Journal of Classification 27:307-321 (2010)
DOI: 10.1007/s00357-010-9059-3

Dimensionality Reduction on the Cartesian Product of Embeddings of Multiple Dissimilarity Matrices

Zhiliang Ma

Applied Mathematics & Statistics, Johns Hopkins University

Adam Cardinal-Stakenas

Applied Mathematics & Statistics, Johns Hopkins University

Youngser Park

Center for Imaging Science, Johns Hopkins University

Michael W. Trosset

Department of Statistics, Indiana University

Carey E. Priebe

Applied Mathematics & Statistics, Johns Hopkins University

Abstract: We consider the problem of combining multiple dissimilarity representations via the Cartesian product of their embeddings. For concreteness, we choose the inferential task at hand to be classification. The high dimensionality of this Cartesian product space implies the necessity of dimensionality reduction before training a classifier. We propose a supervised dimensionality reduction method, which utilizes the class label information, to help achieve a favorable combination. The simulation and real data results show that our approach can improve classification accuracy compared to the alternatives of principal components analysis and no dimensionality reduction at all.

Keywords: Dissimilarity representation; Multidimensional scaling; Dimensionality reduction; Principal components analysis; Linear discriminant analysis.

Support for this effort was provided in part by the Office of Naval Research, the Acheson J. Duncan Fund for the Advancement of Research in Statistics, and Raytheon Corporation. The image and caption dataset used in this article was provided by Dr. Jeffrey L. Solka of NSWC Dahlgren.

Author Addresses: C. Priebe, Applied Mathematics and Statistics, Johns Hopkins University, 302 Whitehead Hall, 3400 North Charles Street, Baltimore, MD USA 21218-2682, e-mails: cep@jhu.edu; zhiliang.ma@gmail.com; cardinal.stakenas@gmail.com; youngser@jhu.edu; mtrosset@indiana.edu.

Published online 8 October 2010

1. Introduction

Most traditional statistical pattern recognition techniques rely on objects represented by points in a feature (vector) space. In this space, classifiers are developed to best separate the objects of different classes. As an alternative to the feature-based representation, the dissimilarity representation describes objects by their interpoint comparisons. The dissimilarity representation has attracted substantial interest in various areas (Clarke 1993; Maa, Pearl, and Bartoszyński 1996; Priebe 2001; Anderson and Robinson 2003; Pękalska and Duin 2005).

Because there are many ways to compare two objects—for example, the L^p-distances—it is possible to construct many dissimilarity representations. Ideally, each dissimilarity representation captures different aspects of the underlying patterns. Consequently, combining multiple dissimilarity representations can be beneficial. One way to combine multiple dissimilarity representations is via the Cartesian product of their embeddings. The high dimensionality of this embedding product space implies the necessity of dimensionality reduction before training a classifier. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ denote the original (or transformed) d-dimensional data matrix and let $\mathbf{X}_A$ denote a submatrix that contains only the columns of $\mathbf{X}$ with indices in $A \subseteq \{1, \ldots, d\}$. The problem of dimensionality reduction is to find an index set A of size $p \triangleq |A| < d$ such that the classification error based on the p-dimensional data $\mathbf{X}_A$ is small.

Principal component analysis (PCA) is the most widely used method for dimensionality reduction, but it does not take into account the class label information, which is crucial for extracting discriminative features. Linear discriminant analysis (LDA) is also broadly used for dimensionality reduction (or as a classifier), and it uses class label information. However, relatively small sample size (compared to dimensionality d) may cause LDA's performance to decrease when adding more dimensions, even though the extra dimensions contain discriminative information. Trunk (1979) affirmed this phenomenon by investigating an illuminating simple example. Chang (1983), Dillon, Mulani, and Frederick (1989), Kshirsagar, Kocherlakota, and Kocherlakota (1990) all established a statistic θ_k for the kth principal component (PC) and use θ_k to decide which PCs should be used in discrimination. Jolliffe, Morgan, and Young (1996) observed that, for two-class problem, the sample estimate $\hat{\theta}_k$ is equivalent to a t-statistic and the hypothesis test based on θ_k to decide whether or not to include the kth PC is equivalent to the t-test with null hypothesis H_{0k} that there is no difference between the two class means. The statistic θ_k is useful in determining the order of importance of PCs in separating the two populations. However, the best p individual PCs do not necessarily constitute the best subset of

p PCs (e.g., Toussaint 1971). Takemura (1985) proposed a decomposition of the Hotelling's T^2 statistic by projecting data onto the principal axes of the pooled covariance matrix, then calculating t-statistic t_k for the kth PC. Takemura suggested using the first p PCs ("... to look at $t_1^2, t_1^2 + t_2^2, \ldots$") and briefly mentioned "If one has a prior idea about the importance of various axes, a weighted sum of t_k^2, $T_w^2 = \sum_{k=1}^{d} w_k t_k^2$, might be considered." (Equation 4.5, Takemura 1985). Following Takemura's framework for decomposing Hotelling's T^2, we propose choosing the p PCs that correspond to the p largest values of $J_k \equiv |t_k|$. We show that, under the assumption of mixture of two multivariate Gaussian distributions with equal covariance matrices, the best p individual PCs coincide with the best subset of p PCs. We demonstrate the use of this approach with simulation, image and caption data. The results show that, for classification, our approach outperforms both PCA and no dimensionality reduction.

In Section 2, we describe the background of combining multiple dissimilarity representations—in particular, via the Cartesian product of their embeddings. Section 3 details the proposed supervised dimensionality reduction method. Simulation and real data examples are presented in Section 4. Section 5 provides conclusions and how to extend our approach to suit a problem with more than two classes.

2. Combining Multiple Dissimilarity Representations

A *dissimilarity measure* is a function $\delta : \Xi \times \Xi \to \mathbb{R}_+$ with $\delta(z_1, z_2) \geq 0$, $\delta(z_1, z_2) = \delta(z_2, z_1)$ and $\delta(z, z) = 0$. It measures the magnitude of difference between two objects. Asymmetric functions are also of interest, but this is beyond the scope of this paper. Notice that in most cases $\Xi = \mathbb{R}^d$. However, we wish to leave open the possibility for applications where the original data are infinite dimensional, graph-valued, or occupying some other nonstandard space. In cases where we observe only the dissimilarities, it will still be useful to imagine that they are computed from a set of Ξ-valued vectors—the "measurements" of objects. The *dissimilarity representation* of a set of objects is obtained by computing δ on each pair of objects. It is expressed as a nonnegative and symmetric matrix Δ with all zero diagonal entries.

Let $\delta_1, \ldots, \delta_K$ denote K dissimilarity measures. Given n object-label pairs $(z_i, y_i) \in \Xi \times \{0, 1\}$, $i = 1, \ldots, n$, let

$$\Delta_k = [\delta_k(z_i, z_j)], \; k = 1, \ldots, K,$$

be the corresponding K dissimilarity matrices. The task is to combine these K dissimilarity matrices in order to obtain superior (compared to any one of the Δ_k alone) performance in classification.

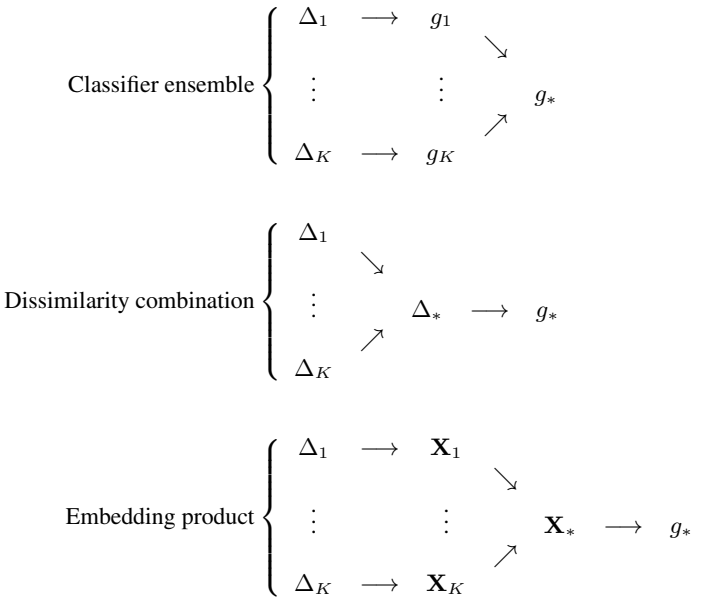


Figure 1. $\Delta_1, \cdots, \Delta_K$ are K dissimilarity matrices. "Classifier ensemble" combines separate classifiers g_k that were trained on individual dissimilarity matrices Δ_k; "dissimilarity combination" trains one classifier g_* from a single combined dissimilarity matrix Δ_*; "embedding product" embeds each Δ_k into $\mathbf{X}_k$ and combines those embeddings to obtain $\mathbf{X}_*$, and then a classifier is trained.

As illustrated in Figure 1, there are (at least) three possible ways to combine dissimilarities: (1) "classifier ensemble" combines separate classifiers that were trained on individual dissimilarity matrices; (2) "dissimilarity combination" trains one classifier from a single combined dissimilarity matrix; (3) "embedding product" embeds each dissimilarity matrix first, then combines the embeddings to build a classifier. The process of embedding an $n \times n$ dissimilarity matrix, $\Delta = [\delta(z_i, z_j)]$, involves finding a configuration of points, $x_1, \ldots, x_n$, in a normed linear space, such that the interpoint distances, $\|x_i - x_j\|$, approximate the $\delta(z_i, z_j)$. When the normed linear space is Euclidean, embedding is widely known as multidimensional scaling. The configuration of points, here denoted $\mathbf{X}$, is called the *embedding* of Δ. (In this paper, we use the bold $\mathbf{X}$ to denote an $n \times d$ data matrix, where each row corresponds to a d-dimensional observation; and we use X to denote a d-dimensional random vector.) In this work, we focus on the "embedding product" approach and discuss in depth how to perform dimensionality reduction in the Cartesian product space.

The key to the "embedding product" approach is to determine the "right" embedding dimensionality d_k of each Δ_k and the dimensionality of the Cartesian product space. Miller et al. (2009) gave an example in the $K = 2$ case. They embedded Δ_1 and Δ_2 into $\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$, ranging d_1 and d_2 from 0 to some maximum d_1^{max} and d_2^{max}, respectively. (In their case, $d_1^{max} = d_2^{max} = 15$.) They then built a classifier for each possible combination of (d_1, d_2), and obtained an estimate of the classification error L_{d_1,d_2}. In the end, they chose $(\hat{d}_1, \hat{d}_2) = \arg\min L_{d_1,d_2}$. This method is necessarily suboptimal as it includes all the first $\hat{d}_1$ and $\hat{d}_2$ PCs, but ignores higher rank PCs, which may contain discriminative information. It also becomes unwieldy for $K > 2$.

An alternative way to implement the "embedding product" approach is to embed each Δ_k into $\mathbf{X}_k \in \mathbb{R}^{n \times d_k}$, and construct a classifier in the Cartesian product space $[\mathbf{X}_1, \ldots, \mathbf{X}_K]$. The dimensionality of the product space could be very high, especially when K is large. Dimensionality reduction is needed to alleviate the "curse of dimensionality," the phenomenon that the number of data points needed to learn a classifier increases exponentially with the dimension of the representation space (Bellman 1961; Bishop 1995).

3. Dimensionality Reduction

PCA is widely used to create low-dimensional representations of high-dimensional data. PCA constructs a new coordinate system in such a way that the span of the first k principal components (PCs) is the k-dimensional linear subspace that best summarizes (in the sense of squared error) the data. PCA is unsupervised, so applying PCA within classes may result in different PCs than applying PCA to the entire data set. Furthermore, the PCs that best summarize variation in the data may not be the dimensions that best discriminate between classes, as in the case of "parallel cigars." In contrast to PCA, LDA uses the class labels to find the best dimensions for class discrimination. Unfortunately, LDA may perform badly in high-dimensional spaces (cf. Trunk 1979). To address this difficulty, Belhumeur, Hespanha, and Kriegman (1997) proposed a two-step procedure (LDA $\circ$ PCA) in which PCA is first used to reduce dimensionality, after which LDA is used to train a linear classifier; however, if the PCA step discards dimensions that are important for discrimination, then LDA $\circ$ PCA may also perform badly. To remedy this failing, we develop an alternative PCA step that we call the J-function procedure. The essential idea is to extract (class-conditional uncorrelated) PCs based on their ability to discriminate, rather than based on how much variation they contain. As explained in Figure 2, LDA $\circ J$ improves on LDA $\circ$ PCA.

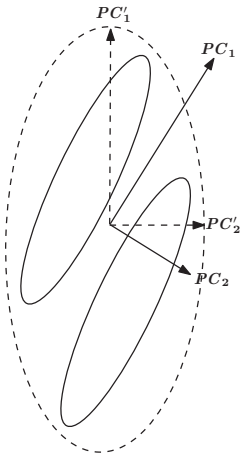


Figure 2. The solid ellipses represent the data from the two classes. The dashed ellipse represents the entire dataset, on which performing PCA reports PC_1' and PC_2' as the 1st and 2nd principal components, respectively. The J-function approach first finds the principal components PC_1 and PC_2 by performing eigenvalue decomposition on the pooled covariance matrix. It then computes the J value, a measure of discriminative power, for each PC and reorders the PCs by the J values associated with them. PCs with larger J values will have higher rank in the order. For this dataset $J_1 < J_2$ (J_i is the J value of the PC_i). Therefore the final first and second PCs generated by the J-function approach are PC_2 and PC_1, respectively. Notice that for low dimensional data, the J-function approach is essentially the same as LDA. For high dimensional data, where LDA has problems, one can use the two-step approach, LDA $\circ$ J.

3.1 J-function

Consider data matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{n\times d}$ and class vector $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ with $\{0, 1\}$ entries. The goal is to find a p-dimensional ($p < d$) representation of $\tilde{\mathbf{X}}$ that contains the most class information. The J-function procedure can be described via the following steps:

1. Compute the pooled sample covariance matrix $\mathbf{S} = \pi\mathbf{S}_1 + (1-\pi)\mathbf{S}_0$, where $\pi = \sum_{i=1}^{n} y_i/n$ and $\mathbf{S}_j$ is the sample covariance matrix for class j.
2. Perform eigenvalue decomposition on $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ and transform $\tilde{\mathbf{X}}$ to $\mathbf{X} = \tilde{\mathbf{X}}\mathbf{U}$. (Assume that the columns of $\tilde{\mathbf{X}}$ have been centered to have mean zero.)
3. Compute the J value for the ith dimension of $\mathbf{X}$

$$J_i = \begin{cases} |\boldsymbol{m}_{\mathbf{1}_i} - \boldsymbol{m}_{\mathbf{0}_i}|/\sqrt{\lambda_i}, & \lambda_i > 0, \\ 0, & \lambda_i = 0, \end{cases}$$

where $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$ are the sample means of classes 0 and 1, respectively, and λ_i is the ith largest eigenvalue of $\mathbf{S}$.

4. Obtain $\mathbf{X}^J$ by reordering the dimensions of $\mathbf{X}$ according to the J values—dimensions with larger values have higher rank in the order. Let $\mathbf{X}_p^J$ be the first p dimensions of $\mathbf{X}^J$.

Then $\mathbf{X}_p^J$ is the p-dimensional representation of $\tilde{\mathbf{X}}$ obtained by the J-function approach. In summary, this approach first projects data onto the principal axes of the pooled covariance matrix to obtain conditionally uncorrelated (given class label Y) PCs, then ranks them by a quantity J, which is the absolute value of a t-statistic, and finally includes only these PCs with large J values. Devroye, Györfi, and Lugosi (1996, p. 566) sketched a similar idea to rank (class) independent Gaussian distributed features. We show in the following theorem that, under the assumption of mixture of two multivariate Gaussian distributions with equal covariance matrices, $\mathbf{X}_p^J$ contains the most class information among a collection of p-dimensional projections of $\tilde{\mathbf{X}}$. That is, for the transformed data $\mathbf{X}$, the best p individual PCs constitute the best subset of p PCs.

Theorem. *Suppose that (X, Y) is distributed as F_{XY}, where $X : \Omega \to \mathbb{R}^d$, Y is Bernoulli distributed with parameter π, and that the conditional distribution of $X|Y = j$ is $N(\boldsymbol{\mu}_j, \Sigma)$. Let $f : \mathbb{R}^d \to \mathbb{R}^p$ be the function that projects X onto the space spanned by any p of the d eigenvectors of Σ, where $p < d$, and let f^* be the projection function deduced from the above J-function procedure. If $L^*_{f(X)}$ and $L^*_{f^*(X)}$ denote the Bayes error probabilities for $(f(X), Y)$ and $(f^*(X), Y)$ respectively, then*

$$L^*_{f^*(X)} \leq L^*_{f(X)}. \tag{1}$$

Proof. We assume that $\Sigma = I_d$, where the dimensions of X are ordered (from largest to smallest) by the magnitudes of the elements of $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$. To see that this assumption entails no loss of generality, first note that if Σ is of less than full rank, then X can be projected into a lower dimensional space in which the covariance matrix is of full rank. Second, we can assume that $\Sigma = I_d$ because there exists a matrix A for which $(AX|Y = j) \triangleq (X_A|Y = j) \sim N(A\boldsymbol{\mu}_j, I_d)$. Hence, any linear projection function of X can be written as $f(X) = f(A^{-1}X_A) \triangleq f_A(X_A)$. Third, assuming that $\Sigma = I_d$, we can further assume that the dimensions are in any prescribed order—we simply apply the same argument with A chosen to be a suitable permutation matrix.

Then the projection $f^*(X) = T^*X$ chooses the first p dimensions of X. That is, T^* is a $p \times d$ matrix with all 1's on the diagonal of its leftmost $p \times p$ block and 0's elsewhere; and the projection $f(X) = TX$ chooses any p dimensions of X. That is, T has same columns as T^* does, but with

different order. By the previous assumptions, we have

$$\|T^*\boldsymbol{\mu}_1 - T^*\boldsymbol{\mu}_0\| \geq \|T\boldsymbol{\mu}_1 - T\boldsymbol{\mu}_0\|,$$

which implies

$$L^*_{T^*X} \leq L^*_{TX}$$

and

$$L^*_{f^*(X)} \leq L^*_{f(X)}.$$

■

In practice, the sample covariance matrix $\mathbf{S}_j$ usually is not an accurate and reliable estimator of the population covariance matrix, especially when the data have a large number of dimensions but contain comparatively few samples. This will decrease J-function's power in determining discriminative dimensions. To alleviate this problem, in the following experiments section, we used Schäfer and Strimmer's (2005) shrinkage estimation of covariance matrix to obtain $\mathbf{S}_j$.

4. Experiments

4.1 Simulation Experiment

To illustrate the J-function approach and its advantages, we conduct a simple simulation experiment. Let $F_{XY} = \pi N(\boldsymbol{\mu}, \Sigma) + (1-\pi)N(-\boldsymbol{\mu}, \Sigma)$, where

$$\pi = \frac{1}{2},\ \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix},\ \boldsymbol{\mu}_a = \boldsymbol{\mu}_b = (1,1,1,1,1,0,\cdots,0)^T \in \mathbb{R}^{40},$$
$$\Sigma = \begin{pmatrix} \Sigma_a & \mathbf{0} \\ \mathbf{0} & \Sigma_b \end{pmatrix},\ \Sigma_a = \text{diag}(1,\ldots,40),$$
$$\Sigma_b(i,j) = \frac{\sqrt{ij}}{2^{|i-j|}},\ i,j = 1,\ldots,40.$$

Notice that the two multivariate Gaussian distributions have the same covariance matrix Σ, and the only difference is in the means. The reason we construct Σ in this special form is that we try to simulate two different data sources, analogous to the Cartesian product of embeddings of $K = 2$ dissimilarity matrices.

We randomly draw $2n$ samples $\mathbf{X} = [x_1, \ldots, x_{2n}]^T$ from F_{XY}, with the first n samples as training data and the rest as testing data. We then perform dimensionality reduction, build LDA based on the training data and then classify the testing observations. For comparison, we consider PCA and the J-function method in the dimensionality reduction step. In addition,

we let p, the reduced dimensionality, range from 1 to 80 ($p = 80$ means no dimensionality reduction). Notice that the J-function approach is a supervised dimensionality reduction method. That is, it utilizes the class label information. We perform two experiments: in the first one we use only the class labels of the training observations, and in the second one we use the class labels of both the training and the testing observations. The following LDA step remains the same for both experiments. Because the dimensionality reduction step (although not the LDA step) in the second experiment uses the testing class labels, this experiment leads to an overly optimistic classification error. It provides a (meaningful) lower bound on the error from the first (valid) experiment. We call the dimensionality reduction method used in the first experiment the J-function approach and that used in the second experiment the $\underline{J}$-function approach. We use L_J and $L_{\underline{J}}$ to denote the classification errors corresponding to the J-function and $\underline{J}$-function.

We repeat the above process 100 times each for three different sample sizes: $n = 100$, $n = 200$ and $n = 400$. Let $\bar{L}_P(p)$, $\bar{L}_J(p)$ and $\bar{L}_{\underline{J}}(p)$ denote the means of the estimated classification errors resulting from the p-dimensional data, which are obtained through PCA, the J-function and $\underline{J}$-function procedures, respectively. Let $\bar{L}_\emptyset$ denote the mean of the estimated classification error when using LDA only, that is, no dimensionality reduction. This simulation experiment shows that (1) for classification, for fixed reduced dimensionality $p < 80$, the J-function procedure outperforms PCA and no dimensionality reduction: $\bar{L}_{\underline{J}}(p) < \bar{L}_J(p) < \bar{L}_\emptyset \leq \bar{L}_P(p)$, for all $p < 80$; (2) the J-function procedure works better than PCA, when both use optimal reduced dimensionalities: $\min_p \bar{L}_{\underline{J}} < \min_p \bar{L}_J < \min_p \bar{L}_P$; (3) for classification, the $\underline{J}$-function procedure provides a lower bound on the error, and the difference between the J- and $\underline{J}$-procedure, $\bar{L}_J(p) - \bar{L}_{\underline{J}}(p)$, decreases as the sample size increases. We plot the results in Figure 3.

4.2 The Tiger Data

In this section, we present an example of combining image and caption data. The data are 140,577 images and captions collected from the Yahoo! Photos website. We selected 1,600 pairs by using the query word "tiger" on captions. The "tiger" data were manually labeled into 6 classes based only on captions (see Figure 4). For simplicity we consider the problem of discriminating between the two classes of "Tiger Woods" and "Tamil Tigers".

The image, text, and joint image-text spaces are rather complicated, so there is no simple way to combine them directly. We use the first and second order pixel derivatives (Gonzalez and Woods 2007; Jain 1989) on the images, and the mutual information (Lin and Pantel 2002) on the captions to

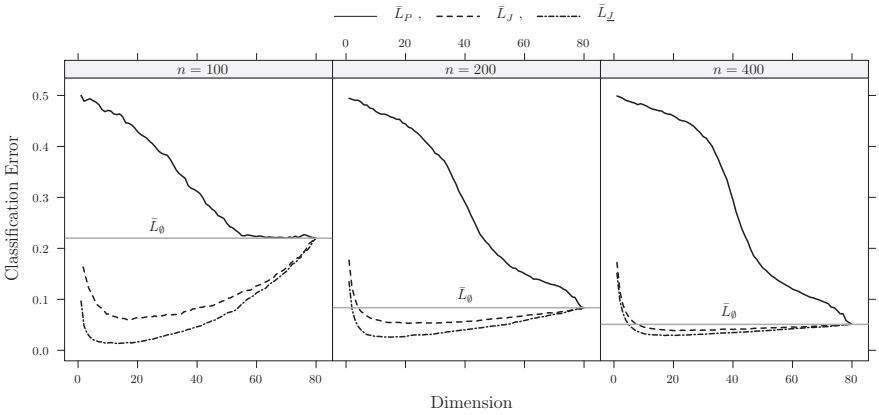
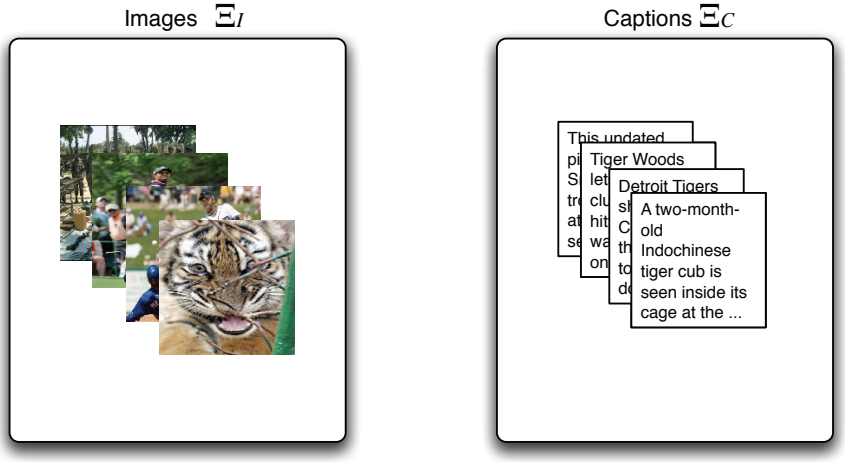


Figure 3. Let $\bar{L}_P(p)$, $\bar{L}_J(p)$ and $\bar{L}_{\underline{J}}(p)$ denote the mean of the estimated classification errors resulting from the p-dimensional data, which are obtained through PCA, the J- and $\underline{J}$-function procedure, respectively. Let $\bar{L}_\emptyset$ denote the mean of the estimated classification error when using LDA only, that is no dimensionality reduction. These plots depict that (1) $\bar{L}_{\underline{J}}(p) < \bar{L}_J(p) < \bar{L}_\emptyset \leq \bar{L}_P(p)$, for all $p < 80$; (2) $\min_p \bar{L}_{\underline{J}} < \min_p \bar{L}_J < \min_p \bar{L}_P$; (3) $\bar{L}_J(p) - \bar{L}_{\underline{J}}(p)$ decreases as the sample size increases.



Label	**#**
Animal tiger	148
Detroit Tigers baseball team	145
Tiger Woods the golfer	897
Tamil Tigers soldiers of Sri Lanka	330
Leicester Tigers rugby team	48
Others	32

Figure 4. The "tiger" data. Each observation consists of an image/caption pair.

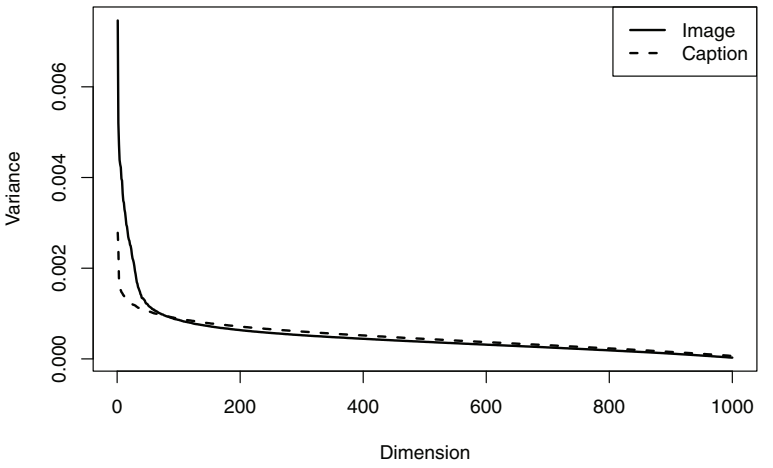


Figure 5. "Tiger" data. Using the classical multidimensional scaling to embed both Δ_C and Δ_I into 1000-dimensional Euclidean space. The scree plot depicts the variance for each dimension.

extract features from each space. We then compute R_{ij}, the random forest proximity (Breiman 2001), for each pair of observations and use $1 - R_{ij}$ as the dissimilarity measure to generate two dissimilarity matrices, Δ_C and Δ_I. Classical multidimensional scaling (CMDS) (Torgerson 1952; Gower 1966; Borg and Groenen 2005) is used to embed Δ_C into $\tilde{\mathbf{X}}_C \in \mathbb{R}^{n \times d(n)}$ and Δ_I into $\tilde{\mathbf{X}}_I \in \mathbb{R}^{n \times d(n)}$. We used $d(n) = 1000$. Because the coordinates of the embedding constructed by CMDS are its PCs, it is easy to perform PCA. Figure 5 displays a scree plot of variances. The automatic dimensionality selection of Zhu and Ghodsi (2006) was used to determine reduced dimensionalities of $d_C = 473$ for caption and $d_I = 152$ for image. These choices err on the side of anti-parsimony, but further dimensionality reduction in the Cartesian product space will follow. For comparison, we considered also the J-function on $\tilde{\mathbf{X}}_C$ and $\tilde{\mathbf{X}}_I$ (Zhu and Ghodsi's approach was used also to determine reduced dimensionality). For the Cartesian product, we separately performed PCA, the J-function and $\underline{J}$-function to reduce the dimensionality. A linear classifier was built on caption alone, image alone and their combination, respectively. Leave-one-out cross-validation was used to estimate classification errors. Figure 6 shows the above procedures.

Table 1 reports classification errors for several procedures. The results suggest that (i) the two step procedure LDA $\circ$ J works better than LDA only (no dimensionality reduction) and than LDA $\circ$ PCA; (ii) LDA $\circ$ J is better than LDA $\circ$ PCA$'$, which is the same as LDA $\circ$ PCA, except using the reduced dimensionalities determined by the J-function; (iii) LDA $\circ$ $\underline{J}$ is apparently better than the other procedures, but recall that $\underline{J}$ uses testing class

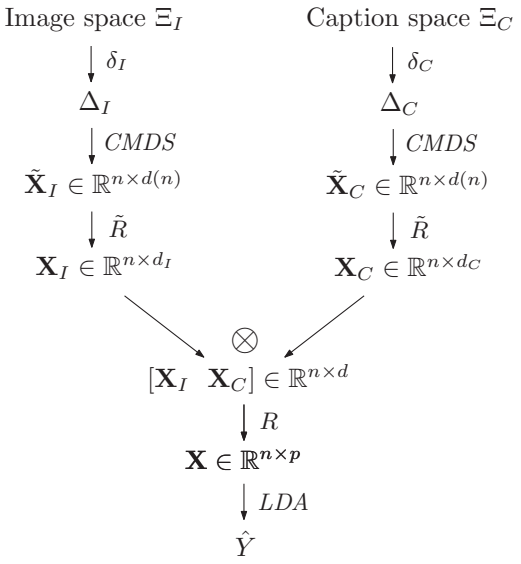


Figure 6. "Tiger" data. We combined image and caption data using dissimilarity representation: image and caption data were transformed into dissimilarity matrices Δ_I and Δ_C, which were then embedded into $d(n)$-dimensional Euclidean space. Dimensionality reduction procedures $\tilde{R}$ and R were performed on each embedding and then on the Cartesian product, respectively. Finally, a linear classifier was trained. We considered $\tilde{R} \in \{\text{PCA}\ , J\text{-function}\}$ and $R \in \{\text{PCA}, J, \underline{J}, \emptyset\}$, where $\emptyset$ means no dimensionality reduction.

Table 1. "Tiger" data. We use the two-step approach LDA $\circ$ R—perform dimensionality reduction procedure R and then train linear classifier on the low-dimensional data—together with leave-one-out cross validation to estimate classification error. The notation $\emptyset$ means no dimensionality reduction and PCA′ is PCA but using the dimensionalities determined by the J-function procedure. The bar on dimensionality means that the corresponding number is the average of dimensionalities used in leave-one-out cross validation by J-function.

Data	$\boldsymbol{R}$	$\mathbf{X} \in \mathbb{R}^p$	p	**Error**
Image: $\tilde{\mathbf{X}}_I \in \mathbb{R}^{n\times 1000}$	PCA	$\mathbf{X}_I^P$	152	0.1491
	J-function	$\mathbf{X}_I^J$	$\overline{62}$	0.1133
Caption: $\tilde{\mathbf{X}}_C \in \mathbb{R}^{n\times 1000}$	PCA	$\mathbf{X}_C^P$	473	0.1883
	J-function	$\mathbf{X}_C^J$	$\overline{384}$	0.1345
Combination: $\mathbf{X}^P = [\mathbf{X}_I^P \ \mathbf{X}_C^P]$	$\emptyset$		625	0.1557
	PCA		160	0.1125
	PCA′		$\overline{205}$	0.1182
	J-function		$\overline{205}$	0.0815
	$\underline{J}$-function		71	0.0171
Combination: $\mathbf{X}^J = [\mathbf{X}_I^J \ \mathbf{X}_C^J]$	J-function		$\overline{186}$	0.0864

Table 2. "Tiger" data. McNemar's test was used to compare the various dimensionality reduction procedures $R \in \{\emptyset, \text{PCA}, J, \underline{J}\}$. The alternative hypothesis H_A is listed in the first column and the corresponding null hypothesis replaces "$<$" with "$\geq$". We use $L(\mathbf{X})$ to denote the LDA leave-one-out cross validation classification error based on data $\mathbf{X}$, and use $R(\mathbf{X})$ to denote the low-dimensional data obtained by R. The definitions of various forms of $\mathbf{X}$ can be found in Table 1. These p-values, together with Table 1, show that (i) LDA $\circ$ J works better than LDA only (no dimensionality reduction) and better than LDA $\circ$ PCA; (ii) LDA $\circ$ J is better than LDA $\circ$ PCA$'$, which is the same as LDA $\circ$ PCA except using the reduced dimensionalities determined by the J-function; (iii) LDA $\circ$ $\underline{J}$ is apparently better than the other procedures, but recall that $\underline{J}$ uses testing class labels (the error rate for LDA $\circ$ $\underline{J}$ *is* a meaningful lower bound on the error rate of LDA $\circ$ J).

H_A	p-value
$L(J(\tilde{\mathbf{X}}_I)) < L(\text{PCA}(\tilde{\mathbf{X}}_I))$	4.803e-07
$L(J(\tilde{\mathbf{X}}_C)) < L(\text{PCA}(\tilde{\mathbf{X}}_C))$	5.215e-04
$L(\text{PCA}(\mathbf{X}^P)) < L(\emptyset(\mathbf{X}^P))$	4.643e-05
$L(J(\mathbf{X}^P)) < L(\text{PCA}(\mathbf{X}^P))$	8.095e-05

labels (the error rate for LDA $\circ$ $\underline{J}$ *is* a meaningful lower bound on the error rate of LDA $\circ$ J). We used McNemar's test to validate these statements and show the results in Table 2.

We also applied the two-step procedure LDA$\circ$ R ($R \in \{\emptyset, \text{PCA}, J\text{-function}\}$) on the two classes of "animal" versus "baseball". The resulted leave-one-out cross-validation classification errors are: $L(\emptyset(\mathbf{X})) = 0.0751$, $L(\text{PCA}(\mathbf{X})) = 0.0648$ and $L(J(\mathbf{X})) = 0.0444$. At level of significance $\alpha = 0.05$, McNemar's test shows that PCA is not statistically significantly different from no dimensionality reduction (p-value = 0.2249); J-function is statistically significantly better than no dimensionality reduction (p-value = 0.0038); J-function is not statistically significantly better than PCA (p-value = 0.0745).

5. Conclusion

The main obstacles to combining multiple dissimilarity matrices via the Cartesian product of their embeddings are the curse of dimensionality and the parallel cigars phenomenon. We have proposed a new supervised dimensionality reduction approach and show by theorem, simulation and real data experiments that the J-function approach can improve classification performance compared to the alternatives of principal components analysis and no dimensionality reduction at all. The proposed approach is not specific to this type of data and can serve as a general dimensionality reduction

technique. It is particularly useful when (1) the data is high-dimensional and (2) many dimensions of the data have similar variances and PCA is liable to fail in extracting discriminative dimensions.

The proposed dimensionality reduction approach has been developed for the simple two-class problem. One way to extend it to $C > 2$ classes is the following: (1) project data onto the principal axes of the pooled sample covariance matrix; (2) calculate the absolute differences between each class mean and the overall mean; (3) normalize and weight them by corresponding eigenvalues and class proportions, respectively, to obtain a $C \times d$ matrix $\boldsymbol{J}$; (4) finally use the column sums of $\boldsymbol{J}$ to rank and choose principal components. Alternatively, the two-step LDA $\circ$ J approach for $C > 2$ classes can be addressed in two other ways: (1) perform LDA $\circ$ J on each pair of classes and combine the $\binom{C}{2}$ classifiers in the end (Friedman 1996; Hastie and Tibshirani 1998); or (2) perform LDA $\circ$ J on each pair of "class i versus not class i" and combine the K classifiers in the end.

References

ANDERSON, M.J., and ROBINSON, J. (2003), "Generalized Discriminant Analysis Based on Distances.", *Australian & New Zealand Journal of Statistics, 45(3)*, 301–318.

BELHUMEUR, P., HESPANHA, J., and KRIEGMAN, D. (1997), "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7)*, 711–720.

BELLMAN, R.E. (1961), *Adaptive Control Processes - A Guided Tour*, Princeton, NJ: Princeton University Press.

BISHOP, C.M. (1995), *Neural Networks for Pattern Recognition*, Oxford: Clarendon Press; New York: Oxford University Press.

BORG, I., and GROENEN, P.J.F. (2005), *Modern Multidimensional Scaling: Theory and Applications*(2nd ed.), New York: Springer.

BREIMAN, L. (2001), "Random Forests", *Machine Learning, 45(1)*, 5–32.

CHANG, W.C. (1983), "On Using Principal Components Before Separating a Mixture of Two Multivariate Normal Distributions", *Applied Statistics, 32(3)*, 267–275.

CLARKE, K.R. (1993), "Non-parametric Multivariate Analyses of Changes in Community Structure", *Australian Journal of Ecology, 18(1)*, 117–143.

DEVROYE, L., GYÖRFI, L., and LUGOSI, G. (1996). *A Probabilistic Theory of Pattern Recognition*, New York: Springer.

DILLON, W.R., MULANI, N., and FREDERICK, D.G. (1989), "On the Use of Component Scores in the Presence of Group Structure", *The Journal of Consumer Research, 16(1)*, 106–112.

FRIEDMAN, J.H. (1996), "Another Approach to Polychotomous Classification", Technical report, Stanford University, Department of Statistics.

GONZALEZ, R.C., and WOODS, R.E. (2007), *Digital Image Processing* (3rd ed.), Upper Saddle River NJ: Prentice Hall.

GOWER, J.C. (1966), "Some Distance Properties of Latent Root and Vector Methods Used in Multivariate Analysis", *Biometrika, 53(3/4)*, 325–338.

HASTIE, T., and TIBSHIRANI, R. (1998), "Classification by Pairwise Coupling", *The Annals of Statistics, 26(2)*, 451–471.

JAIN, A.K. (1989), *Fundamentals of digital Image Processing*, Englewood Cliffs NJ: Prentice Hall.

JOLLIFFE, I.T., MORGAN, B.J.T., and YOUNG, P.J. (1996), "A Simulation Study of the Use of Principal Components in Linear Discriminant Analysis", *Journal of Statistical Computation and Simulation, 55(4)*, 353–366.

KSHIRSAGAR, A.M., KOCHERLAKOTA, S., and KOCHERLAKOTA, K. (1990), "Classification Procedures Using Principal Component Analysis and Stepwise Discriminant Function", *Communications in Statistics – Theory and Methods, 19(1)*, 91–109.

LIN, D., and PANTEL, P. (2002), "Concept Discovery from Text", in *Proceedings of the 19th International Conference on Computational Linguistics*, Morristown, NJ: Association for Computational Linguistics, pp. 1–7.

MAA, J.F., PEARL, D.K., and BARTOSZYŃSKI, R. (1996), "Reducing Multidimensional Two-sample Data to One-dimensional Interpoint Comparison", *The Annals of Statistics, 24(3)*, 1069–1074.

MILLER, M.I., PRIEBE, C.E., QIU, A., FISCHL, B., KOLASNY, A., BROWN, T., PARK, Y., RATNANATHER, J.T., BUSA, E., JOVICICH, J., YU, P., DICKERSON, B.C., BUCKNER, R.L., and THE MORPHOMETRY BIRN (2009), "Collaborative Computational Anatomy: An MRI Morphometry Study of the Human Brain via Diffeomorphic Metric Mapping", *Human Brain Mapping, 30(7)*, 2132–2141.

PĘKALSKA, E., and DUIN, R.P.W. (2005), *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*, River Edge NJ: World Scientific Publishing Company.

PRIEBE, C.E. (2001), "Olfactory Classification via Interpoint Distance Analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(4)*, 404–413.

SCHÄFER, J., and STRIMMER, K. (2005), "A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics", *Statistical Applications in Genetics and Molecular Biology, 4(1)*.

TAKEMURA, A. (1985), "A Principal Decomposition of Hotelling's T^2 Statistic", in *Multivariate Analysis VI,* ed. P. Krishnaiah), Amsterdam: Elsevier, pp. 583–597.

TORGERSON, W. (1952), "Multidimensional Scaling: I. Theory and Method", *Psychometrika, 17(4)*, 401–419.

TOUSSAINT, G.T. (1971), Note on Optimal Selection of Independent Binary-valued Features for Pattern Recognition (Correspondence), *IEEE Transactions on Information Theory, 17(5)*, 618–618.

TRUNK, G.V. (1979), "A Problem of Dimensionality: A Simple Example", *IEEE Transactions on Pattern Analysis and Machine Intelligence, 1(3)*, 306–307.

ZHU, M., and GHODSI, A. (2006), "Automatic Dimensionality Selection from the Scree Plot via the Use of Profile Likelihood", *Computational Statistics & Data Analysis, 51(2)*, 918–930.