

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

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.

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 (Maa, Pearl, and Bartoszyński, 1996). The dissimilarity representation has attracted substantial interest in various areas (Priebe, 2001; Anderson and Robinson, 2003; Pękalska and Duin, 2005; Trosset and Priebe, 2008; Trosset, Priebe, Park, and Miller, 2008; Miller, Priebe, Qiu, Fischl, Kolasny, Brown, Park, Ratnanather, Busa, Jovicich, Yu, Dickerson, Buckner, and the Morphometry BIRN, 2008).

Since 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 $A \subseteq \{1, \dots, d\}$. The problem of dimensionality reduction is to find an index set A of size $p \equiv |A| < d$ 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 k th principal component (PC) and use θ_k to decide which PCs should be used in discrimination. Jolliffe, Morgan, and Young (1996) pointed out, 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 k th 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, and then calculating t -statistic t_k for the k th PC. Takemura suggested to use the first p PCs (“... to look at $t_1^2, t_1^2 + t_2^2, \dots$ ”) 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). We propose a new supervised dimensionality reduction method, following Takemura’s framework of decomposing Hotelling’s T^2 , to rank $J_k \equiv |t_k|$ to obtain $J_{(1)} \geq J_{(2)} \geq \dots \geq J_{(d)}$ and choose the p PCs corresponding to the first p largest J_k ’s. 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 our approach works better than PCA and no dimensionality reduction, with regards to classification.

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 supervised dimensionality reduction method. Simulation and real data examples are presented in Section 4. Section 5 provides conclusions for our supervised dimensionality reduction approach and how to extend it to suit a problem with more than two classes.

2. Combining Multiple Dissimilarity Representations

A *dissimilarity measure* is a function $\delta : \Xi \times \Xi \rightarrow \mathbb{R}_+$ with $\delta(x_1, x_2) \geq 0$, $\delta(x_1, x_2) = \delta(x_2, x_1)$ and $\delta(x_1, x_2) = 0$ if and only if $x_1 = x_2$. 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 non-standard space. In cases where we observe only the dissimilarities, it will still be useful to think that they are computed from a set of Ξ -valued vectors—the “measurements” of objects. The *dissimilarity representation* of a set of objects is expressed as a nonnegative, symmetric and hollow matrix Δ , which results from computing δ on each pair of objects.

Consider K dissimilarity measures— $\delta_1, \dots, \delta_K$. Let $(X_i, Y_i), i = 1, \dots, n$, be independent and identically distributed, where the X_i are Ξ -valued and the class labels Y_i are (say) $\{0, 1\}$ -valued. And let $\Delta_1, \dots, \Delta_K$ 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.

In principle there are three ways to combine dissimilarities, see Figure 1: (1) “classifier ensemble” combines the classifiers which are separately trained on each dissimilarity matrix; (2) “dissimilarity combination” combines all available dissimilarity matrices into a new one to train a classifier; (3) “embedding product” embeds each dissimilarity matrix first, then combines the embeddings to build a classifier. The process of embedding a dissimilarity matrix, generally known as multidimensional scaling, finds a configuration of n points in some real normed space such that the interpoint distances $\|x_i - x_j\|$ approximate the dissimilarities Δ_{ij} , and the resulting n points are referred to as *embedding*, which is denoted by \mathbf{X} . (In this paper, we use the bold \mathbf{X} to denote

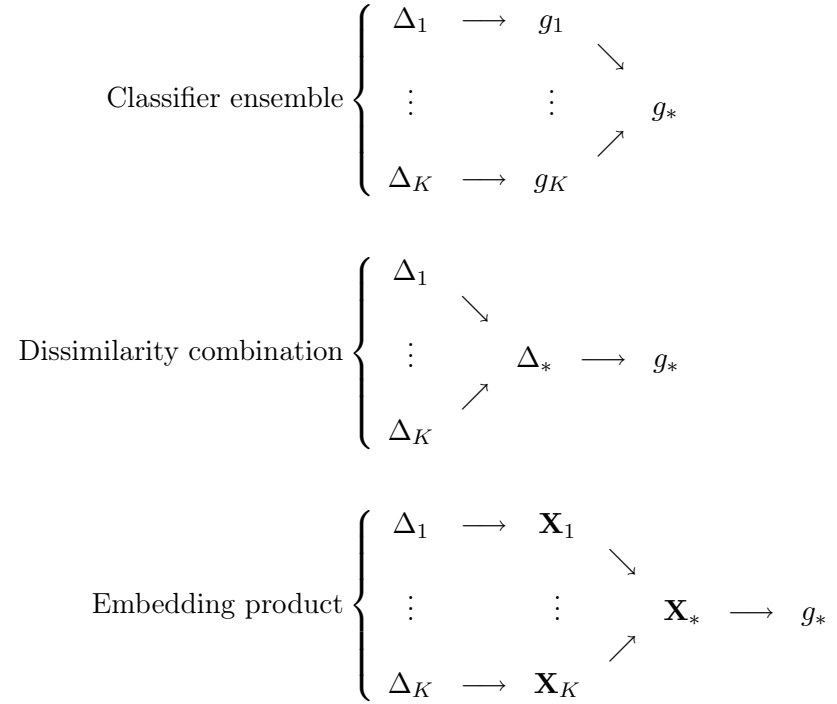


Figure 1: $\Delta_1, \dots, \Delta_K$ are K dissimilarity matrices. “Classifier ensemble” trains classifier g_k on Δ_k and then combines those K classifiers to obtain g_* ; “dissimilarity combination” combines all K dissimilarity matrices to obtain Δ_* , on which a single classifier g_* is trained; “embedding product” embeds each Δ_k into \mathbf{X}_k and combines those embeddings to obtain \mathbf{X}_* , and then a classifier is trained.

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. (2008) 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 only the first few PCs of each embedding in the Cartesian product. 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 $[\mathbf{X}_1, \dots, \mathbf{X}_K]$. The dimensionality of the product space could be very high, especially when K is large. Therefore, dimensionality reduction is necessary to alleviate “the curse of dimensionality”, which is a phenomenon that the number of data points needed to learn in a space increases exponentially with dimension (Bishop, 1995).

3. Dimensionality Reduction

PCA is a widely used technique for creating low dimensional representation of high dimensional data. It is an orthogonal linear transformation that transforms the data to a new (lower dimensional) coordinate system that retains most of the variation in the data. When the data are from several different classes, PCA considers the entire dataset, irrespective of class-membership, to generate PCs. These PCs are usually slightly different from the class-dependent PCs. In addition, the “parallel cigars” phenomenon, where the most discriminative dimensions are not necessarily those with largest variances, may still exist. LDA has the ability to overcome the latter problem by utilizing the label information of the data. However high dimensionality may cause LDA’s performance to decrease (e.g. Trunk, 1979). Belhumeur, Hespanha, and Kriegman (1997) proposed the two-step LDA \circ PCA approach: PCA is used as a preprocessing step for dimensionality reduction before training a linear classifier. The potential problems of PCA mentioned earlier still remain. To overcome these problems, we develop a method for dimensionality reduction which we refer to as the J -function procedure. In particular, we achieve improved performance by composing the J -function procedure with LDA, see Figure 2.

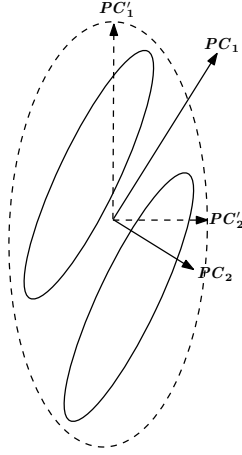


Figure 2: The solid ellipses represent the data from the two classes. The dashed ellipse represents the entire dataset, on which performing PCA reports PC'_1 and PC'_2 as the 1st and 2nd principal components, respectively. The J -function approach first finds the principal components PC_1 and PC_2 by performing eigenvalue decomposition on the pooled covariance matrix. It then computes a J value for each PC and reorders the PCs by the J values associated with them. PCs with larger J values 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 , which utilizes the J -function approach to perform dimensionality reduction before applying LDA, analogous to LDA \circ PCA (Belhumeur et al., 1997).

3.1 J -function

Consider $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times d}$, the data matrix, and $\mathbf{y} = (y_1, \dots, y_n)^T$, the class vector with $\{0, 1\}$ elements. 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 $\mathbf{S} = U\Lambda U^T$ and transform $\tilde{\mathbf{X}}$ to $\mathbf{X} = \tilde{\mathbf{X}}U$.
3. Compute the J value for the i th dimension of \mathbf{X}

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

where \mathbf{m}_0 and \mathbf{m}_1 are the sample means of class 0 and 1, respectively.

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 $X : \Omega \rightarrow \mathbb{R}^d$, Y is Bernoulli distributed with parameter π , and that the conditional distribution of $X|Y = j$ is $N(\boldsymbol{\mu}_j, \Sigma)$. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}^p$ 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

$$L_{f^*(X)}^* \leq L_{f(X)}^*. \quad (1)$$

Proof. Without loss of generality, we assume (i) Σ has full rank d . Since if it does not, the data can be projected into a lower dimensional space, where the covariance matrix is non-singular, without loss of information; (ii) $\Sigma = I_d$. Since there exists a matrix A such that $(AX|Y = j) \triangleq (X_A|Y = j) \sim N(A\boldsymbol{\mu}_j, I_d)$, and any linear projection function of X can be written as $f(X) = f(A^{-1}X_A) \triangleq f_A(X_A)$; (iii) by the same argument used in (ii), we can assume the dimensions of X are ordered according to the values (largest to smallest) of the elements of $|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0|$.

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

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

□

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

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

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 $\mathbf{X} = [x_1, \dots, x_{2n}]^T$, from F_{XY} and then perform dimensionality reduction. LDA is trained on the first n samples and tested on the remaining n samples. 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 training observation class labels, and in the second one we use both the training observation class labels and the testing observation class labels. The following LDA step remains the same for both experiments. Obviously the second experiment leads to an overly optimistic classification error, since the dimensionality reduction step uses the testing observation class labels—but not the LDA step. The error estimate from the second experiment 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— $n = 100, 200, 400$. 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}_0 denote the mean of the estimated classification error when using LDA only, that is no dimensionality reduction. This simulation experiment shows that: (1) $\bar{L}_{\underline{J}}(p) < \bar{L}_J(p) < \bar{L}_0 \leq \bar{L}_P(p)$, 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. 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 select 1,600 pairs by using the query word “tiger” on captions. The “tiger” data are manually labeled into 6 classes based only on captions (see Figure 4). For simplicity we consider the two class problem—“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 unify both image and caption data into one space—the dissimilarity space, where the combination is achieved. We use the first and second order pixel derivatives

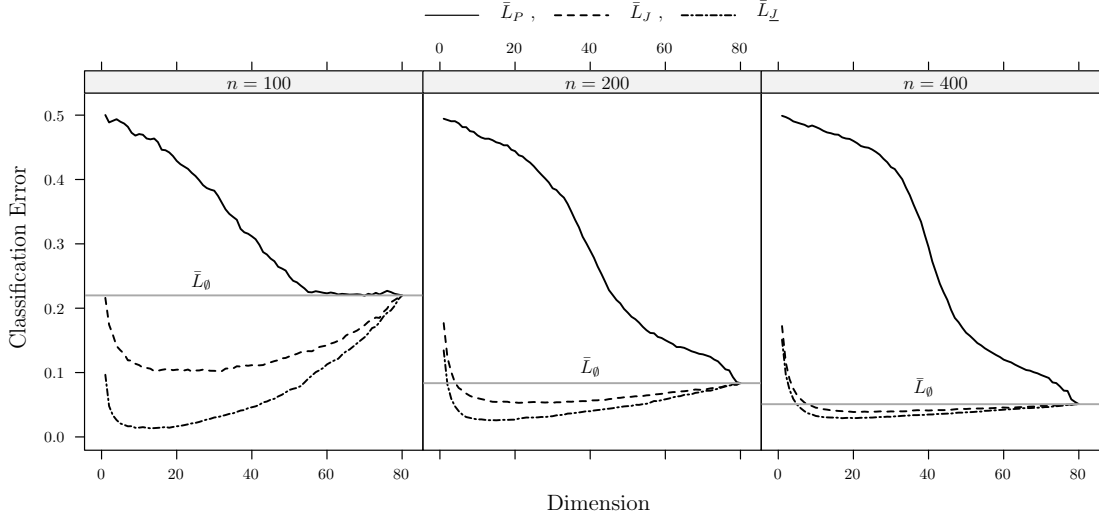


Figure 3: Let $\bar{L}_P(p)$, $\bar{L}_J(p)$ and $\bar{L}_{\underline{J}}(p)$ denote the mean of the estimated classification errors resulting from the p -dimensional data, which are obtained through PCA, the J - and \underline{J} -function procedure, respectively. Let \bar{L}_0 denote the mean of the estimated classification error when using LDA only, that is no dimensionality reduction. These plots depict that (1) $\bar{L}_{\underline{J}}(p) < \bar{L}_J(p) < \bar{L}_0 \leq \bar{L}_P(p)$, 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.

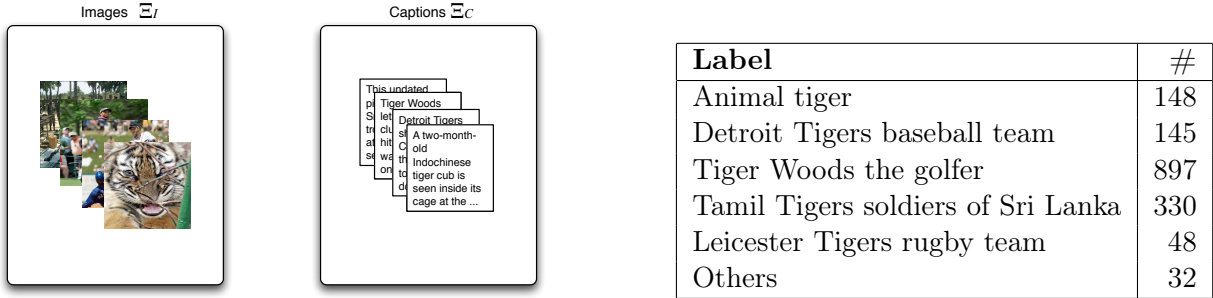


Figure 4: The “tiger” data. Each observation consists of an image/caption pair.

(Gonzalez and Woods, 2007; Jain, 1989) on the images, and the mutual information (Lin and Pantel, 2002) on the captions to 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; Cox and Cox, 2001; 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 $\tilde{\mathbf{X}}_I \in \mathbb{R}^{n \times d(n)}$. (In theory, $d(n)$, the largest embedding dimension of an $n \times n$ dissimilarity matrix by CMDS, is $n - 1 = 1226$. In this example, we choose $d(n) = 1000$ for some numerical reason.) By property of CMDS, the embeddings are given in form of principal components. Hence, the most natural way to perform dimensionality reduction is by examining the scree plot of standard deviations (Figure 5) and ignoring higher rank PCs—a.k.a. PCA. The automatic dimensionality selection technique introduced by Zhu and Ghodsi (2006) is used to determine the reduced dimensionalities $d_C = 473$ (caption) and $d_I = 152$ (image). Notice that the “elbows” are not very good, especially for caption. We want to err on the side of anti-parsimony at this point, since additional dimensionality reduction in the Cartesian product space will occur later. For comparison, we consider also the J -function on $\tilde{\mathbf{X}}_C$ and $\tilde{\mathbf{X}}_I$ (Zhu and Ghodsi’s approach is used

also to determine reduced dimensionality). For the Cartesian product, we separately perform PCA, the J -function and \underline{J} -function to reduce the dimensionality. A linear classifier is built on caption alone, image alone and their combination, respectively. Leave-one-out cross-validation is used to estimate classification errors. Figure 6 shows the above procedures.

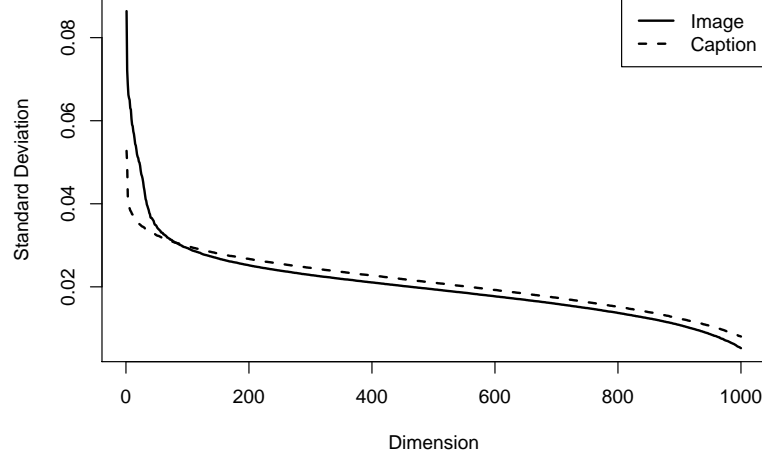


Figure 5: “Tiger” data. Using the classical multidimensional scaling to embed both Δ_C and Δ_I into 1000-dimensional Euclidean space. The scree plot depicts the standard deviation for each dimension.

$$\begin{array}{lcl}
 \text{Image space } \Xi_I \xrightarrow{\delta_I} \Delta_I \xrightarrow{\text{CMDS}} \tilde{\mathbf{X}}_I \in \mathbb{R}^{n \times d(n)} \xrightarrow{\tilde{R}} \mathbf{X}_I \in \mathbb{R}^{n \times d_I} & & \xrightarrow{\text{LDA}} \hat{Y} \\
 & \searrow & \\
 & \otimes & [\mathbf{X}_I \ \mathbf{X}_C] \in \mathbb{R}^{n \times d} \xrightarrow{R} \mathbf{X} \in \mathbb{R}^{n \times p} \xrightarrow{\text{LDA}} \hat{Y} \\
 & \nearrow & \\
 \text{Caption space } \Xi_C \xrightarrow{\delta_C} \Delta_C \xrightarrow{\text{CMDS}} \tilde{\mathbf{X}}_C \in \mathbb{R}^{n \times d(n)} \xrightarrow{\tilde{R}} \mathbf{X}_C \in \mathbb{R}^{n \times d_C} & & \xrightarrow{\text{LDA}} \hat{Y}
 \end{array}$$

Figure 6: “Tiger” data. We combine image and caption data using the dissimilarity representation—image and caption data are transformed into dissimilarity matrices Δ_I and Δ_C , which are then embedded into $d(n)$ -dimensional Euclidean space ($d(n) = 1000$). Dissimilarity reduction procedures \tilde{R} and R are performed on each embedding and then on the Cartesian product, respectively. Finally, a linear classifier is trained. We consider $\tilde{R} \in \{\text{PCA}, J\text{-function}\}$ and $R \in \{\text{PCA}, J, \underline{J}, \emptyset\}$, where \emptyset means no dimensionality reduction.

In Table 1 we list some of classification errors, along with dimensionality reduction procedures applied and reduced dimensionalities. The results suggest that (i) the two step procedure $\text{LDA} \circ J$ works better than LDA only (no dimensionality reduction) and than $\text{LDA} \circ \text{PCA}$; (ii) $\text{LDA} \circ J$ is better than $\text{LDA} \circ \text{PCA}'$, which is the same as $\text{LDA} \circ \text{PCA}$, except using the reduced dimensionalities determined by the J -function; (iii) $\text{LDA} \circ \underline{J}$ is much better than the other procedures—this error estimate is only meaningful as a lower bound of that of $\text{LDA} \circ J$. We use McNemar’s test to validate these statements and show the results in Table 2.

Data	R	Low-dimensional Data	Dimensionality	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	$\overline{178}$	0.0570
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	$\overline{349}$	0.0570
Combination: $\mathbf{X}^P = [\mathbf{X}_I^P \ \mathbf{X}_C^P]$	\emptyset		625	0.1557
	PCA		160	0.1125
	PCA'		$\overline{84}$	0.1174
	J -function		$\overline{84}$	0.0742
	\underline{J} -function		71	0.0171
Combination: $\mathbf{X}^J = [\mathbf{X}_I^J \ \mathbf{X}_C^J]$	J -function		$\overline{112}$	0.0562

Table 1: “Tiger” data. We use the two-step approach $\text{LDA} \circ R$ —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.

5. Conclusion

When combining multiple dissimilarity matrices via the Cartesian product of their embeddings, the curse of dimensionality of the product space and the parallel cigars phenomenon are the main obstacles. In this paper, we propose 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 for this type of data and can serve as a general dimensionality reduction technique. We think it is particular 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 suit $C > 2$ classes can be described as: (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 \mathbf{J} ; (4) finally use the column sums of \mathbf{J} to rank and choose principal components. Alternatively, the two-step $\text{LDA} \circ J$ approach for $C > 2$ classes can be addressed in two other ways: (1) perform $\text{LDA} \circ J$ on each pair of classes and combine the $\binom{C}{2}$ classifiers in the end (Friedman, 1996; Hastie and Tibshirani, 1998); or (2) perform $\text{LDA} \circ J$ on each pair of “class i versus not class i ” and combine the K classifiers in the end.

Acknowledgments

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.

H_A	p -value
$L(J(\tilde{\mathbf{X}}_I)) < L(\text{PCA}(\tilde{\mathbf{X}}_I))$	0
$L(\text{PCA}(\tilde{\mathbf{X}}_I)) < L(\emptyset(\mathbf{X}^P))$	0.6258
$L(\text{PCA}(\mathbf{X}^P)) < L(\emptyset(\mathbf{X}^P))$	4.643e-05
$L(J(\mathbf{X}^P)) < L(\text{PCA}(\mathbf{X}^P))$	2.013e-04
$L(J(\mathbf{X}^P)) < L(\underline{J}(\mathbf{X}^P))$	6.075e-15

Table 2: “Tiger” data. The alternative hypothesis H_A is listed in the first column and the corresponding null hypothesis H_0 is the two dimensionality reduction procedures lead to same classification error. 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 dimensionality reduction procedure $R \in \{\emptyset, \text{PCA}, J, \underline{J}\}$. The definitions of various forms of \mathbf{X} can be found in Table 1. These p -values, together with Table 1, depict that (i) $\text{LDA} \circ J$ works better than LDA only (no dimensionality reduction) and better than $\text{LDA} \circ \text{PCA}$; (ii) $\text{LDA} \circ J$ is better than $\text{LDA} \circ \text{PCA}'$, which is the same as $\text{LDA} \circ \text{PCA}$, except using the reduced dimensionalities determined by the J -function; (iii) $\text{LDA} \circ \underline{J}$ is better than the other procedures—this error is not valid, though it is a (meaningful) lower bound of that of $\text{LDA} \circ J$.

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.
- 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.
- COX, T.F. and COX, M.A.A. (2001). *Multidimensional scaling*. Boca Raton: Chapman & Hall/CRC.
- 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, Department of Statistics, Stanford University.
- GONZALEZ, R.C. and WOODS, R.E. (2007). *Digital image processing*. 3rd ed. Upper Saddle River, NJ: Prentice Hall.
- 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*, 1–7. Morristown, NJ, USA: Association for Computational Linguistics.
- 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 (2008). Collaborative computational anatomy: An MRI Morphometry Study of the Human Brain via Diffeomorphic Metric Mapping. *Human Brain Mapping*.
- PEKALSKA, E. and DUIN, R.P.W. (2005). *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*. 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.
- TAKEMURA, A. (1985). A principal decomposition of Hotelling’s T^2 statistic. In *Multivariate Analysis VI* (ed. P. Krishnaiah), 583–597. Amsterdam: Elsevier.
- 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 (Corresp.). *IEEE Transactions on Information Theory*, 17(5):618–618.
- TROSSET, M.W. and PRIEBE, C.E. (2008). The out-of-sample problem for classical multidimensional scaling. *Computational Statistics & Data Analysis*, 52(10):4635–4642.
- TROSSET, M.W., PRIEBE, C.E., PARK, Y., and MILLER, M.I. (2008). Semisupervised learning from dissimilarity data. *Computational Statistics & Data Analysis*, 52(10):4643–4657.
- 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.