

nificantly. A simple example is shown in Fig.1, where a 2-dimensional data from three classes are shown in Fig.(1(a)). Each class has 50 data points. The covariance for data from each class is $\Sigma_1, \Sigma_2, \Sigma_3$:

$$\Sigma_1 = \begin{bmatrix} 2.336 & 0.015 \\ 0.015 & 1.097 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 1.704 & -0.539 \\ -0.539 & 1.575 \end{bmatrix},$$

$$\Sigma_3 = \begin{bmatrix} 1.514 & 0.512 \\ 0.512 & 1.531 \end{bmatrix}; \quad \Sigma_{123} = \begin{bmatrix} 1.851 & -0.004 \\ -0.004 & 1.401 \end{bmatrix}.$$

These individual covariances are very different. In standard LDA, we average all the classes and obtain $\mathbf{S}_w = \Sigma_{123}$. In this paper, we propose a formulation of LDA that uses pairwise classes. The three pairwise averaged class-covariance: Σ_{12}, Σ_{13} and Σ_{23} are

$$\Sigma_{12} = \begin{bmatrix} 2.020 & -0.262 \\ -0.262 & 1.336 \end{bmatrix}, \quad \Sigma_{13} = \begin{bmatrix} 1.925 & 0.264 \\ 0.264 & 1.314 \end{bmatrix},$$

$$\Sigma_{23} = \begin{bmatrix} 1.609 & -0.013 \\ -0.013 & 1.553 \end{bmatrix}.$$

We see that the pairwise averaged covariance are *much closer* to the two individual covariances as compared to the global average.

Formulation For simplicity, we define the distance $d_{k,l}(\mathbf{G})$ between two classes k, l as

$$d_{k,l}(\mathbf{G}) = 2D_{KL}^{\mathbf{Y}}(\mathcal{N}_k, \mathcal{N}_l),$$

where $D_{KL}^{\mathbf{Y}}(\mathcal{N}_k, \mathcal{N}_l)$ is defined in Eq.(5), and Σ_{kl} is a pairwise covariance matrix (average of the pair of classes) and defined as

$$\Sigma_{kl} = \beta \frac{n_k \Sigma_k + n_l \Sigma_l}{n_k + n_l} + (1 - \beta) \Sigma. \quad (8)$$

Here we use the globally averaged covariance $\Sigma = \mathbf{S}_w$ as a regularization. Parameter $0 \leq \beta \leq 1$ controls the balance of global covariance matrix Σ and local pairwise covariance matrix Σ_k, Σ_l .

The pairwise-covariance LDA is defined the same as that in Theorem 1:

$$\max_{\mathbf{G}} J_1(\mathbf{G}) = \sum_{k < l} n_k n_l d_{kl}(\mathbf{G}), \quad (9)$$

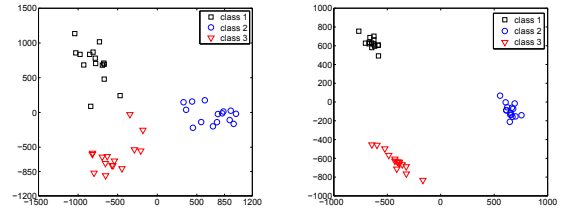
where $\mathbf{G} \in \mathbb{R}^{p \times (K-1)}$ is the projection. The objective in Eq.(9) is similar to standard LDA (except that we use pairwise covariance instead of global averaged covariance).

The proposed new model

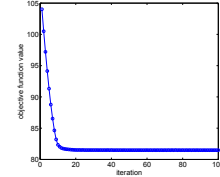
Back to the form of Eq.9, it is easy to see that we can define a better objective. In maximizing J_1 , all pairs of distances are treated equally. However, in classification, we wish the pair of classes with smaller distances to be given more weight, i.e., after projecting to $\mathbf{Y} = \mathbf{G}^T \mathbf{X}$ subspace, they are more separated (as compared to other pairs of classes). On the other hand, if two classes are already well-separated, i.e., their distances are large, they can have less weight in the objective function. Therefore, we propose the following pairwise covariance *properly weighted* objective function:

$$\min_{\mathbf{G}} J_2(\mathbf{G}) = \sum_{k < l} \frac{n_k n_l}{[d_{kl}(\mathbf{G})]^q}, \quad s.t. \quad \mathbf{G}^T \mathbf{G} = \mathbf{I}, \quad (10)$$

where $q \geq 1$ is a hyper-parameter. In this objective function, the pair of classes with smaller distances contribute more



(a) Results of standard LDA. (b) Results of pcLDA.



(c) Convergence of algorithm.

Figure 3: Data: 45 data points (images) from 3 classes on mnist dataset. Original 784-dimensional data are projected into 2-dimension. (a) Results of standard LDA; (b) Results of pcLDA; (c) Convergence of algorithm on mnist. Shown are objective function vs. iterations.

than the pair of classes with larger distances. Parameter q controls how much the pair of classes with smaller distances are weighted. The larger q is, the stronger that pair of classes is weighted. In practice, we found that $q = \{1, 2\}$ are good choices. This model is our final proposed model. For simplicity, we call it **pairwise-covariance LDA (pcLDA)** with the proper weighting implicit.

As defined in Eq.(10), the objective is invariant under any non-singular transformation using $\mathbf{A} \in \mathbb{R}^{(K-1) \times (K-1)}$, i.e., $J_2(\mathbf{GA}) = J_2(\mathbf{G})$. To fix this uncertainty, we require $\mathbf{G}^T \mathbf{G} = \mathbf{I}$.

Illustrations of pcLDA

We illustrate pcLDA on synthetic and real data. In Fig.1, LDA and pcLDA results on a synthetic 2D dataset of 150 data points (50 data of each class) are shown. We show the data distribution and 1-dimensional projection results using LDA and pcLDA. The point here is that the globally averaged covariance \mathbf{S}_w is a poor representation of the individual covariances, but the pairwise-covariance approach seems to give a better representation such that a single pcLDA dimension can clearly separate the 3 classes, while standard LDA needs 2-dimensions to separate data from different classes (results not shown).

In Fig.2, we show the results on the widely used iris data¹. Iris has 150 data points with $K=3$ classes. Thus LDA project to $K-1=2$ dimensions. Fig.2 indicates that pcLDA gives clear discrimination between classes 2 and 3 while standard LDA has strong mixing between classes 2 and 3. In Fig.3, we show results on 45 images (from $K=3$ classes) from mnist handwritten digits image dataset. LDA projections to 2-dimension are shown. Result of pcLDA shows that

¹<http://archive.ics.uci.edu/ml/datasets/Iris>

the 3 classes contract strongly and become more separated as compared to the LDA results. These results demonstrate the benefits of the pairwise-covariance properly weighted LDA. More experiments and comparisons with related methods are reported in §7.

Algorithm to solve Pairwise-covariance LDA

The key idea of our approach is to use gradient descent algorithm to solve pcLDA of Eq.(10). The gradient of $J_2(G)$ is

$$\nabla J_2 \triangleq \frac{\partial J_2}{\partial \mathbf{G}} = - \sum_{k < l} \frac{qn_k n_l}{[d_{kl}(\mathbf{G})]^{q+1}} \frac{\partial d_{kl}(\mathbf{G})}{\partial \mathbf{G}}. \quad (11)$$

For notational simplicity, we write

$$\begin{aligned} \mathbf{B}_{kl} &= (\mu_k - \mu_l)(\mu_k - \mu_l)^T, \\ d_{kl}(\mathbf{G}) &= \text{Tr}(\mathbf{G}^T \mathbf{B}_{kl} \mathbf{G}) (\mathbf{G}^T \Sigma_{kl} \mathbf{G})^{-1}. \end{aligned} \quad (12)$$

Using Eq.(12), the derivative of $d_{kl}(\mathbf{G})$ is

$$\begin{aligned} \frac{\partial d_{kl}(\mathbf{G})}{\partial \mathbf{G}} &= 2[\mathbf{B}_{kl} \mathbf{G} (\mathbf{G}^T \Sigma_{kl} \mathbf{G})^{-1} \\ &\quad - \Sigma_{kl} \mathbf{G} (\mathbf{G}^T \Sigma_{kl} \mathbf{G})^{-1} (\mathbf{G}^T \mathbf{B}_{kl} \mathbf{G}) (\mathbf{G}^T \Sigma_{kl} \mathbf{G})^{-1}]. \end{aligned} \quad (13)$$

Note that $(\mathbf{G}^T \Sigma_{kl} \mathbf{G})^{-1}$ is an inverse of a small $(K-1)$ -by- $(K-1)$ matrix. ∇J_2 can be efficiently computed using Algorithm 1.

Algorithm 1 Computation of $\nabla J_2(\mathbf{G})$ (i.e., Eq.11) or $\nabla J_2(\mathbf{A})$ (i.e., gradient of Eq.21).

Input: $\mathbf{G}, \{\Sigma_k, \mu_k\}, q$

Output: ∇J_2

Algorithm:

- 1: $\mathbf{F} = 0$
 - 2: **for** $l = 1$ to K **do**
 - 3: **for** $k = l + 1$ to K **do**
 - 4: Compute $\mu_{kl} = \mu_k - \mu_l$.
 - 5: Compute $\mathbf{b} = \mathbf{G}^T \mu_{kl}$.
 - 6: Compute Σ_{kl} according to Eq.(8). % Σ_{kl}^ϕ according to Eq.(23)
 - 7: Compute $\mathbf{B} = \Sigma_{kl} \mathbf{G}$.
 - 8: Compute $\mathbf{b} = (\mathbf{G}^T \mathbf{B})^{-1} \mathbf{b}$.
 - 9: Compute $\mathbf{a} = n_k n_l (\mu_{kl} - \mathbf{B} \mathbf{b}) / (\mu_{kl}^T \mathbf{G} \mathbf{b})^{q+1}$.
 - 10: Compute $\mathbf{F} = \mathbf{F} + \mathbf{a} \times \mathbf{b}^T$ % cross-product between vectors \mathbf{a}, \mathbf{b}
 - 11: **end for**
 - 12: **end for**
 - 13: $\nabla J_2 = -2q\mathbf{F}$.
 - 14: **Output:** ∇J_2 .
-

The constraint $\mathbf{G}^T \mathbf{G} = \mathbf{I}$ enforces \mathbf{G} on the Stiefel manifold. Variations of \mathbf{G} on this manifold is parallel transport, which gives some restriction to the gradient. This has been worked out in (Edelman, Arias, and Smith 1998). The gradient that preserves the manifold structure is

$$\nabla J_2 - \mathbf{G}[\nabla J_2]^T \mathbf{G}. \quad (14)$$

Thus the algorithm computes the new \mathbf{G} as follows,

$$\mathbf{G} \leftarrow \mathbf{G} - \eta(\nabla J_2 - \mathbf{G}[\nabla J_2]^T \mathbf{G}) \quad (15)$$

The step size η is usually chosen as,

$$\eta = \rho \|\mathbf{G}\|_1 / \|\nabla J_2 - \mathbf{G}[\nabla J_2]^T \mathbf{G}\|_1, \quad \rho = 0.001 \sim 0.01. \quad (16)$$

where $\|\mathbf{A}\|_1 = \sum_{ij} |\mathbf{A}_{ij}|$. Occasionally, due to the loss of numerical accuracy, we do the projection: $\mathbf{G} \leftarrow \mathbf{G}(\mathbf{G}^T \mathbf{G})^{-\frac{1}{2}}$ to restore $\mathbf{G}^T \mathbf{G} = \mathbf{I}$. Starting with the standard LDA solution of \mathbf{G} , this algorithm is iterated until the algorithm converges to a local optimal solution. Fig. 3(c) shows the convergence of algorithm on dataset `mnist`.

Pairwise-covariance Kernel LDA

Kernel LDA (Mika et al. 1999; Tao et al. 2004) is non-linear generalization of LDA. We can derive the kernel version of pcLDA. Let $\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i)$ or $\mathbf{X} \rightarrow \phi(\mathbf{X}) = (\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_n))$. For 2-class LDA, the projection vector is $\mathbf{g} = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i) = \phi(\mathbf{X}) \boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = (\alpha_1 \dots \alpha_n)^T$. For K -class LDA, the projection vector $\mathbf{g}_k = \sum_{i=1}^n \alpha_{ik} \phi(\mathbf{x}_i) = \phi(\mathbf{X}) \boldsymbol{\alpha}_k$, thus, $\mathbf{G} = (\mathbf{g}_1 \dots \mathbf{g}_{K-1}) = \phi(\mathbf{X}) \mathbf{A}$, where $\mathbf{A} = (\boldsymbol{\alpha}_1 \dots \boldsymbol{\alpha}_{K-1})$.

Under the transformation $\mathbf{X} \rightarrow \phi(\mathbf{X})$, $\mathbf{G} \rightarrow \phi(\mathbf{X}) \mathbf{A}$, it is easy to see that the LDA objective of Eq.(1) transforms into

$$\text{Tr}(\mathbf{G}^T \mathbf{S}_b^\phi \mathbf{G}) (\mathbf{G}^T \mathbf{S}_w^\phi \mathbf{G})^{-1} \rightarrow \text{Tr}(\mathbf{A}^T \mathbf{S}_b^\phi \mathbf{A}) (\mathbf{A}^T \mathbf{S}_w^\phi \mathbf{A})^{-1} \quad (17)$$

where the kernel within-class scatter matrix is:

$$\begin{aligned} (\Sigma_k^\phi)_{ij} &= \phi(\mathbf{x}_i)^T \left[\frac{1}{n_k} \sum_{s \in C_k} \phi(\mathbf{x}_s) \phi(\mathbf{x}_s)^T \right] \phi(\mathbf{x}_j) \\ &= \frac{1}{n_k} \sum_{s \in C_k} \mathcal{K}_{is} \mathcal{K}_{sj}, \quad \mathbf{S}_w^\phi = \frac{1}{n} \sum_{k=1}^K n_k (\Sigma_k^\phi) = \frac{1}{n} \mathcal{K}^2, \end{aligned} \quad (18)$$

and the kernel between-class scatter matrix is:

$$\begin{aligned} (\mathbf{S}_b^\phi)_{ij} &= \phi(\mathbf{x}_i)^T \left[\frac{1}{n} \sum_{k=1}^K n_k (\bar{\phi}_k - \bar{\phi})(\bar{\phi}_k - \bar{\phi})^T \right] \phi(\mathbf{x}_j) \\ &= \frac{1}{n} \sum_{k=1}^K n_k (\mathcal{K}_{i\bar{k}} - \mathcal{K}_{i\cdot})(\mathcal{K}_{\bar{k}j} - \mathcal{K}_{\cdot j}), \end{aligned} \quad (19)$$

where we use the shorthand notations:

$$\begin{aligned} \bar{\phi} &= \frac{1}{n} \sum_{s=1}^n \phi(\mathbf{x}_s), \quad \bar{\phi}_k = \frac{1}{n_k} \sum_{s \in C_k} \phi(\mathbf{x}_s), \\ \mathcal{K}_{i\cdot} &= \mathcal{K}_{\cdot i} = \frac{1}{n} \sum_{s=1}^n \mathcal{K}_{is}, \quad \mathcal{K}_{\bar{k}i} = \mathcal{K}_{i\bar{k}} = \frac{1}{n_k} \sum_{s \in C_k} \mathcal{K}_{is}. \end{aligned} \quad (20)$$

The solution of kernel LDA is given by the largest k eigenvectors of the eigen-equation $\mathbf{S}_b^\phi v = \lambda \mathbf{S}_w^\phi v$. When $K = 2$, this reduces to the familiar 2-class kernel LDA (Tao et al. 2004). Efficient computation of \mathbf{S}_b^ϕ is given in the end of §5.1.

We are now ready to present the pairwise-covariance kernel LDA. We apply the same transformation to the pairwise-covariance LDA. We have

Theorem 2. *Under the transformation $\mathbf{X} \rightarrow \phi(\mathbf{X})$, $\mathbf{G} \rightarrow \phi(\mathbf{X}) \mathbf{A}$, the pairwise-covariance LDA of $J_2(\mathbf{G})$ becomes $J_2(\mathbf{A})$:*

$$\min_{\mathbf{A}} \sum_{k < l} \frac{n_k n_l}{[\text{Tr}(\mathbf{A}^T \mathbf{B}_{kl}^\phi \mathbf{A}) (\mathbf{A}^T \Sigma_{kl}^\phi \mathbf{A})^{-1}]^q}, \quad \text{s.t. } \mathbf{A}^T \mathbf{A} = \mathbf{I}. \quad (21)$$

where

$$\begin{aligned} (\mathbf{B}_{kl}^\phi)_{ij} &= \phi(\mathbf{x}_i)^T (\bar{\phi}_k - \bar{\phi}_l) (\bar{\phi}_k - \bar{\phi}_l)^T \phi(\mathbf{x}_j) \\ &= (\mathcal{K}_{i\bar{k}} - \mathcal{K}_{i\bar{l}}) (\mathcal{K}_{\bar{k}j} - \mathcal{K}_{\bar{l}j}) \end{aligned} \quad (22)$$

where shorthand notations are defined in Eq.(20), Σ_k^ϕ is defined in Eq.(18), and

$$\Sigma_{kl}^\phi = \beta \frac{n_k \Sigma_k^\phi + n_l \Sigma_l^\phi}{n_k + n_l} + (1 - \beta) \Sigma^\phi. \quad (23)$$

Algorithm for Kernel PC-LDA

We solve $J_2(\mathbf{A})$ of Eq.(21) using the same algorithm in computing pcLDA using $J_2(\mathbf{G})$ of Eq.(10). The derivative is the same as Eqs.(19,20) except \mathbf{B}_{kl} is replaced by \mathbf{B}_{kl}^ϕ , Σ_{kl} replaced by Σ_{kl}^ϕ , \mathbf{G} by \mathbf{A} . The constraint $\mathbf{A}^T \mathbf{A} = \mathbf{I}$ is handled in same way as $\mathbf{G}^T \mathbf{G} = \mathbf{I}$ in Eqs.(20,21). The step size is given in Eq.(22). The remaining part is the efficient computation of the gradient $\nabla J_2(\mathbf{A})$. First, we note that $\{\mathbf{B}_{kl}^\phi\}, \{\Sigma_k^\phi\}$ of Eqs.(22,23) can be efficiently computed. Let \mathbf{V}_k be a n -by- n_k matrix consisting of n_k columns of \mathcal{K} belonging to class k . It is ready to see that in Eq.(21),

$$\Sigma_k^\phi = \frac{1}{n_k} \mathbf{V}_k \mathbf{V}_k^T, \quad \mathbf{u}_k = \frac{1}{n_k} \mathbf{V}_k \mathbf{e}, \quad (24)$$

where $\mathbf{e} = (1 \cdots 1)^T$. Here for clarity, we use \mathbf{u}_k to represent the vector $\mathcal{K}_{i,\bar{k}}, i = 1 \cdots n$. Clearly, $\mathbf{B}_{kl}^\phi = (\mathbf{u}_k - \mathbf{u}_l)(\mathbf{u}_k - \mathbf{u}_l)^T$. Now $\nabla J_2(\mathbf{A})$ is computed using Algorithm 1, with the replacement

$$\mu_k \leftarrow \mathbf{u}_k, \quad \Sigma_k \leftarrow \Sigma_k^\phi. \quad (25)$$

\mathbf{S}_b^ϕ can be efficiently computed as $\mathbf{S}_b^\phi = (1/n) \sum_k n_k (\mathbf{u}_k - \mathbf{v})(\mathbf{u}_k - \mathbf{v})^T$, $\mathbf{v} = (1/n) \phi(\mathbf{X}) \mathbf{e}$.

Related Work

A detailed survey of recent LDA works can be found in (Ye and Ji 2008).

Other LDA formulation There exist earlier works (Li, Jiang, and Zhang 2003), (Yan et al. 2004) which maximize the difference of traces, a.k.a maximum margin criteria (MMC). Several LDA formulations with different constraints and overfit analysis are given in (Luo, Ding, and Huang 2011), (Yan et al. 2004). To solve the well-known singularity or under-sampled problem, there are many extensions of LDA methods proposed, such as Regularized LDA (RLDA) (Hastie, Tibshirani, and Friedman 2001), uncorrelated LDA (ULDA) (Ye 2005b), orthogonal LDA (OLDA) (Ye 2005a) and orthogonal centroid method (OCM) (Park, Jeon, and Z 2003), etc. Among these, ULDA extracts the feature vectors which are mutually uncorrelated in low-dimensional space.

Connection with metric learning David et.al. (Alipanahi, Biggs, and Ghodsi 2008) showed a strong relationship between distance metric learning methods and the Fisher Discriminant Analysis. Our pairwise-covariance LDA formulation of Eq.(10) and kernel pcLDA of Eq.(21) can serve for distance metric learning purpose, which can be used for many applications (e.g., (Kong and Yan 2013), (Kong et al. 2012), etc).

Table 1: Characteristics of datasets

Dataset	# data	#dimension	#Class
MSRCv1	210	432	7
Umist	360	644	20
Mnist	150	784	10
Binalpha	1014	320	36

There are also works discussing local discriminative Gaussian (LDG) dimensionality reduction (Parrish and Gupta 2012), local fisher discriminant analysis (Sugiyama 2006). Sparsity in the LDA solution (Clemmensen et al. 2011), (Zhang and Chu 2013) is also desirable for interpretation purpose, because it is robustness to the noise and will lead to efficient computation in prediction. However, to our knowledge, none of the above works consider the pairwise covariance by computing distance of the projection in a pairwise way, which is the focus of this paper.

Experiment results

Dataset We evaluate the proposed pairwise-covariance LDA using four data sets (see Table 1) for multi-class classification experiments, including one face dataset `umist`, two digit datasets `mnist` (Lecun et al. 1998), `binalpha`, one image scene dataset `MSRCv1` (Lee and Grauman 2009)². Due to space limit, we omit more details of datasets. Table 1 summarizes the datasets.

Methods & Parameter Settings In our experiment, we use 5-round 5-fold cross validation to evaluate the classification performance. Each dataset is evenly partitioned into 5 parts. Only one part is used as testing and the other 4 parts are used for training. We report the average results for 5 rounds. Next, we give an overview of the dimension reduction and classification methods used in our experiment. The compared methods can be divided into several groups.

(1) **LDA and MMC (Li, Jiang, and Zhang 2003; Yan et al. 2004), kernel LDA (KLDA)** For LDA, maximum margin criterion(MMC) ((Li, Jiang, and Zhang 2003; Yan et al. 2004)), kernel-LDA of Eq.(17) method, we project original data into LDA-subspace, and $k(k=3)$ nearest neighbor classifier is used for classification. For kernel LDA, we use RBF kernel to construct the pairwise similarity $\mathbf{W}_{ij} = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}$, where bandwidth γ is searched in the grid $\{10^{-4}, 10^{-3}, \dots, 10^3, 10^4\}$.

(2) **Regularized LDA (RLDA) (Hastie, Tibshirani, and Friedman 2001), uncorrelated LDA (ULDA) (Ye 2005b), orthogonal LDA (OLDA) (Ye 2005a) and orthogonal centroid method (OCM) (Park, Jeon, and Z 2003)**. We compare our method against four methods of generalized LDA. It has been shown (Ye and Ji 2008) that these four LDA-extensions can be described in a unified framework for generalized LDA. However, there still exist subtle differences among them. The parameter μ in regularized LDA is determined by cross validation.

(3) **Proposed pairwise-covariance LDA model of**

²<http://research.microsoft.com/en-us/projects/ObjectClassRecognition/>

Table 2: Multi-class Classification Accuracy on 4 datasets using 9 different dimension reduction methods: LDA, kernel LDA(KLDA), pcLDA, kernel pcLDA (pcKLDA), and 5 other methods: MMC, RLDA, ULDA, OLDA, OCM.

Data	LDA	MMC	RLDA	ULDA	OLDA	OCM	pcLDA ($\beta=1$)	KLDA	pcKLDA($\beta=1$)
MSRC	68.57	67.45	68.54	69.11	67.34	68.91	71.32	68.78	72.39
Binalpha	76.37	72.38	77.66	77.95	72.30	78.89	81.38	79.23	80.12
Mnist	84.37	85.29	84.14	85.01	86.69	84.45	87.10	83.09	86.26
Umist	94.16	93.45	94.44	94.24	91.94	93.61	95.35	91.41	92.07

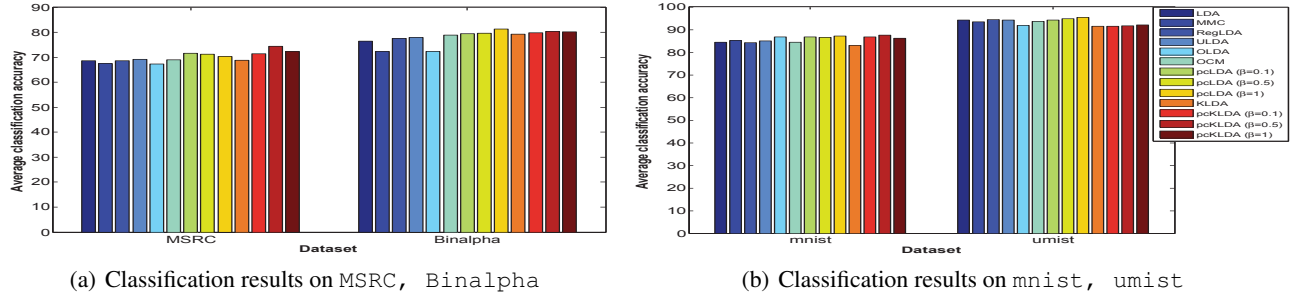


Figure 4: Classification results comparisons on 4 datasets, including our methods: pcLDA, pcKLDA at $\beta = \{0.1, 0.5, 1\}$ and seven other methods: LDA, KLDA, MMC, RLDA, ULDA, OLDA, OCM.

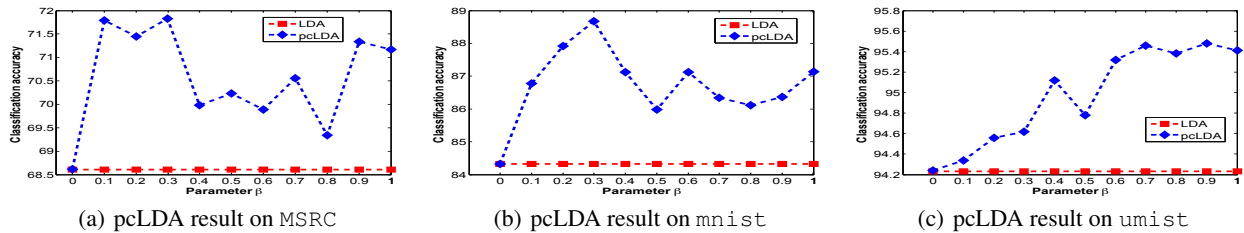


Figure 5: Classification accuracy w.r.t different parameter β for our model of Eq.(10) on dataset MSRC, mnist, umist. Red line gives LDA results, and blue line draws pcLDA results at $\beta = \{0, 0.1, \dots, 0.9, 1.0\}$.

Eq.(10)(pcLDA) and kernel pairwise-covariance LDA model (pcKLDA) of Eq.(21) We set $q = 1$ for Eq.(10), Eq.(21) in our experiments. The parameter β is set to be $\{0.1, 0.5, 1\}$. To make a fair comparison, we project all original data to $(C-1)$ dimension, and $k(k=3)$ nearest neighbor classifier is used for classification purpose.

Classification Performance Analysis Table 2 and Fig.4 present the classification performance using different dimension reduction methods. We make several important observations from experiment results.

(1) As compared to standard LDA, MMC and other dimension reduction methods, pcLDA consistently provides better classification performance at different β values (e.g., $\beta = \{0.1, 0.5, 1\}$). For example, there is nearly 5% performance improvement on binalpha dataset when compared with standard LDA method. Note binalpha dataset is composed of data from $K=36$ classes, this indicates that the proposed pairwise pairwise-covariance LDA method gives much performance improvement at large class numbers.

(2) In kernel space, kernel version of LDA and pcLDA do not improve the classification performance quite a bit (sometimes even worse). However, pcKLDA still outperforms standard KLDA in kernel space.

(3) β controls the complexity of our model, i.e., when β approaches 1, pcLDA uses local pairwise covariance matrix, and when β approaches 0, pcLDA uses global covariance matrix which is equivalent to standard LDA. Fig.(5) shows the classification results on three datasets: MSRC, mnist and umist. The experiment results suggest that, generally, we tend to get better classification results for larger values of β . This further confirms our intuition, the pairwise covariance really helps to capture the data distribution as compared to globally averaged variance, and thus the projection and classification results are improved. Moreover, rather than maximizing the sum of inter-class distances, we minimize the sum of inverse inter-class distances. This choice makes classes that are close together have more influence on the LDA fit than those classes that are well-separated.

Conclusion

We present a pairwise-covariance model for linear discriminant analysis. The proposed model computes the projection by utilizing the pairwise class information. An efficient algorithm is present to solve the proposed model. Proposed method can be easily extended in kernel space. Experiment results indicate the good performance of proposed method.

Acknowledgement. This research is partially supported by NSF-CCF-0917274 and NSF-DMS-0915228 grants.

References

- Alipanahi, B.; Biggs, M.; and Ghodsi, A. 2008. Distance metric learning vs. fisher discriminant analysis. In *AAAI*.
- Clemmensen, L.; Hastie, T.; Wiiten, D.; and Ersboll, B. 2011. Sparse discriminant analysis. *Technometrics*.
- Edelman, A.; Arias, T. A.; and Smith, S. T. 1998. The geometry of algorithms with orthogonality constraints. *SIAM J. MATRIX ANAL. APPL* 20(2):303–353.
- Hastie, T.; Tibshirani, R.; and Friedman, J. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
- Hoi, S. C. H.; Liu, W.; Lyu, M. R.; and Ma, W.-Y. 2006. Learning distance metrics with contextual constraints for image retrieval. In *CVPR*.
- Kong, D., and Ding, C. H. Q. 2012. A semi-definite positive linear discriminant analysis and its applications. In *ICDM*, 942–947.
- Kong, D., and Yan, G. 2013. Discriminant malware distance learning on structural information for automated malware classification. In *KDD*, 1357–1365.
- Kong, D.; Ding, C. H. Q.; Huang, H.; and Zhao, H. 2012. Multi-label relief and f-statistic feature selections for image annotation. In *CVPR*, 2352–2359.
- Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 2278–2324.
- Lee, Y. J., and Grauman, K. 2009. Foreground focus: Unsupervised learning from partially matching images. *International Journal of Computer Vision* 85(2):143–166.
- Li, H.; Jiang, T.; and Zhang, K. 2003. Efficient and robust feature extraction by maximum margin criterion. In *Proceedings of Advances in Neural Information Processing Systems (NIPS 2003)*.
- Luo, D.; Ding, C.; and Huang, H. 2011. Linear discriminant analysis: New formulations and overfit analysis. In *AAAI2011*.
- Mika, S.; Ratsch, G.; Weston, J.; Scholkopf, B.; and Muller, K. 1999. Fisher discriminant analysis with kernels.
- Park, H.; Jeon, L. M.; and Z, J. B. R. 2003. Lower dimensional representation of text data based on centroids and least squares. *BIT* 43:2003.
- Parrish, N., and Gupta, M. 2012. Dimensionality reduction by local discriminative gaussians. In *ICML*.
- Sugiyama, M. 2006. Local fisher discriminant analysis for supervised dimensionality reduction. In *ICML*, 905–912.
- Tao, X.; Ye, J.; Li, Q.; Janardan, R.; and Cherkassky, V. 2004. Efficient kernel discriminant analysis via qr decomposition. In *The Eighteenth Annual Conference on Neural Information Processing Systems (NIPS 2004)*, 1529–1536.
- Wang, H.; Yan, S.; Xu, D.; Tang, X.; and Huang, T. 2007. Trace ratio vs. ratio trace for dimensionality reduction. In *CVPR*.
- Xiang, S.; Nie, F.; and Zhang, C. 2008. Learning a mahalanobis distance metric for data clustering and classification.
- Yan, J.; Zhang, B.; Yan, S.; Yang, Q.; and Li, H. 2004. Immc: incremental maximum margin criterion. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Ye, J., and Ji, S. 2008. *Discriminant Analysis for Dimensionality Reduction: An Overview of Recent Developments, Biometrics: Theory, Methods & Applications*. IEEE/Wiley.
- Ye, J. 2005a. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *The Journal of Machine Learning Research* 6.
- Ye, J. 2005b. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning* 6:483–502.
- Zhang, X., and Chu, D. 2013. Sparse uncorrelated linear discriminant analysis. In *ICML*, 45–52.