MATH 250: Mathematical Data Visualization

Applications: Classification

Peter A. Gao

2025-04-21

San José State University

Outline

- Classification via linear discriminant analysis (based on lecture by Guangliang Chen)

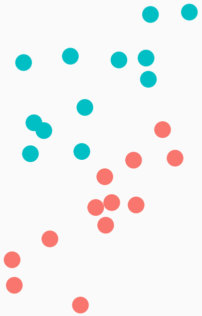
Other readings

- Chapter on LDA (applied)
- PSU Stat 508

Linear discriminant analysis

Recall that PCA is a linear dimension reduction technique that identifies the subspace spanned by the maximum-variance directions.

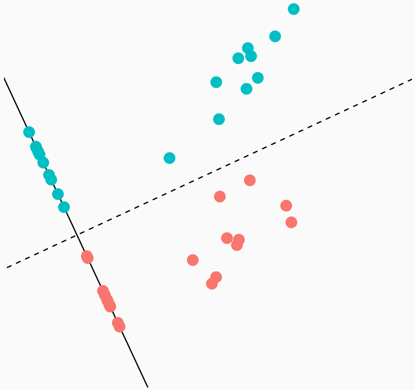
When performing classification, we may want to identify a lower dimensional representation that preserves differences between groups.



Linear discriminant analysis

Linear discriminant analysis seeks to identify the direction that best **separates** the various classes of the dataset.

Note that projections onto parallel lines yield the same separation.



Two-class linear discriminant analysis

Objective: Given data $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^p$ belonging to disjoint classes C_1 and C_2, identify a line that best separates the classes:

$$\mathbf{w}(t) = t\mathbf{w} + \mathbf{b}, \quad t \in \mathbb{R}$$

where $\mathbf{w}, \mathbf{b} \in \mathbb{R}^p$ and $||\mathbf{w}|| = 1$.

Two-class linear discriminant analysis

Since parallel lines have the same separation, we focus on lines passing through the origin:

$$\mathbf{w}(t) = t\mathbf{w}, \quad t \in \mathbb{R}$$

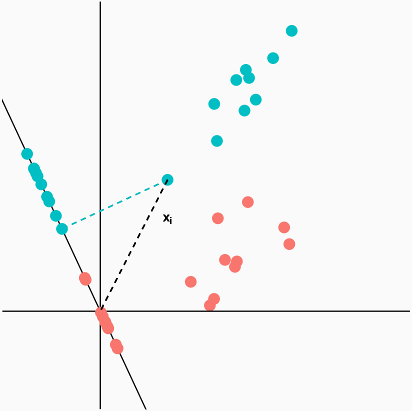
where $||\mathbf{w}|| = 1$. Consider that the projections of the data onto this line are given by

$$\mathbf{p}_i = (\mathbf{x}_i^\top \mathbf{w})\mathbf{w} := a_i \mathbf{w}, \quad i = 1, \ldots, n$$

Two-class linear discriminant analysis

Given a projection onto a line, how can we quantify the “amount of separation” between the classes?

Our goal will be to identify the direction that yields the best separation.



Two-class linear discriminant analysis

One possibility is simply to measure the distances between the two means along the line: $|\mu_1 - \mu_2|$, where

$$\mu_1 = \frac{1}{n_1}\sum_{i \in C_1} \mathbf{w}^\top \mathbf{x}_i^\top = \mathbf{w}^\top \left(\frac{1}{n_1} \sum_{i \in C_1} \mathbf{x}_i \right) = \mathbf{w}^\top \mathbf{m}_1$$

where

$$\mathbf{m_1} = \frac{1}{n_1} \sum_{i \in C_1} \mathbf{x}_i$$

Similarly,

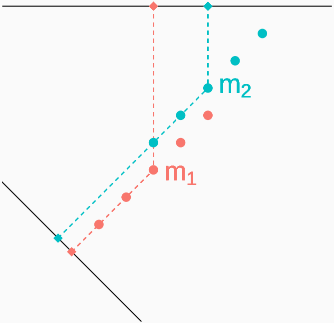
$$\mu_2 = \mathbf{w}^\top \mathbf{m}_2, \quad \mathbf{m_2} = \frac{1}{n_2} \sum_{i \in C_2} \mathbf{x}_i$$

Two-class linear discriminant analysis

In other words, the problem becomes

$$\max_{\mathbf{w}:||\mathbf{w}||=1} |\mu_1 - \mu_2| = \max_{\mathbf{w}:||\mathbf{w}||=1} |\mathbf{w}^\top(\mathbf{m}_1 - \mathbf{m}_2)|$$

However, it turns out this problem does not necessarily work well, since separating the means does not always separate the data well.



Two-class linear discriminant analysis

One way to improve this criterion is to consider the **within-class variances** of the projections:

$$s_1^2 = \sum_{i \in C_1} (\mathbf{w}^\top \mathbf{x}_i - \mu_1)^2, \quad s_2^2 = \sum_{i \in C_i} (\mathbf{w}^\top \mathbf{x}_i - \mu_i)^2$$

Ideally the direction used will ensure that the class means are well-separated (maximizing $(\mu_1 - \mu_2)^2$), while also minimizing the within-class variances s_1^2 and s_2^2:

$$\max_{\mathbf{w}:||\mathbf{w}||=1} \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2}$$

Obtaining a solution

How do we solve this optimization problem? First, consider that

$$\begin{aligned}(\mu_1 - \mu_2)^2 &= (\mathbf{w}^\top \mathbf{m}_1 - \mathbf{w}^\top \mathbf{m}_2)^2 \\ &= (\mathbf{w}^\top (\mathbf{m}_1 - \mathbf{m}_2))^2 \\ &= \mathbf{w}^\top (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\top \mathbf{w} \\ &:= \mathbf{w}^\top S_b \mathbf{w}\end{aligned}$$

where

$$S_b = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\top \in \mathbb{R}^{p\times p}$$

is called the between-class scatter matrix. What can we say about S_b?

Obtaining a solution

Observe that S_b is square, symmetric, and positive semidefinite.

In addition, S_b is rank one, which in turn implies that it has only one positive eigenvalue.

Obtaining a solution

For each class j, the variance of the projections is

$$\begin{aligned} s_j^2 &= \sum_{i \in C_j} (\mathbf{w}^\top \mathbf{x}_i - \mu_j)^2 \\ &= \sum_{i \in C_j} (\mathbf{w}^\top \mathbf{x}_i - \mathbf{w}^\top \mathbf{m}_j)^2 \\ &= \sum_{i \in C_j} \mathbf{w}^\top (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^\top \mathbf{w} \\ &= \mathbf{w}^\top \left(\sum_{i \in C_j} (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^\top \right) \mathbf{w} \\ &= \mathbf{w}^\top S_j \mathbf{w} \end{aligned}$$

where

$$S_j = \sum_{i \in C_j} (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^\top$$

is called the within-class scatter matrix for class j.

Obtaining a solution

Thus,

$$s_1^2 + s_2^2 = \mathbf{w}^\top S_1 \mathbf{w} + \mathbf{w}^\top S_2 \mathbf{w} = \mathbf{w}^\top (S_1 + S_2)\mathbf{w} = \mathbf{w}^\top S_w \mathbf{w}$$

where

$$S_w = S_1 + S_2 = \sum_j \sum_{i \in C_j} (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^\top$$

is called the **total within-class scatter matrix** of the data.

What do we know about S_w?

Obtaining a solution

Thus,

$$s_1^2 + s_2^2 = \mathbf{w}^\top S_1 \mathbf{w} + \mathbf{w}^\top S_2 \mathbf{w} = \mathbf{w}^\top (S_1 + S_2)\mathbf{w} = \mathbf{w}^\top S_w \mathbf{w}$$

where

$$S_w = S_1 + S_2 = \sum_j \sum_{i \in C_j} (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^\top$$

is called the **total within-class scatter matrix** of the data.

What do we know about S_w?

S_w is square, symmetric, and positive semidefinite.

Obtaining a solution

Thus,

$$\max_{\mathbf{w}:||\mathbf{w}||=1} \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2} = \max_{\mathbf{w}:||\mathbf{w}||=1} \frac{\mathbf{w}^\top S_b \mathbf{w}}{\mathbf{w}^\top S_w \mathbf{w}}$$

Look familiar? When S_w is nonsingular, this is a **generalized Rayleigh quotient**.

Theorem

If S_w is nonsingular, the above optimization problem is solved by the generalized eigenvector $\mathbf{w}_1$ of (S_b, S_w):

$$S_b \mathbf{w}_1 = \lambda_1 S_w \mathbf{w}_1 \iff S_w^{-1} S_b \mathbf{w}_1 = \lambda_1 \mathbf{w}_1$$

where λ_1 is the largest generalized eigenvalue of (S_b, S_w).

Review: Generalized Rayleigh quotients

Proposed solution:

$$S_b \mathbf{w}_1 = \lambda_1 S_w \mathbf{w}_1 \iff S_w^{-1} S_b \mathbf{w}_1 = \lambda_1 \mathbf{w}_1$$

Proof: Since S_w is positive definite, it has a square root $S_w^{1/2}$. If $\mathbf{y} = S_w^{1/2}\mathbf{w}$, then

$$\mathbf{w}^\top S_w \mathbf{w} = \mathbf{y}^\top \mathbf{y}$$

Moreover,

$$\mathbf{w}^\top S_b \mathbf{w} = \mathbf{y}^\top (S_w^{-1/2})^\top S_b S_w^{-1/2} \mathbf{y}$$

so the generalized Rayleigh quotient problem is now

$$\max_{\mathbf{w}:||\mathbf{w}||=1} \frac{\mathbf{w}^\top S_b \mathbf{w}}{\mathbf{w}^\top S_w \mathbf{w}} = \max_{\mathbf{y}:||\mathbf{y}||=1} \frac{\mathbf{y}^\top (S_w^{-1/2})^\top S_b S_w^{-1/2} \mathbf{y}}{\mathbf{y}^\top \mathbf{y}}$$

Review: Generalized Rayleigh quotients

Thus, we obtain the following solution:

$$\begin{aligned}(S_w^{-1/2})^\top S_b S_w^{-1/2}\mathbf{y} = \lambda_1 \mathbf{y} \iff& (S_w^{-1/2})^\top S_b S_w^{-1/2} S_w^{1/2}\mathbf{w} = \lambda_1 S_w^{1/2}\mathbf{w} \\ \iff& S_w^{-1} S_b \mathbf{w}_1 = \lambda_1 \mathbf{w}_1\end{aligned}$$

Note that for the LDA problem, $\mathrm{rank}(S_w^{-1}S_b) = \mathrm{rank}(S_b) = 1$ so λ_1 is the only non-zero eigenvalue (and it is positive).

Review: Generalized Rayleigh quotients

Alternatively,

$$\max_{\mathbf{w}:||\mathbf{w}||=1} \frac{\mathbf{w}^\top S_b \mathbf{w}}{\mathbf{w}^\top S_w \mathbf{w}}$$

is equivalent to the constrained optimization problem

$$\max_{\mathbf{w}} \mathbf{w}^\top S_b \mathbf{w} \;\text{ subject to }\; \mathbf{w}^\top S_w \mathbf{w} = 1$$

We can utilize the method of Lagrange multipliers to compute a solution (exercise).

Computation

Mathematically, we can solve this problem as follows:

1. Invert the $p \times p$ matrix S_w.
2. Compute $S_w^{-1} S_b$
3. Solve the eigenvalue problem $S_w^{-1} S_b \mathbf{w}_1 = \lambda_1 \mathbf{w}_1$.

This will be computationally expensive (especially inverting S_w).

Computation

Alternatively, observe that

$$\begin{aligned} \lambda_1 \mathbf{w}_1 &= S_w^{-1} S_b \mathbf{w}_1 \\ &= S_w^{-1} (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\top \mathbf{w}_1 \end{aligned}$$

Moreover, $(\mathbf{m}_1 - \mathbf{m}_2)^\top \mathbf{w}_1$ is a scalar. Therefore,

$$\mathbf{w}_1 \propto S_w^{-1} (\mathbf{m}_1 - \mathbf{m}_2)$$

Instaed of computing this directly (and inverting S_w), we can solve the linear system

$$= S_w (\mathbf{m}_1 - \mathbf{m}_2) = \mathbf{b}$$

for $\mathbf{b} \in \mathbb{R}^p$.

Two-class linear discriminant analysis: summary

The linear discriminant is in the direction

$$\mathbf{w} \propto S_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

which solves the optimization problem

$$\max_{\mathbf{w}:||\mathbf{w}||=1} \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2} = \max_{\mathbf{w}:||\mathbf{w}||=1} \frac{\mathbf{w}^\top S_b \mathbf{w}}{\mathbf{w}^\top S_w \mathbf{w}}$$

where

$$S_b = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\top$$

$$S_w = S_1 + S_2 = \sum_j \sum_{i \in C_j} (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^\top$$

Dealing with singularity of S_w

Note that solving the generalized eigenvalue problem requires that the **total within-class scatter matrix** S_w is non singular, which yields a solution to

$$\max_{\mathbf{w}:||\mathbf{w}||=1} \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2} = \max_{\mathbf{w}:||\mathbf{w}||=1} \frac{\mathbf{w}^\top S_b \mathbf{w}}{\mathbf{w}^\top S_w \mathbf{w}}$$

However, S_w may be singular (or nearly singular), especially if p is large.

This results when the n rows of the data matrix do not span $\mathbb{R}^p$.

Dealing with singularity of S_w

A simple solution is to apply PCA to the data matrix X, yielding principal components Y.

Note that in this case, we do not keep all of the principal components. One rule of thumb is to keep as many PCs as needed to capture 95% of the variation in the data.

We can then perform LDA using the rank-k reduced data matrix.

Dealing with singularity of S_w

Another approach is to regularize S_w to create a nonsingular matrix:

$$\begin{aligned} S_w^\beta &= S_w + \beta I_p \\ &= Q \Lambda Q^\top + \beta I_p \\ &= Q(\Lambda + \beta I_p) Q^\top \end{aligned}$$

where $\beta > 0$ is a tuning parameter.

Classification

We can classify new observations by:

1. Projecting the new observations onto the linear discriminant direction(s) $\mathbf{y}_i = \mathbf{w}^\top \mathbf{x}_i$
2. Assume that for each class j, the projected observations $\mathbf{y}_i \sim N(\mu_j, \Sigma_j)$ for $i \in C_j$. This is called the **class conditional distribution**
3. Compute the class posterior

$$p(i \in C_j \mid X, \theta) \propto \pi_j f(\mathbf{y}_i \mid \mu_j, \Sigma_j)$$

where X is the data matrix, θ denotes model parameters, π_c is the **prior probability** of belonging to class j, and $f(\mathbf{y}_i \mid \mu_j, \Sigma_j)$ is the **class conditional density**.

Two-class linear discriminant analysis: example

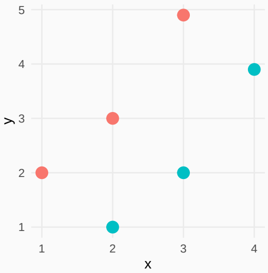
Consider the following example:

Class 1 has points

$$\{(1,2),(2,3),(3,4.9)\}$$

and Class 2 has points

$$\{(2,1),(3,2),(4,3.9)\}$$



Two-class linear discriminant analysis: example

Then $\mathbf{m}_1 = (2, 3.3)^\top$ and

$$\begin{aligned}
S_1 &= \left(\begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 2 \\ 3.3 \end{bmatrix}\right)\left(\begin{bmatrix} 1 & 2 \end{bmatrix} - \begin{bmatrix} 2 & 3.3 \end{bmatrix}\right) \\
&+ \left(\begin{bmatrix} 2 \\ 3 \end{bmatrix} - \begin{bmatrix} 2 \\ 3.3 \end{bmatrix}\right)\left(\begin{bmatrix} 2 & 3 \end{bmatrix} - \begin{bmatrix} 2 & 3.3 \end{bmatrix}\right) \\
&+ \left(\begin{bmatrix} 3 \\ 4.9 \end{bmatrix} - \begin{bmatrix} 2 \\ 3.3 \end{bmatrix}\right)\left(\begin{bmatrix} 3 & 4.9 \end{bmatrix} - \begin{bmatrix} 2 & 3.3 \end{bmatrix}\right) \\
&= \begin{bmatrix} 1 & 1.3 \\ 1.3 & 1.69 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0.09 \end{bmatrix} + \begin{bmatrix} 1 & 1.6 \\ 1.6 & 2.56 \end{bmatrix} \\
&= \begin{bmatrix} 2 & 2.9 \\ 2.9 & 4.34 \end{bmatrix}
\end{aligned}$$

Two-class linear discriminant analysis: example

Next,

$$S_2 = \begin{bmatrix} 2 & 2.9 \\ 2.9 & 4.34 \end{bmatrix}$$

and

$$S_w = \begin{bmatrix} 4 & 5.8 \\ 5.8 & 8.68 \end{bmatrix}$$

Thus, the optimal linear discriminant is

$$\mathbf{w} \propto S_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2) = (-13.4074, 9.0741)^\top \propto (-0.8282, 0.5605)^\top$$

Two-class linear discriminant analysis: example

And the projections onto the linear discriminant for

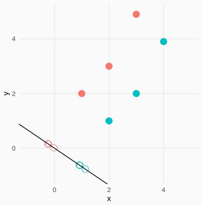
Class 1 are

$$\{0.2928, 0.0252, 0.2619\}$$

and Class 2 are

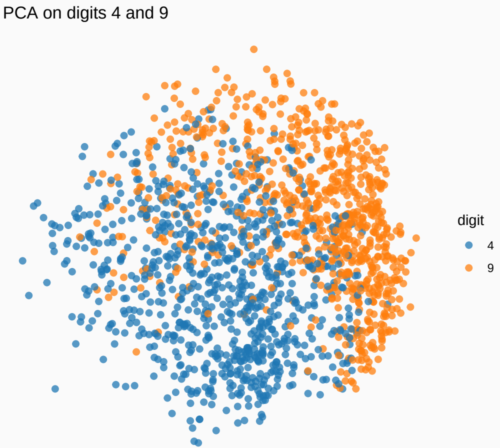
$$\{-1.0958, -1.3635, -1.1267\}$$

.



USPS digits data

A two-dimensional PCA does not separate the digits 4 and 9 well:



USPS digits data

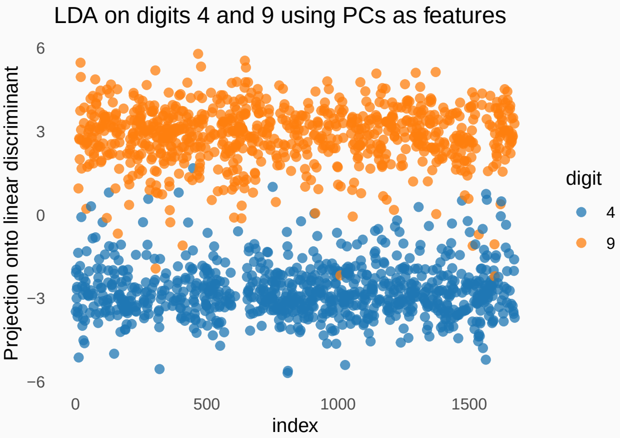
```r
library(IMIFA)
library(ggsci)
# Load USPS data
data("USPSdigits")
usps <- list()
usps$data <-
  (as.matrix(rbind(USPSdigits$train[, -1],
                   USPSdigits$test[, -1])) - -1) * 255 / 2
usps$label <- c(USPSdigits$train[, 1], USPSdigits$test[, 1])
usps_4_9 <- usps$data[usps$label %in% c(4, 9), ]

# Perform PCA
pca_4_9 <- prcomp(usps_4_9)
```

USPS digits data

```
plot_dat <-
  data.frame(PC1 = pca_4_9$x[, 1],
             PC2 = pca_4_9$x[, 2],
             label = usps$label[usps$label %in% c(4, 9)])
ggplot(plot_dat, aes(x = PC1, y = PC2, color = as.factor(label))) +
         geom_point(alpha = .75, size = 2) +
  theme_minimal() +
  scale_color_d3(name = "digit", scale_name = "category10") +
  ggtitle("PCA on digits 4 and 9") +
  theme(aspect.ratio = 1,
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.text = element_blank(),
        axis.title =  element_blank())
```

USPS digits data



USPS digits data

```r
library(MASS)
pca_95_idx <- which(cumsum(pca_4_9$sdev^2) / sum(pca_4_9$sdev^2) < .95)
pca_4_9_df <- as.data.frame(pca_4_9$x[, pca_95_idx]) |>
  mutate(label = usps$label[usps$label %in% c(4, 9)])
lda_4_9 <- lda(label ~., pca_4_9_df)

# compute projections
projections <-
  as.vector(pca_4_9$x[, pca_95_idx] %*% lda_4_9$scaling)
```

USPS digits data

```r
plot_dat <- data.frame(
  index = 1:nrow(usps_4_9),
  a = projections,
  label = usps$label[usps$label %in% c(4, 9)]
)
ggplot(plot_dat, aes(x = index, y = a, color = as.factor(label))) +
         geom_point(alpha = .75, size = 2) +
  theme_minimal() +
  scale_color_d3(name = "digit", scale_name = "category10") +
  ggtitle("LDA on digits 4 and 9 using PCs as features") +
  ylab("Projection onto linear discriminant") +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```

Multiple class linear discriminant analysis

What if we have $c \geq 3$ classes? We can use the same idea, projecting data such that:

1. Points in the same class are tightly clustered together (minimize within-class variation)
2. The centers of the classes are as far apart from each other as possible (maximize between-class separation).

Multiple class linear discriminant analysis

We can still use the **total within-class scatter** matrix to describe the within-class variation:

$$\sum_{j=1}^{c} s_j^2 = \sum_j \mathbf{w}^\top S_j \mathbf{w} = \mathbf{w}^\top S_w \mathbf{w}$$

where

$$S_w = \sum_j S_j = \sum_j \sum_{i \in C_j} (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^\top$$

Multiple class linear discriminant analysis

For multiple class LDA, we define the **between-class scatter** as follows:

$$\sum_{j=1}^{c} n_j(\mu_j - \mu)^2 \quad \text{where} \;\; \mu = \frac{1}{n}\sum_{j=1}^{c} n_j \mu_j \quad \text{(projected center)}$$

Consider that

$$\mu = \frac{1}{n}\sum_{j=1}^{c} n_j(\mathbf{w}^\top \mathbf{m}_j) = \mathbf{w}^\top \left(\sum_{j=1}^{c} n_j \mathbf{m}_j\right) = \mathbf{w}^\top \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\right) = \mathbf{w}^\top \mathbf{m}$$

where $\mathbf{m}$ is the global centroid of $\mathbf{x}_1, \dots, \mathbf{x}_n$.

Multiple class linear discriminant analysis

The **between-class scatter** can be further simplified:

$$\begin{aligned}\sum_{j=1}^{c} n_j(\mu_j - \mu)^2 &= \sum_j n_j(\mathbf{w}^\top(\mathbf{m}_j - \mathbf{m}))^2 \\ &= \sum_j n_j(\mathbf{w}^\top(\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^\top \mathbf{w} \\ &= \mathbf{w}^\top(\sum_j n_j(\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^\top)\mathbf{w} \\ &= \mathbf{w} S_b \mathbf{w}\end{aligned}$$

where

$$S_b = \sum_j n_j(\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^\top$$

is the new between-class scatter matrix.

Multiple class linear discriminant analysis

This again yields a generalized Rayleigh quotient problem:

$$\max_{\mathbf{w}:||\mathbf{w}||=1} \frac{\sum_j n_j(\mu_j - \mu)^2}{\sum_j s_j^2} = \max_{\mathbf{w}:||\mathbf{w}||=1} \frac{\mathbf{w}^\top S_b \mathbf{w}}{\mathbf{w}^\top S_w \mathbf{w}}$$

Again, if S_w is nonsingular, then the solution the generalized eigenvector $\mathbf{w}_1$ of (S_b, S_w):

$$S_b \mathbf{w} = \lambda_1 S_w \mathbf{w}_1 \iff S_w^{-1} S_b \mathbf{w}_1 = \lambda_1 \mathbf{w}_1$$

where λ_1 is the largest generalized eigenvalue of (S_b, S_w).

Multiple class linear discriminant analysis

Note that now, we cannot simply use $\mathbf{w} \propto S_w^{-1}(\mathbf{m}_1 - \mathbf{m_2})$:

$$\lambda_1 S_w \mathbf{w}_1 = S_w^{-1} S_b \mathbf{w}_1 = \sum_{j=1}^{c} n_j S_w^{-1}(\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^\top \mathbf{w}_1$$

so all we know is that

$$\mathbf{w}_1 \in \mathrm{Span}\{S_w^{-1}(\mathbf{m}_1 - \mathbf{m}), \dots, S_w^{-1}(\mathbf{m}_c - \mathbf{m})\}$$

Thus, $\mathbf{w}_1$ must be computed by solving the generalized eigenvalue problem.

Multiple class linear discriminant analysis

Recall that for the two-class problem, S_b was rank 1, so there was only one non zero eigenvalue. It turns out that

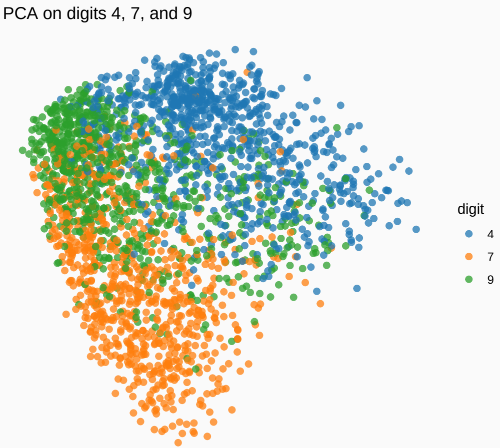
$$\mathrm{rank}(S_w^{-1} S_b) = \mathrm{rank}(S_b) \leq c - 1 \qquad \text{(exercise)}$$

so that LDA can be used to find at most a $c-1$-dimensional space upon which to project the data.

Multiple-class linear discriminant analysis: summary

1. Center and scale the data and find the class centroids $\mathbf{m}_1, \dots, \mathbf{m}_c$.
2. Compute S_w (within-class scatter) and S_b (between-class scatter).
3. Solve the generalized eigenvalue problem $S_b\mathbf{w} = \lambda S_w\mathbf{w}$ and find all non-zero eigenvalues.
4. Project the data onto the corresponding eigenvector(s).

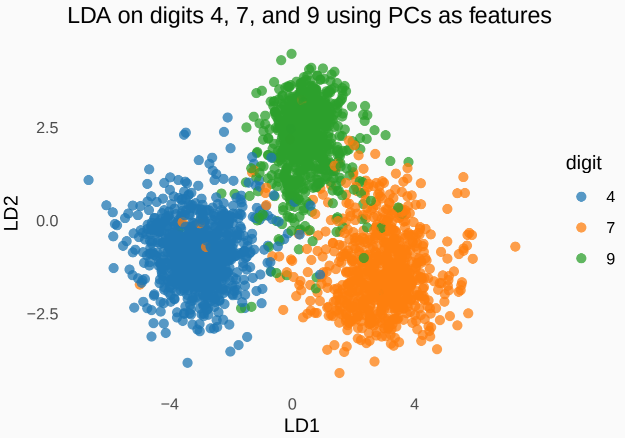
USPS digits data: Multiple classes



USPS digits data: Multiple classes

```
usps_4_7_9 <- usps$data[usps$label %in% c(4, 7, 9), ]
pca_4_7_9 <- prcomp(usps_4_7_9)
plot_dat <-
  data.frame(PC1 = pca_4_7_9$x[, 1],
             PC2 = pca_4_7_9$x[, 2],
             label = usps$label[usps$label %in% c(4, 7, 9)])
ggplot(plot_dat,
       aes(x = PC1, y = PC2, color = as.factor(label))) +
  geom_point(alpha = .75, size = 2) +
  theme_minimal() +
  scale_color_d3(name = "digit", scale_name = "category10") +
  ggtitle("PCA on digits 4, 7, and 9") +
  theme(aspect.ratio = 1,
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.text = element_blank(),
        axis.title =  element_blank())
```

USPS digits data: Multiple classes



USPS digits data: Multiple classes

```
pca_95_idx <-
  which(cumsum(pca_4_7_9$sdev^2) /
          sum(pca_4_7_9$sdev^2) < .95)
pca_4_7_9_df <-
  as.data.frame(pca_4_7_9$x[, pca_95_idx]) |>
  mutate(label = usps$label[usps$label %in% c(4, 7, 9)])
lda_4_7_9 <- lda(label ~., pca_4_7_9_df)
ld1 <- as.vector(pca_4_7_9$x[, pca_95_idx] %*% lda_4_7_9$scaling[,1])
ld2 <- as.vector(pca_4_7_9$x[, pca_95_idx] %*% lda_4_7_9$scaling[,2])
```

USPS digits data: Multiple classes

```
plot_dat <- data.frame(
  index = 1:nrow(usps_4_7_9),
  ld1 = ld1,
  ld2 = ld2,
  label = usps$label[usps$label %in% c(4, 7, 9)]
)
ggplot(plot_dat, aes(x = ld1, y = ld2, color = as.factor(label))) +
         geom_point(alpha = .75, size = 2) +
  theme_minimal() +
  scale_color_d3(name = "digit", scale_name = "category10") +
  ggtitle("LDA on digits 4, 7, and 9 using PCs as features") +
  xlab("LD1") + ylab("LD2") +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```

Darlingtonia data

```
darl <-
  read.csv(
    paste0("https://raw.githubusercontent.com/",
           "jon-bakker/",
           "appliedmultivariatestatistics/main/",
           "Darlingtonia_GE_Table12.1.csv"),
    header = TRUE
  )
```

Darlingtonia data

```
options(width = 80)
head(darl)
```

```
  site plant height mouth.diam tube.diam keel.diam wing1.length wing2.length
1  TJH     1    654       38.4      16.6       6.4           85           76
2  TJH     2    413       22.2      17.2       5.9           55           26
3  TJH     3    610       31.2      19.9       6.7           62           60
4  TJH     4    546       34.4      20.8       6.3           84           79
5  TJH     5    665       30.5      20.4       6.6           60           51
6  TJH     6    665       33.6      19.5       6.6           84           66
  wingsprea hoodmass.g tubemass.g wingmass.g
1        55       1.38       3.54       0.29
2        60       0.49       1.48       0.06
3        78       0.60       2.20       0.16
4        95       1.12       2.95       0.24
5        30       0.67       3.36       0.08
6        82       1.27       4.05       0.21
```

Darlingtonia data

```r
darl_data <- scale(darl[,3:ncol(darl)])
darl_da <- lda(x = darl_data, grouping = darl$site)
darl_da <- lda(darl$site ~ darl_data) # equivalent
darl_da$prior
```

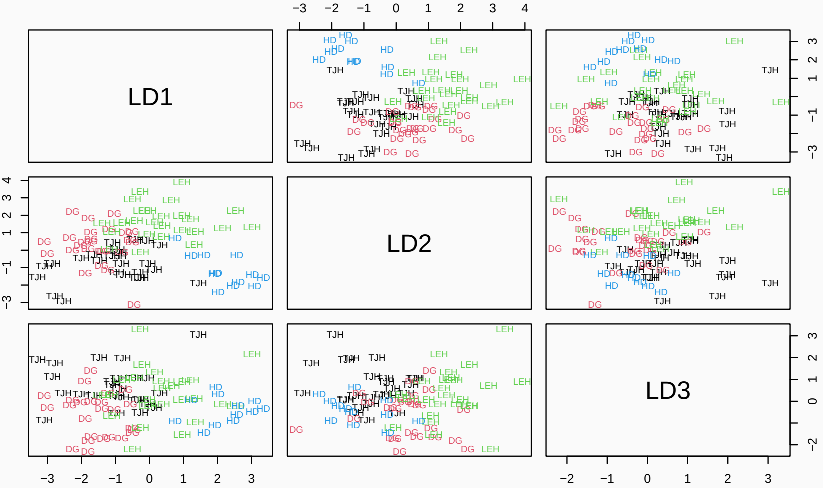
```
       DG        HD       LEH       TJH
0.2873563 0.1379310 0.2873563 0.2873563
```

```r
darl_da$scaling
```

```
                             LD1        LD2         LD3
darl_dataheight       1.40966787  0.2250927 -0.03191844
darl_datamouth.diam  -0.76395010  0.6050286  0.45844178
darl_datatube.diam    0.82241013  0.1477133  0.43550979
darl_datakeel.diam   -0.17750124 -0.7506384 -0.35928102
darl_datawing1.length 0.34256319  1.3641048 -0.62743017
darl_datawing2.length -0.05359159 -0.5310177 -1.25761674
darl_datawingsprea    0.38527171  0.2508244  1.06471559
darl_datahoodmass.g  -0.20249906 -1.4065062  0.40370294
darl_datatubemass.g  -1.58283705  0.1424601 -0.06520404
darl_datawingmass.g   0.01278684  0.0834041  0.25153893
```

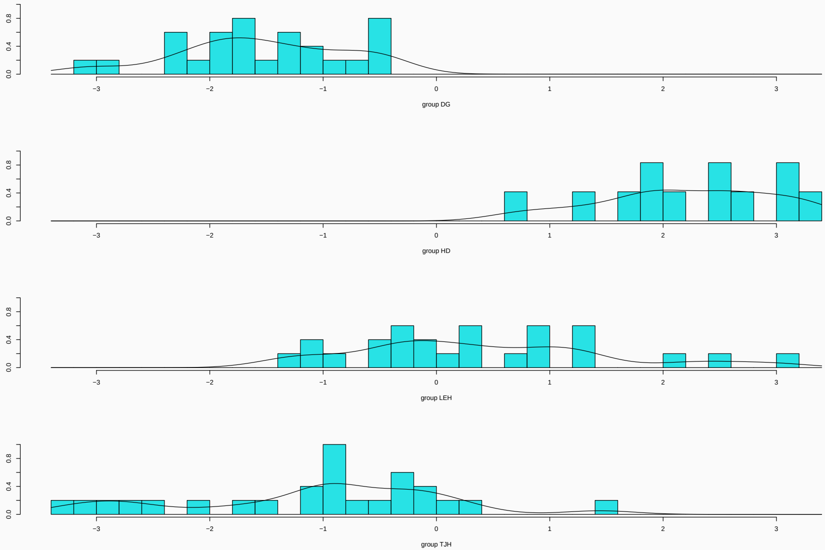
Darlingtonia data

```
plot(darl_da, col = match(darl$site, unique(darl$site)))
```



Darlingtonia data

```
plot(darl_da, dimen = 1, type = "both")
```



Darlingtonia data

```r
darl_predict <- predict(darl_da)
darl_table <- table(darl$site, darl_predict$class)
darl_table
```

```
     DG HD LEH TJH
DG   18  0   2   5
HD    0 11   1   0
LEH   3  0  21   1
TJH   2  0   3  20
```

Prediction

How might dimension reduction methods be helpful in prediction?

Examples:

- Matrix completion (Netflix prize)
- Image denoising
- Timeseries forecasting
- Spatial interpolation (kriging)

In all of these cases, we predict a large data matrix/vector whose entries are only partially or noisily observed.

References