San José State University

Math 251: Statistical and Machine Learning Classification

Dimensionality reduction for classification

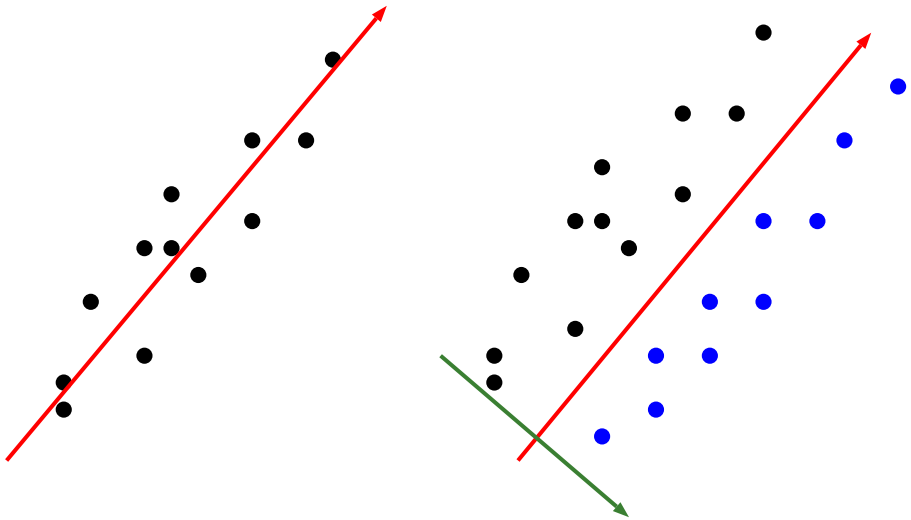
Dr. Guangliang Chen

Outline

- PCA
- LDA
- 2DLDA (new)
- Assignment 2

Dimensionality reduction methods

- Principal Component Analysis (PCA): preserving overall variance
- Linear Discriminant Analysis (LDA): preserving between-class separation

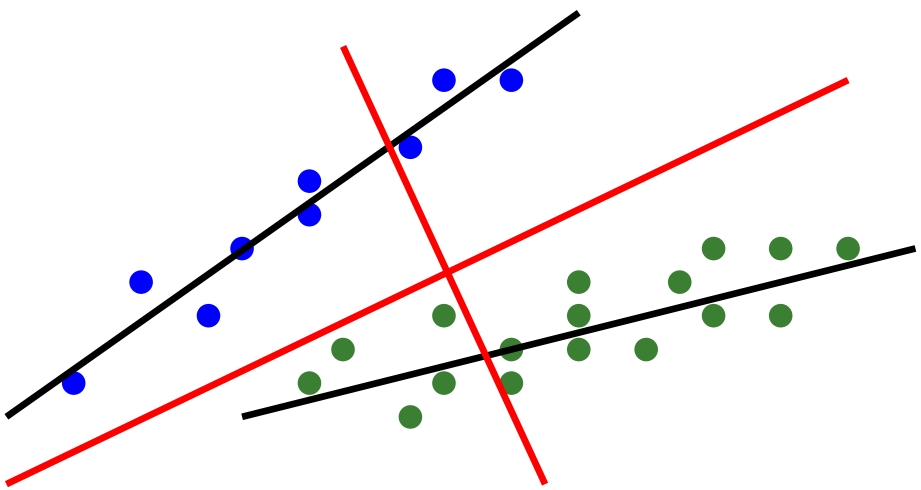


I will also introduce **2DLDA**, a variant of LDA that can directly work on matrix data.

Principle Component Analysis (PCA)

PCA for labeled data

In the supervised setting (when data points have labels), one can perform PCA on the full data set without using the labels to project the different classes onto the same PCA plane.



We call this procedure global PCA.

PCA for classification

Note that in classification, there are two data sets: $\mathbf{X}_{\text{train}}$ and $\mathbf{X}_{\text{test}}$.

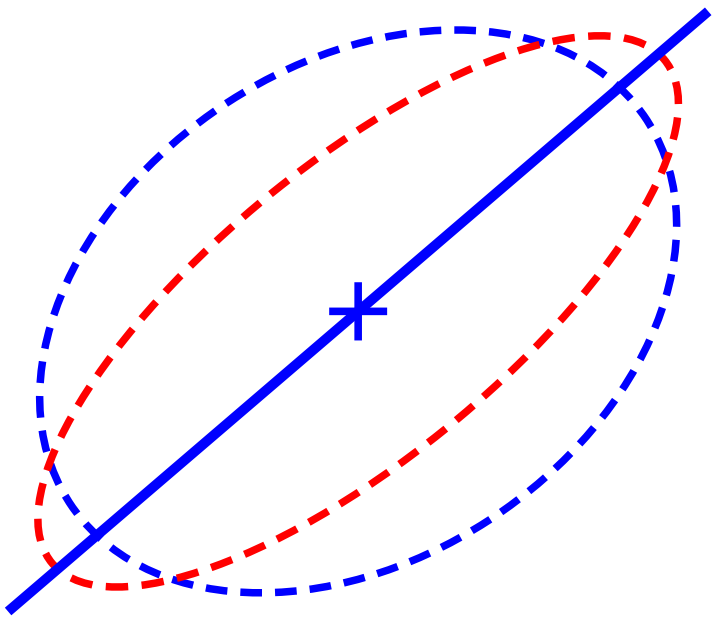
One should perform global PCA on the training set

$$\mathbf{X}_{\text{train}} - \mathbf{1}\mathbf{m}_{\text{train}}^T = \mathbf{U}_{\text{train}}\mathbf{\Sigma}_{\text{train}}\mathbf{V}_{\text{train}}^T$$

and use it to project both sets of data onto the PCA plane:

$$\mathbf{Y}_{\text{train}} = \mathbf{U}_{\text{train}}\mathbf{\Sigma}_{\text{train}}$$

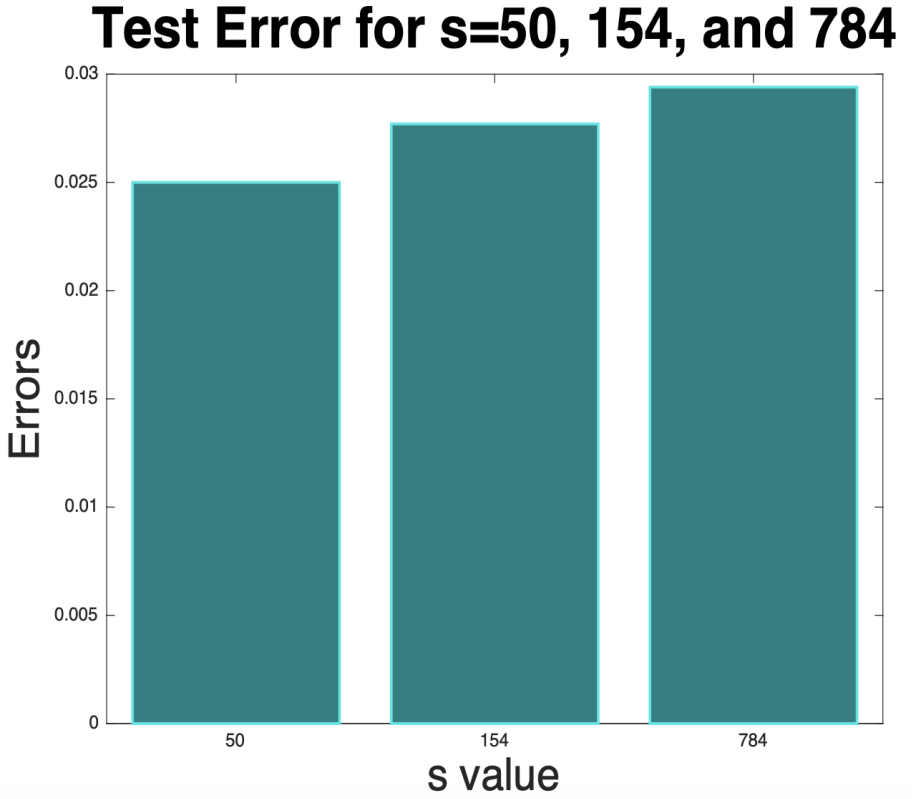
$$\mathbf{Y}_{\text{test}} = \left(\mathbf{X}_{\text{test}} - \mathbf{1}\mathbf{m}_{\text{train}}^T\right) \cdot \mathbf{V}_{\text{train}}$$



Finally, select a classifier to work in the reduced space, e.g.,

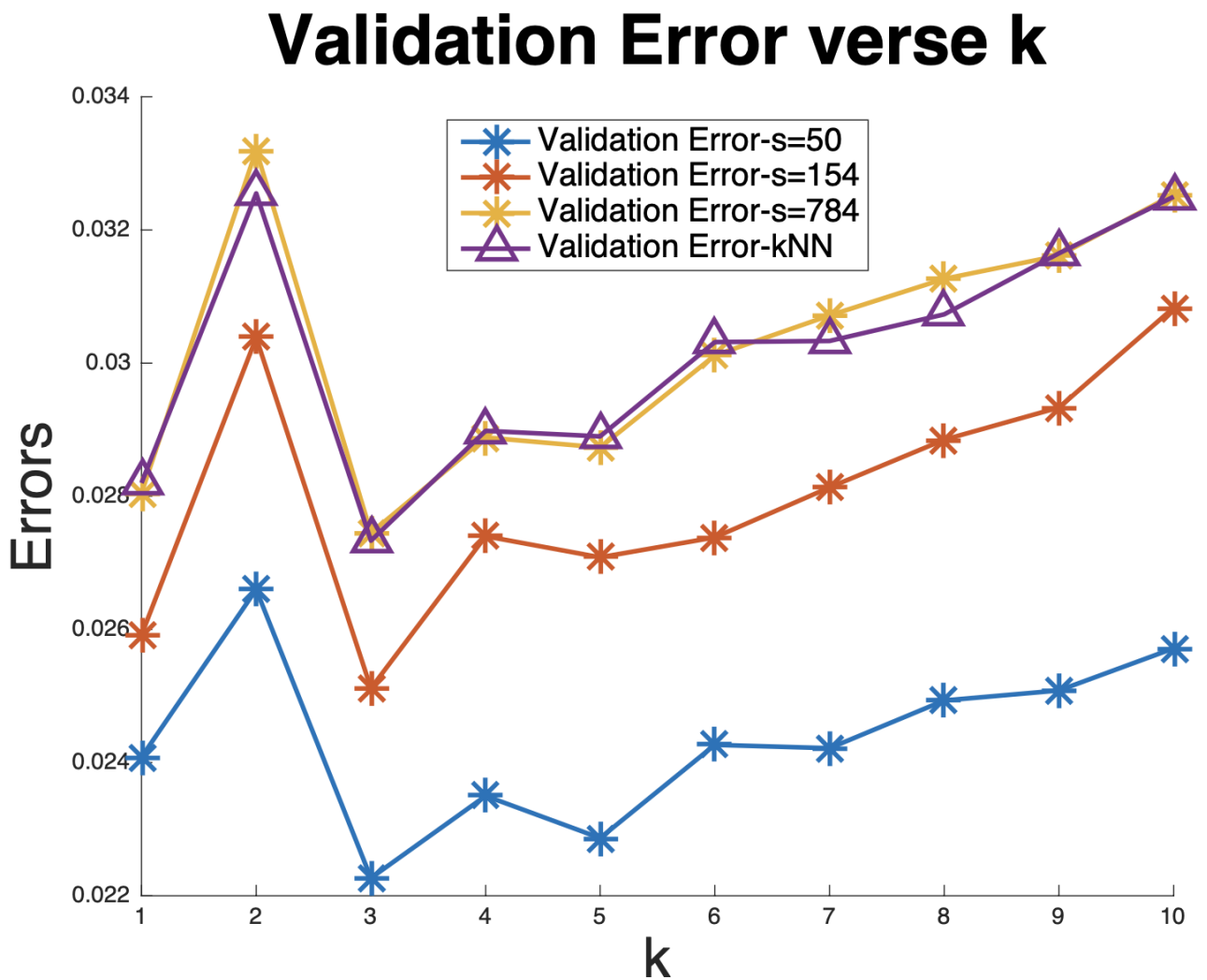
- PCA + kNN
- PCA + nearest local centroid

PCA (s) + 3NN on the MNIST digits

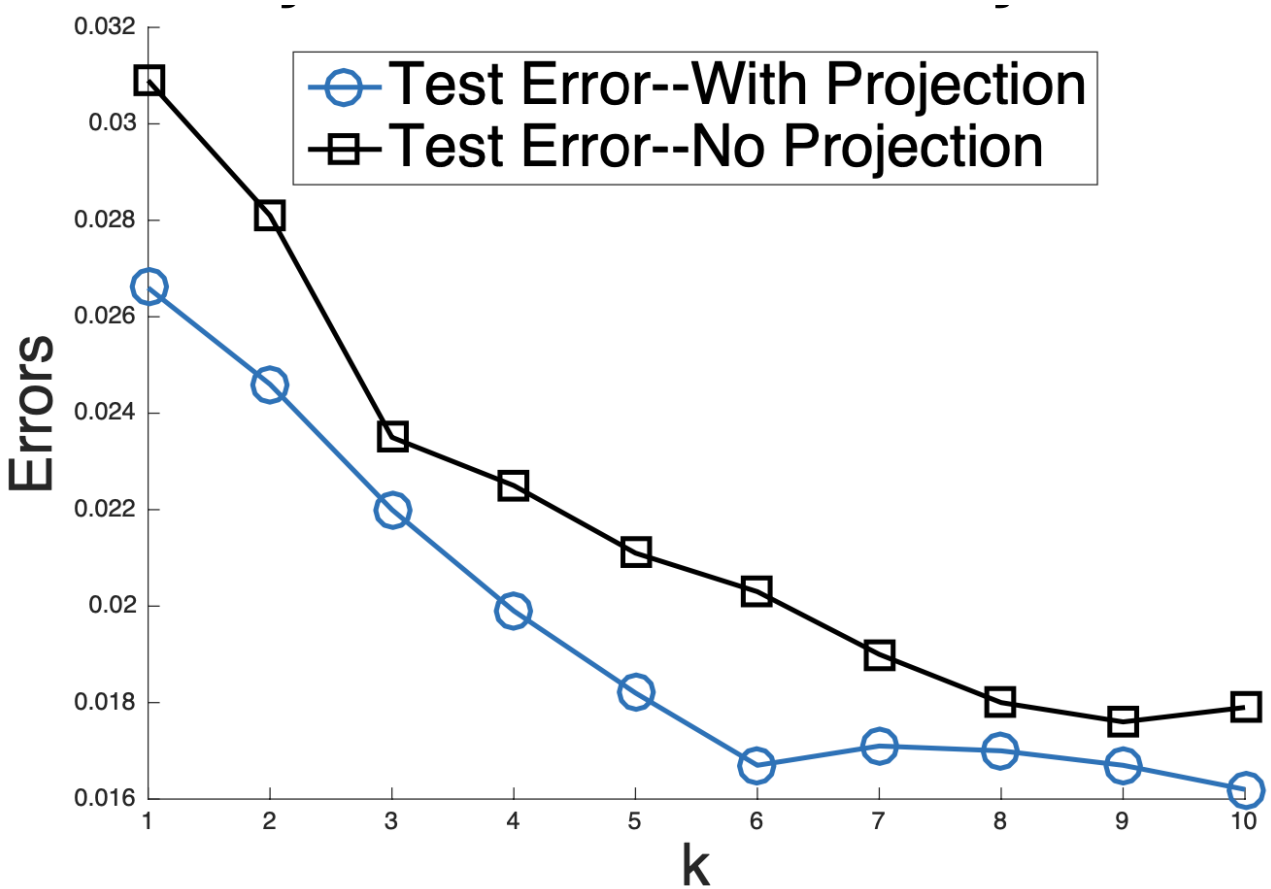


Remark.

- PCA with $s = 154$ preserves 95% variance
- 3NN on the original data has a 2.94% error rate.
- PCA ($s = 50$) + 3NN has the lowest test error (2.50%).


Validation Error verse k
Validation Error-s=50
Validation Error-s=154
Validation Error-s=784
Validation Error-kNN
Errors
k
0.034
0.032
0.03
0.028
0.026
0.024
0.022
1
2
3
4
5
6
7
8
9
10

PCA ($s = 50$) + NLC (k) on the MNIST digits



Some further comments

PCA is an *unsupervised* method, and the 95% criterion is a conservative choice which discards only the directions with smallest amounts of variance.

In the context of classification it is possible to get much lower than this threshold while maintaining or even improving the classification accuracy.

The reason is that large-variance directions are representative, but not necessarily discriminatory.

In practice, one may want to use cross validation to select the optimal projection dimension.

Linear Discriminant Analysis (LDA)

LDA for classification

First, apply global PCA to the labeled data (see slide 6),

$$\mathbf{X}_{\text{train}} \longrightarrow \mathbf{Y}_{\text{train}}, \qquad \text{and} \qquad \mathbf{X}_{\text{test}} \longrightarrow \mathbf{Y}_{\text{test}},$$

to reduce the dimensionality and meanwhile and avoid the singularity issue for LDA.

Next, perform LDA on the PCA-reduced training data $\mathbf{Y}_{\text{train}}$:

$$\mathbf{Y}_{\text{train}} \longrightarrow \mathbf{V}_{\text{lda}}$$
$$\mathbf{Z}_{\text{train}} = \mathbf{Y}_{\text{train}} \cdot \mathbf{V}_{\text{lda}}$$

We extend LDA to the PCA-reduced test data $\mathbf{Y}_{\text{test}}$ as follows:

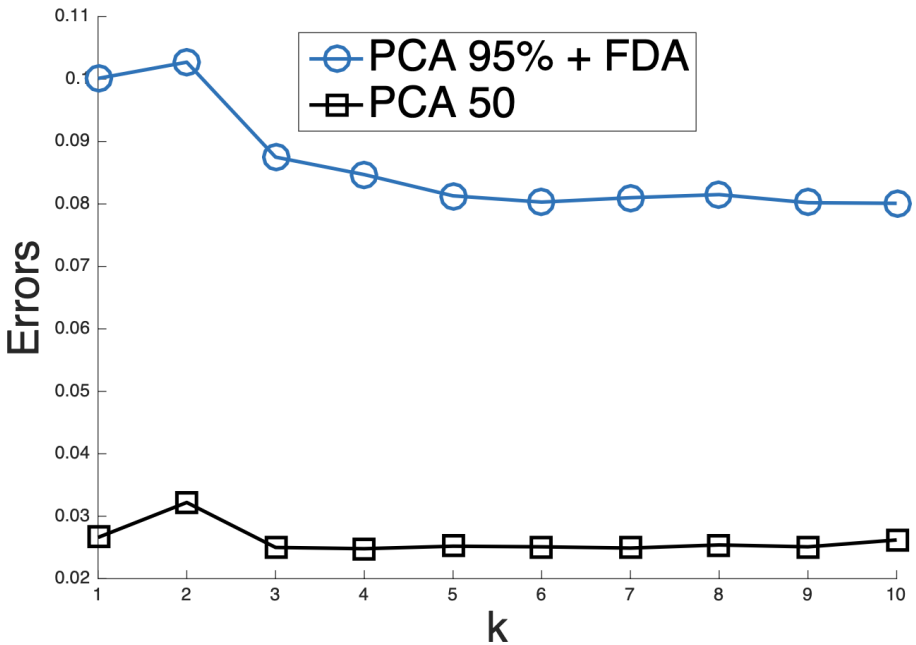
$$\mathbf{Z}_{\text{test}} = \mathbf{Y}_{\text{test}} \cdot \mathbf{V}_{\text{lda}}$$

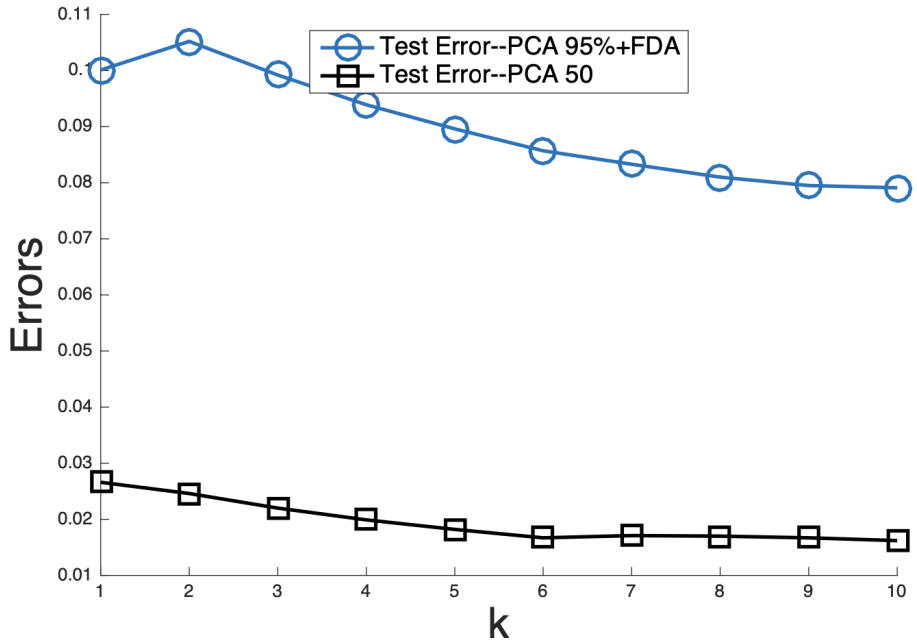
Lastly, just select a classifier to make predictions for $\mathbf{Z}_{\text{test}}$ based on $\mathbf{Z}_{\text{train}}$:

- (PCA +) LDA + kNN
- (PCA +) LDA + nearest local centroid

LDA on the MNIST digits

Left: kNN; right: NLC





LDA is much worse than PCA (50) on this data set!

Two-dimensional LDA (2DLDA)

What is 2DLDA?

Both PCA and LDA require vectorizing the images. The projections are also in vector form.

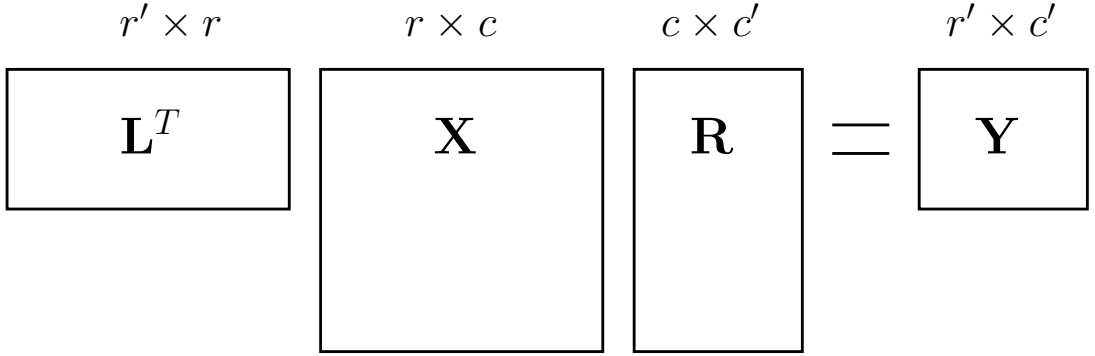
2DLDA treats images as two-dimensional signals and works with matrices directly (no vectorization needed). The projections will still be images (but smaller in size).

2DLDA has the advantage of preserving information along both dimensions (i.e., rows and columns).

How does 2DLDA work?

2DLDA transforms $r \times c$ images to smaller $r' \times c'$ images.

Let $\mathbf{X} \in \mathbb{R}^{r \times c}$ be a given image. The transformation is defined by two tall matrices with orthonormal columns, denoted by $\mathbf{L} \in \mathbb{R}^{r \times r'}$ and $\mathbf{R} \in \mathbb{R}^{c \times c'}$:



Like LDA, 2DLDA finds the best transformations $\mathbf{L}, \mathbf{R}$ by

$$\max_{\mathbf{L},\mathbf{R}} \frac{\text{between-class scatter}}{\text{within-class scatter}}$$

Notation and definitions

Let $\mathbf{X}_i \in \mathbb{R}^{r\times c}, 1 \le i \le n$ be the images in the training set, which consist of k classes $C_1, \ldots, C_k$.

Let

$$\mathbf{M}_j = \frac{1}{n_j} \sum_{\mathbf{X}\in C_j} \mathbf{X}$$

be the (matrix) mean of class j, and

$$\mathbf{M} = \frac{1}{n} \sum_{1\le j\le k} \sum_{\mathbf{X}\in C_j} \mathbf{X} = \frac{1}{n} \sum_j n_j \mathbf{M}_j$$

the global mean.

In the *original* image space, define

- Within-class scatter:

$$s_w^2 = \sum_j \sum_{\mathbf{X} \in C_j} \|\mathbf{X} - \mathbf{M}_j\|_{\mathrm{F}}^2$$

- Between-class scatter:

$$s_b^2 = \sum_j n_j \|\mathbf{M}_j - \mathbf{M}\|_{\mathrm{F}}^2$$

In the *projection* space, define

- Within-class scatter:

$$\begin{aligned}\tilde{s}_w^2 &= \sum_j \sum_{\mathbf{X}\in C_j} \|\mathbf{L}^T\mathbf{X}\mathbf{R} - \mathbf{L}^T\mathbf{M}_j\mathbf{R}\|_{\mathrm{F}}^2 \\ &= \sum_j \sum_{\mathbf{X}\in C_j} \|\mathbf{L}^T(\mathbf{X} - \mathbf{M}_j)\mathbf{R}\|_{\mathrm{F}}^2\end{aligned}$$

- Between-class scatter:

$$\tilde{s}_b^2 = \sum_j n_j \|\mathbf{L}^T(\mathbf{M}_j - \mathbf{M})\mathbf{R}\|_{\mathrm{F}}^2$$

The mathematical formulation of 2DLDA

2DLDA aims to maximize the between-class scatter ($\tilde{s}_b^2$) while minimizing the within-class scatter ($\tilde{s}_w^2$) in the projection space by solving

$$\max_{\mathbf{L},\mathbf{R}} \frac{\tilde{s}_b^2}{\tilde{s}_w^2} = \frac{\sum_j n_j \|\mathbf{L}^T(\mathbf{M}_j - \mathbf{M})\mathbf{R}\|_{\mathrm{F}}^2}{\sum_j \sum_{\mathbf{X}\in C_j} \|\mathbf{L}^T(\mathbf{X} - \mathbf{M}_j)\mathbf{R}\|_{\mathrm{F}}^2}$$

where $\mathbf{L} \in \mathbb{R}^{r\times r'}, \mathbf{R} \in \mathbb{R}^{c\times c'}$ are tall matrices with orthonormal columns.

Note. The projected images will be given by

$$\mathbf{Y}_i = \mathbf{L}^T\mathbf{X}_i\mathbf{R} \in \mathbb{R}^{r'\times c'}, \quad \forall\, i$$

Rewriting the problem

Using the trace properties we first rewrite the within-class scatter as follows

$$\begin{aligned}
&\sum_j \sum_{\mathbf{X}\in C_j} \|\mathbf{L}^T(\mathbf{X}-\mathbf{M}_j)\mathbf{R}\|_{\mathrm{F}}^2 \\
=&\sum_j \sum_{\mathbf{X}\in C_j} \operatorname{trace}\left(\mathbf{L}^T(\mathbf{X}-\mathbf{M}_j)\mathbf{R}\mathbf{R}^T(\mathbf{X}-\mathbf{M}_j)^T\mathbf{L}\right) \\
=&\operatorname{trace}\left(\sum_j \sum_{\mathbf{X}\in C_j} \mathbf{L}^T(\mathbf{X}-\mathbf{M}_j)\mathbf{R}\mathbf{R}^T(\mathbf{X}-\mathbf{M}_j)^T\mathbf{L}\right)
\end{aligned}$$

Note that $\mathbf{L}^T$ and $\mathbf{L}$ may be factored out of the double summation (but still within the trace operator).

Similarly, for the between-class scatter,

$$\sum_j n_j \|\mathbf{L}^T(\mathbf{M}_j - \mathbf{M})\mathbf{R}\|_{\mathrm{F}}^2$$

$$= \mathrm{trace}\left(\sum_j n_j \mathbf{L}^T(\mathbf{M}_j - \mathbf{M})\mathbf{R}\mathbf{R}^T(\mathbf{M}_j - \mathbf{M})^T\mathbf{L}\right)$$

The 2DLDA problem now becomes

$$\max_{\mathbf{L},\mathbf{R}} \frac{\mathrm{trace}\left(\sum_j n_j \mathbf{L}^T(\mathbf{M}_j - \mathbf{M})\mathbf{R}\mathbf{R}^T(\mathbf{M}_j - \mathbf{M})^T\mathbf{L}\right)}{\mathrm{trace}\left(\sum_j \sum_{\mathbf{X}\in C_j} \mathbf{L}^T(\mathbf{X} - \mathbf{M}_j)\mathbf{R}\mathbf{R}^T(\mathbf{X} - \mathbf{M}_j)^T\mathbf{L}\right)}$$

Solving the problem

The joint optimization problem over $\mathbf{L}, \mathbf{R}$ is very difficult to solve.

We consider a special case when $\mathbf{R}$ is given. The problem reduces to

$$\max_{\mathbf{L}} \frac{\text{trace}\left(\mathbf{L}^T \mathbf{S}_b^{\mathbf{R}} \mathbf{L}\right)}{\text{trace}\left(\mathbf{L}^T \mathbf{S}_w^{\mathbf{R}} \mathbf{L}\right)}$$

where

$$\mathbf{S}_w^{\mathbf{R}} = \sum_j \sum_{\mathbf{X} \in C_j} (\mathbf{X} - \mathbf{M}_j)\mathbf{R}\mathbf{R}^T(\mathbf{X} - \mathbf{M}_j)^T \in \mathbb{R}^{r \times r}$$

$$\mathbf{S}_b^{\mathbf{R}} = \sum_j n_j(\mathbf{M}_j - \mathbf{M})\mathbf{R}\mathbf{R}^T(\mathbf{M}_j - \mathbf{M})^T \in \mathbb{R}^{r \times r}$$

The maximizer $\mathbf{L} = [\mathbf{l}_1 \dots \mathbf{l}_{r'}]$ is found by solving

$$\mathbf{S}_b^{\mathbf{R}}\,\mathbf{l}_j = \lambda_j\,\mathbf{S}_w^{\mathbf{R}}\,\mathbf{l}_j \;\longleftrightarrow\; \left(\mathbf{S}_w^{\mathbf{R}}\right)^{-1}\mathbf{S}_b^{\mathbf{R}}\,\mathbf{l}_j = \lambda_j \mathbf{l}_j$$

for each $j = 1, \dots, r'$.

Remark.

- Both matrices $\mathbf{S}_w^{\mathbf{R}}, \mathbf{S}_b^{\mathbf{R}}$ have the size of $r \times r$, and thus are much smaller than their counterparts in LDA which have a size of $d \times d$ with $d = rc$. Therefore, this problem is much easier to solve numerically.

- In general $\mathbf{S}_w^{\mathbf{R}}$ is nonsingular, so the singularity issue with LDA does not exist in 2DLDA.

Similarly, if $\mathbf{L}$ is given to us, then the problem maybe written as

$$\max_{\mathbf{R}} \frac{\operatorname{trace}\left(\sum_j n_j \mathbf{R}^T(\mathbf{M}_j - \mathbf{M})^T \mathbf{L}\mathbf{L}^T(\mathbf{M}_j - \mathbf{M})\mathbf{R}\right)}{\operatorname{trace}\left(\sum_j \sum_{\mathbf{X}\in C_j} \mathbf{R}^T(\mathbf{X} - \mathbf{M}_j)^T \mathbf{L}\mathbf{L}^T(\mathbf{X} - \mathbf{M}_j)\mathbf{R}\right)} = \frac{\operatorname{trace}\left(\mathbf{R}^T \mathbf{S}_b^{\mathbf{L}} \mathbf{R}\right)}{\operatorname{trace}\left(\mathbf{R}^T \mathbf{S}_w^{\mathbf{L}} \mathbf{R}\right)}$$

where

$$\mathbf{S}_w^{\mathbf{L}} = \sum_j \sum_{\mathbf{X}\in C_j} (\mathbf{X} - \mathbf{M}_j)^T \mathbf{L}\mathbf{L}^T(\mathbf{X} - \mathbf{M}_j) \in \mathbb{R}^{c\times c},$$

$$\mathbf{S}_b^{\mathbf{L}} = \sum_j n_j (\mathbf{M}_j - \mathbf{M})^T \mathbf{L}\mathbf{L}^T(\mathbf{M}_j - \mathbf{M}) \in \mathbb{R}^{c\times c}.$$

The corresponding maximizer is given by the first c' eigenvectors of $\left(\mathbf{S}_w^{\mathbf{L}}\right)^{-1} \mathbf{S}_b^{\mathbf{L}} \in \mathbb{R}^{c\times c}$.

Algorithm for 2DLDA

The previous discussions motivate us to solve the 2DLDA problem using an iterative procedure:

1. Initialize $\mathbf{R} = \begin{pmatrix} \mathbf{I}_{c' \times c'} \\ \mathbf{0}_{(c-c') \times c'} \end{pmatrix} \in \mathbb{R}^{c \times c'}$
2. Iterative until convergence:
   - $\mathbf{L} \longleftarrow$ top r' eigenvectors of $\left(\mathbf{S}_w^{\mathbf{R}}\right)^{-1} \mathbf{S}_b^{\mathbf{R}}$
   - $\mathbf{R} \longleftarrow$ top c' eigenvectors of $\left(\mathbf{S}_w^{\mathbf{L}}\right)^{-1} \mathbf{S}_b^{\mathbf{L}}$
3. Return final versions of $\mathbf{L}$ and $\mathbf{R}$

MATLAB code for 2DLDA

2DLDA is not implemented in MATLAB.

However, there is a toolbox available at MATLAB File Exchange:

`http://www.mathworks.com/matlabcentral/fileexchange/20174-2dlda-pk-lda-for-feature-extraction`

The function to use is

[R, L] = iterative2DLDA(trainImages, trainLabels+1, 10, 10, 28, 28)

% Columns are images
% Labels must start at 1

Ways of using 2DLDA

Like LDA, 2DLDA is a supervised dimensionality reduction methods.

It has the following usage:

- 2DLDA + a classifier (e.g., kNN, $k$means, LDA/QDA, Naive Bayes)
- 2DLDA + LDA + a classifier

2DLDA on the MNIST digits

See poster at

`https://www.sjsu.edu/faculty/guangliang.chen/Math285S16/poster-2DLDA.pdf`

Note that the nearest local centroid classifier is formerly referred to as the local $k$means classifier in the poster.

Comparison between LDA and 2DLDA

Both are supervised methods aiming to preserve discriminative information.

- 2DLDA is more flexible (can project data down to any size $r' \times c'$)
- 2DLDA does not have the singularity issue (no PCA needed)
- 2DLDA is harder to solve (as it has two matrices to choose, so that we can only use alternating optimization) but individual linear algebra problems are much easier to solve (as the scatter matrices are smaller)

Lastly, remember that 2DLDA can be used along with LDA.

HW2

This assignment is also based on the Fashion-MNIST data set. Formatting requirements are the same with HW1.

1. Apply the plain kNN classifier, for each $k = 1, \ldots, 12$, to the following two different projections of the data:

   (a) PCA with 95% variation preserved

   (b) PCA with a dimension of your own choice (that would lead to lower test errors than the 95% criterion)

   Plot both sets of test errors against k. How do they compare with those obtained on the original data (i.e., no PCA projection)?

2. Repeat Question 1 with the NLC classifier instead.

3. First use PCA 95% + LDA to reduce the dimensionality of the data set and then apply the plain kNN classifier, for each $k = 1, \ldots, 12$, to the projected data. Plot the test error curve as a function of k and compare with that of PCA 95% + kNN.