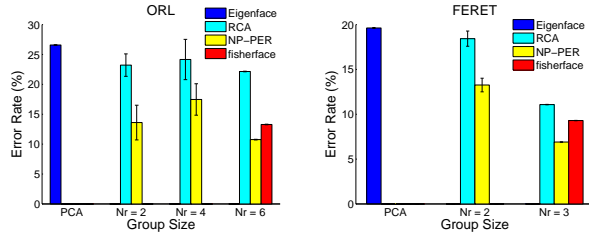


Figure 3. Graphs comparing the mean classification error rates for each algorithm on the ORL and the FERET test sets.

training error on the dimension, m , of the projected vector. For different choices of m , the performance of RCA varies significantly for all the group sizes. We picked the best number ($m = 20$) for the experiments with the Yale set, which gave almost the same accuracy as $m = 50$, but at a lower computational cost. Note that for NP-PER, one does not need to search over m , as it is always fixed for a given N_r , as shown in (11). For Eigenfaces and Fisherfaces, m was fixed to be the same as for RCA in all the experiments.

Next, we compared the performance of the nonlinear versions of RCA and NP-PER, kRCA and kNP-PER, on the Yale set (Figure 4, left graph). A Gaussian kernel, $k(x_1, x_2) = \exp(-\|x_1 - x_2\|^2/\sigma^2)$ was used, which performed better than the cosine kernel suggested in [15]. The bandwidth, σ , was selected using cross-validation. Comparing Figures 2 and 4, both kNP-PER and kRCA perform better than linear NP-PER and RCA, respectively. Also, kNP-PER outperforms kRCA significantly for all the group sizes.

We also conducted the above sets of experiments on two other standard datasets: ORL and FERET. The training set contained 240 images (40 individuals) for ORL, and 717 images (239 individuals) for FERET, while the test sets contained 158 and 473 images, respectively. For the ORL data (6 images per individual), experiments were run with $N_r = 2$, $N_r = 4$, and $N_r = 6$. For the FERET data (3 images per individual), $N_r = 2$ and $N_r = 3$ were used for

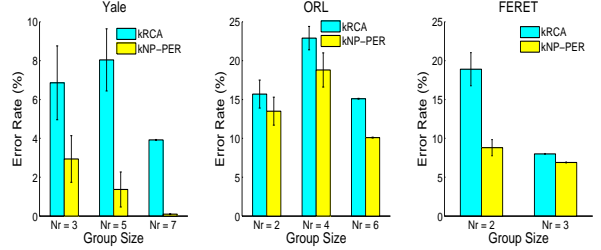


Figure 4. Performance comparison for the nonlinear versions of RCA (kRCA) and NP-PER (kNP-PER) on different test sets.

Table 1. Results of K-means clustering applied to Yale data for different algorithms and three different group sizes. Results are averaged over 100 random K-means initializations for each of the 100 random group samples.

N_r	kRCA		kNP-PER	
	Purity (%)	Accuracy (%)	Purity (%)	Accuracy (%)
3	94.2 (± 3.7)	96.0 (± 2.2)	98.3 (± 2.4)	98.8 (± 1.7)
5	96.9 (± 3.0)	97.9 (± 2.0)	99.7 (± 1.2)	99.8 (± 0.7)
7	93.5 (± 4.2)	95.7 (± 2.5)	97.7 (± 2.9)	98.4 (± 1.6)

the experiments. The dimension m for RCA was chosen for each dataset in the same way as for the Yale set. The performance of the different methods is compared on both datasets in Figure 3 and Figure 4. The plots show better performance for NP-PER and kNP-PER in comparison to the other methods for different group sizes.

5.2. Clustering

In this section, similar to [11], we compare the performance of K-means clustering on the Yale dataset. First, the data is transformed nonlinearly using kRCA and kNP-PER, and then clustering is performed in the transformed space. The K-means clusters are learned by fixing the number of clusters, K , to be the same as the number of individuals in the training set. Thus, the ideal clustering should assign all the images of an individual to a single cluster. Similarly, all the images in a cluster should be from one individual.

Similar to [11], we compare the clustering performance using two measures, *Purity* and *Accuracy*. Purity measures the frequency of data belonging to the same cluster sharing the same class labels, while Accuracy measures the frequency of data from the same class appearing in a single cluster. Thus, the ideal clustering will have 100% Purity and 100% Accuracy.

The clustering results for the two algorithms are shown in Table 1. kNP-PER achieves higher Purity and Accuracy than kRCA for all group sizes. In comparison to these techniques, PCA yields very low rates of Purity (70.6% (± 8.5)) and Accuracy (82.2% (± 8.9)), since it does not exploit the partial equivalence relations in the data.



Figure 5. An illustration of group extraction from videos. Each row shows 10 example frames from a group. The left most image displays the first frame while the rest of the images display the extracted faces from the subsequent frames warped using thin-plate splines. Note the variations in expression and pose within and across chunks.

6. Experiments on web data

6.1. Face recognition in videos

To evaluate the null-space method on a real world dataset, we applied it to the problem of face recognition in videos. For this, we used a corpus of 17 videos downloaded from the Web that contained 8 different individuals. For each video, first a face detector is applied to each frame. Next, a landmark detector finds 13 landmarks on each face, that are used to warp the face to a standard size of 48×48 pixels using thin-plate splines. A sequence of frames, for which the face does not move more than a threshold between frames, is taken as a data group. Each such group contains pose and expression variations of an individual’s face. Example face variations in a group for some of the individuals used in this study are shown in Figure 5. Each group is also manually labeled with the individual’s name for testing purposes.

Overall, 960 frames were used for training (120 frames per individual), and 1440 frames for testing (180 frames per individual). While training, the group sizes were varied to be 7, 15, 30 and 60. Note that, for these sizes, the training set always contained more than one group for each individual. The training times for linear NP-PER and RCA were 244 sec and 237 sec, respectively. Their kernel versions were faster to train due to the fact that $N < d$, with kRCA taking 183 sec and kNP-PER 198 sec. Figure 6 compares the performance of both linear and nonlinear versions of RCA and NP-PER. It is interesting to note that NP-PER gives the least error for $N_r = 7$, while kNP-PER performs the best for all other group sizes.

6.2. Image retrieval

The final set of tests was conducted on an image retrieval task. In this task, the image feature vector is projected using kRCA or kNP-PER, and then the K-nearest neighbor scheme is used for retrieval. The dataset consisted of 3000

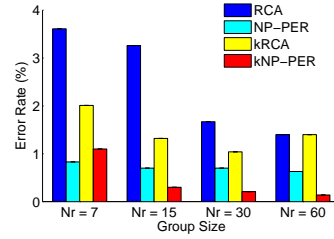


Figure 6. Comparison of linear and kernel NP-PER and RCA techniques on face recognition in web videos. NP-PER and kNP-PER perform better than RCA and kRCA, respectively for all N_r .

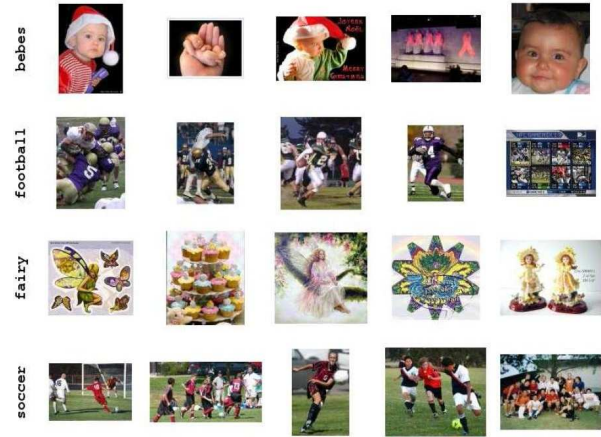


Figure 7. Example query terms and some of the images returned by a web search engine, that composed the retrieval dataset.

images, obtained using 100 popular text queries (30 images per query) from Google Image Search. A few query terms and example images returned by the search engine are shown in Figure 7. This is a very challenging set as the images for a query vary significantly in their visual appearance. Furthermore, images from different queries may appear very similar, e.g., for *football* and *soccer*.

From each image, two types of features are extracted. The global color histogram in the LUV space is used for color features. A bin size of 16 in each dimension yields a 4096 dimensional color feature vector. To represent texture, the input image is divided into a grid of 16×16 blocks. Then, the first four DCT coefficients are kept for each block, yielding a 1024 dimensional texture feature vector. The combination of both types of features gave a $d = 5120$ dimensional feature vector.

To visualize the complexity of the dataset with increasing number of queries, we conducted experiments on reduced datasets containing 20, 50 and 70 queries in addition to the full set of 100 queries. Each dataset is split into a training and a test set of equal size. Random groups of size $N_r = 5$ are generated for each query for training. The training time on the 100 query set was 242 sec for kRCA and 263 sec for kNP-PER. During testing, K nearest neighbors

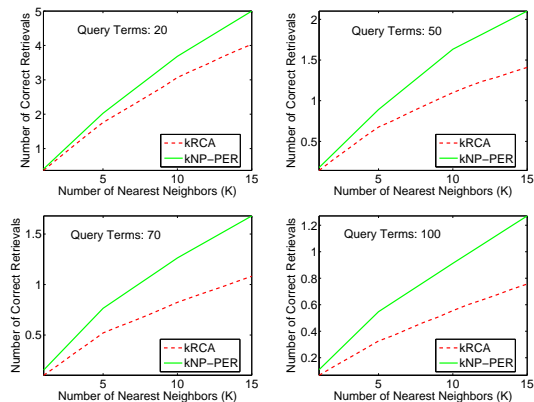


Figure 8. Average number of correct retrievals in top K neighbors for four datasets corresponding to increasing number of query terms (or classes): 20, 50, 70 and 100. The retrieval problem becomes harder as the number of query terms increase.

are found for each test image and the retrieval performance is measured by the average number of correctly retrieved images out of these K images. Figure 8 compares the retrieval performance for kRCA and kNP-PER on the four datasets. The plots indicate that as the number of queries are increased from 20 to 100, the number of average retrievals falls from 5 to 1.2 in the top 15 matches. This clearly shows that the retrieval task becomes more complex as the number of queries is increased. kNP-PER performs better than kRCA, and the advantage is more evident for larger K .

7. Conclusions and future work

We have presented a null-space based technique which learns discriminative linear or nonlinear transformations using only weakly-labeled data. The proposed approach has been shown to outperform the state-of-the-art techniques for learning with partial equivalence relations on recognition, clustering and retrieval tasks. In the future, we intend to apply the proposed technique for large scale retrieval and classification of web data using partial equivalence relations. For this, we plan to explore fast iterative matrix eigen-analysis methods to handle large amounts of data.

Acknowledgments

Our thanks to Dennis Strelow for providing the image-retrieval dataset, Vladimir Pavlovic for feature extraction code, and Hartwig Adams for face landmark detection code.

References

- [1] A. Bar-Hillel, T. Hertz, N. Sental, and D. Weinshall. Learning distance functions using equivalence relations. *In Proc. Int. Conf on Machine Learning (ICML)*, 2003.
- [2] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear

- projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
- [3] L. F. Chen, H. Y. M. Liao, J. C. Lin, M. T. Ko, and G. J. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10), 2000.
- [4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley, New York, 2001.
- [5] G. Golub and C. van Loan. *Matrix computations*. The Johns Hopkins University Press, London, 1996.
- [6] T. Hertz, A. Bar-Hillel, and D. Weinshall. Learning distance functions for image retrieval. *In Proc. IEEE Conf on Computer Vision and Pattern Recognition (CVPR)*, 2004.
- [7] R. Huang, Q. S. Liu, H. Q. Lu, and S. D. Ma. Solving small sample size problem of LDA. *In Proc. Int. Conf. on Pattern Recognition (ICPR)*, 3:29–32, 2002.
- [8] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Muller. Fisher discriminant analysis with kernels. *IEEE Workshop on Neural Networks for Signal Processing*, 1999.
- [9] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [10] N. Sental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing gaussian mixture models with EM using equivalence constraints. *In Adv. in NIPS*, 2003.
- [11] N. Sental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. *In Proc. European Conference on Computer Vision (ECCV)*, pages 776–790, 2002.
- [12] I. W. Tsang, P. M. Cheung, and J. T. Kwok. Kernel relevant component analysis for distance metric learning. *In Proc. Int. Joint Conf. on Neural Networks (IJCNN)*, 2005.
- [13] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591, 1991.
- [14] K. Wagstaff, C. Cardie, S. Rogers, and S. Scroedl. Constrained k-means clustering with background knowledge. *In Proc. Int. Conf. on Machine Learning (ICML)*, 2001.
- [15] L. Wei, S. Z. Li, Y. Wang, and T. Tan. Null space-based kernel fisher discriminant analysis for face recognition. *Int. Conf. on Automat. Face and Gesture Recog.*, 2004.
- [16] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 988–995, 2004.
- [17] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side information. *In Adv. NIPS*, 2002.
- [18] J. Ye and T. Xiong. Computational and theoretical analysis of null space and orthogonal linear discriminant analysis. *Journal of Machine Learning Research*, (7), 2006.