LINGUIST 592B Week 10: Acoustic spaces

Kristine Yu

Department of Linguistics, UMass Amherst

April 1, 2014
1 Administrative, etc.

2 Linear discriminant analysis
   • Introduction
   • Data frame manipulation
   • LDA in R
Notes and reminders

- Projects
  - VS: progress report on coding, literature selection due Thursday
Notes and reminders

- Projects
  - VS: progress report on coding, literature selection due Thursday
  - Spaces: progress report on data and parameter extraction, research question, literature selection due Thursday
Notes and reminders

- Projects
  - VS: progress report on coding, literature selection due Thursday
  - Spaces: progress report on data and parameter extraction, research question, literature selection due Thursday
  - Literature report: due next Thursday
Notes and reminders

- Projects
  - VS: progress report on coding, literature selection due Thursday
  - Spaces: progress report on data and parameter extraction, research question, literature selection due Thursday
  - Literature report: due next Thursday

- Note on logs for HW 4; solutions + Who is Fourier up at Dropbox site
Dispersion depends on dimensions of space

**Figure**: Hillenbrand vowel data, men: American English vowels in \( \langle F_{1\text{ss}}, F_{2\text{ss}} \rangle \) space
Dispersion depends on dimensions of space

Figure: Hillenbrand vowel data, men: American English vowels in $(F_{120\%}, F_{180\%})$ space
Objective: find the line (hyperplane) that maximizes separation of classes (need to define class separation metric)
Objective: find the line (hyperplane) that **maximizes separation of classes** (need to define class separation metric)

Under Fisher’s definition of class separation, optimal classifier for Gaussians of equal covariance (equal size and shape)
Linear discriminant analysis: geometric intuition

Don’t project there!  Project here!

- Project onto axis to maximize ratio of **between-class** to **within-class scatter**
- **Between-class scatter**: roughly, distance between class means
- **Within-class scatter**: class variances

(Hastie, Tibshirani, and Friedman 2009)
Decision line for circular Gaussians of equal size

The line is the perpendicular bisector of the line separating the means.

(Duda Hart and Stork 2001)
Decision lines for circular vs. elliptical Gaussians of equal size

The line is not! the perpendicular bisector of the line separating the means.

(Duda Hart and Stork 2001)
Mahalanobis distance

The Mahalanobis distance effectively transforms the ellipsoid Gaussians into circular ones. LDA finds the locus of points (Mahalanobis!) equidistant from the group means.

https://stats.stackexchange.com/questions/62092/bottom-to-top-explanation-of-the-mahalanobis-distance
Mahalanobis distance: equipotential curves

https://stats.stackexchange.com/questions/62092/bottom-to-top-explanation-of-the-mahalanobis-distance
Accessing LDA source code in R

1. Download source code (MASS_7.3-30.tar.gz) from CRAN, look for R/ and src/ directories (http://cran.r-project.org/web/packages/MASS/index.html)

   ```r
   library(MASS)
   methods("lda")
   MASS:::lda.default  # ::: access namespace MASS
   detach(package:MASS)  # remove package from search path
   getAnywhere("lda.default")  # don't need to know namespace or have package loaded
   ```
Accessing LDA source code in R

1. Download source code (MASS_7.3-30.tar.gz) from CRAN, look for R/ and src/ directories (http://cran.r-project.org/web/packages/MASS/index.html)

2. Print source in R:

```r
library(MASS)
methods("lda")
MASS:::lda.default  # ::: access namespace MASS
detach(package:MASS)  # remove package from search path
getAnywhere("lda.default")  # don't need to know namespace or have package loaded
```
1. What formant information is most important for discriminating two vowel classes across speakers?
Application of LDA: vowel classification

1. What formant information is most important for discriminating two vowel classes across speakers?

2. Is formant information at some timepoints more important than it is at other timepoints?
Application of LDA: vowel classification

1. What formant information is most important for discriminating two vowel classes across speakers?

2. Is formant information at some timepoints more important than it is at other timepoints?

3. Can we visualize vowels in a lower dimensional space (2D) when we use coarsely sampled formant tracks for our feature set?
vow.dat <- read.table("vowdata.dat", skip = 30, header = FALSE, stringsAsFactors = FALSE, na.strings = 0, col.names = c("filename", "dur.ms", "f0.ss", "f1.ss", "f2.ss", "f3.ss", "f4.ss", "f1.20per", "f2.20per", "f3.20per", "f1.50per", "f2.50per", "f3.50per", "f1.80per", "f2.80per", "f3.80per"))

# Strip information from filenames and code data
vow.dat$speaker <- substr(vow.dat$filename, 1, 3)
vow.dat$sex <- ifelse(substr(vow.dat$filename, 1, 1) == "m", "male", ifelse(substr(vow.dat$filename, 1, 1) == "w", "female", ifelse(substr(vow.dat$filename, 1, 1) == "b", "boy", "girl")))
vow.dat$vowel <- substr(vow.dat$filename, 4, 5)
Setting up factors

```r
# Set up factors
vow.dat$speaker <- factor(vow.dat$speaker)
vow.dat$vowel <- factor(vow.dat$vowel)
vow.dat$sex <- factor(vow.dat$sex, levels = c("girl", "boy", "female", "male"))
```
Visualizing vowel data: code (base)

```r
plot(vow.dat$f1.ss, vow.dat$f2.ss)
text(vow.dat$f1.ss, vow.dat$f2.ss, vow.dat$vowel, cex = 0.7,
pos = 4, col = "red")
```
Visualizing vowel data: plot (base)
Accessing parts of data frames

vow.dat[vow.dat$vowel == "ae", ]  # Only ae rows
vow.dat[, "f2.80per"]  # Only f2.80 column
vow.dat[, names(vow.dat) == "f2.80per"]  # Only f2.80 column
vow.dat$f2.80per  # only f2.80 column
Accessing parts of data frames

vow.dat[vow.dat$vowel == "ae", ]  # Only ae rows
vow.dat[, "f2.80per"]  # Only f2.80 column
vow.dat[, names(vow.dat) == "f2.80per"]  # Only f2.80 column
vow.dat$f2.80per  # only f2.80 column

What does this do?

vow.dat[vow.dat$vowel=="ae", "f2.80per"]
Subsetting data

vow.ae.iy <- subset(vow.dat, vowel == "ae" | vowel == "iy")
vow.ae.iy <- droplevels(subset(vow.dat, vowel == "ae" | vowel == "iy"))
Visualizing vowel data subset: code (base)

```r
plot(vow.dat$f1.ss, vow.dat$f2.ss)
text(vow.dat$f1.ss, vow.dat$f2.ss, vow.dat$vowel, cex = 0.7,
pos = 4, col = "red")
```
Visualizing vowel data subset: plot (base)
library(ggplot2)

plot.m.f1.f2 <- ggplot(subset(vow.ae.iy, sex == "male"),
aes(x = f1.ss, y = f2.ss, group = vowel, color = vowel,
    label = vowel)) + geom_text()

plot.m.f1.f2.annot <- plot.m.f1.f2 + scale_y_continuous(name = "Steady state F2 (Hz)") +
scale_x_continuous(name = "Steady state F1") + theme(legend.position = "none") + theme_bw()

print(plot.m.f1.f2.annot)
### Loading required package: methods

```r
# Loading required package: methods
```

![Vowel data visualization](image)
Calculations on data frames

```r
# Summary statistics
mean(vow.dat$f1.ss)
sd(vow.dat$f1.ss)
summary(vow.dat$f1.ss)  # On a column
summary(vow.dat)       # On whole data frame
# Cross-tabulation
with(vow.dat, xtabs(~sex)/length(sex))  # proportion of each sex level
with(vow.dat, xtabs(~sex + vowel))
with(vow.dat, xtabs(~vowel + sex))
# tapply
tapply(vow.dat$f1.ss, vow.dat$vowel, mean)  # mean f1.ss for each vowel
tapply(vow.dat$f1.ss, list(vow.dat$vowel, vow.dat$sex), mean)  # mean f1.ss for each vowel for each sex
```
LDA example: iris dataset

Straight from the `lda()` help file:

```r
Iris <- data.frame(rbind(iris3[, , 1], iris3[, , 2], iris3[, , 3]), Sp = rep(c("s", "c", "v"), rep(50, 3)))
```

- What are the classes?
LDA example: iris dataset

Straight from the lda() help file:

```r
Iris <- data.frame(rbind(iris3[, , 1], iris3[, , 2], iris3[, , 3]), Sp = rep(c("s", "c", "v"), rep(50, 3)))
```

- What are the classes?
- What is the feature set being used for classification?
LDA example: iris dataset

Partition the dataset into a training set and a test set:

```r
train <- sample(1:150, 75)
table(Iris$Sp[train])
```

##
## c s v
## 22 23 30
LDA example: *iris*—run LDA

```r
library(MASS)
z <- lda(Sp ~ ., Iris, prior = c(1, 1, 1)/3, subset = train)
```
LDA object for iris dataset: priors

Priors: no bias for any of the Iris species

Prior probabilities of groups:

<table>
<thead>
<tr>
<th>c</th>
<th>s</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3333333</td>
<td>0.3333333</td>
<td>0.3333333</td>
</tr>
</tbody>
</table>
Effect of priors on decision boundary: circular Gaussians

$p(x | \omega_i)$

$\omega_1$

$\omega_2$

$p(x | \omega_i)$

$\omega_1$

$\omega_2$

$R_1$

$P(\omega_1) = .7$

$P(\omega_2) = .3$

$R_2$

$R_1$

$P(\omega_1) = .9$

$P(\omega_2) = .1$

$R_2$

$P(\omega_1) = .01$

$P(\omega_2) = .99$
Priors: no bias for any of the Iris species, since prior probabilities equal for all classes

Prior probabilities of groups:

<table>
<thead>
<tr>
<th></th>
<th>c</th>
<th>s</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3333333</td>
<td>0.3333333</td>
<td>0.3333333</td>
</tr>
</tbody>
</table>
Group means: gives us the means of the estimated Gaussians for each of the three iris species classes

Group means:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>6.010000</td>
<td>2.750000</td>
<td>4.286667</td>
<td>1.323333</td>
</tr>
<tr>
<td>s</td>
<td>4.976190</td>
<td>3.357143</td>
<td>1.428571</td>
<td>0.233333</td>
</tr>
<tr>
<td>v</td>
<td>6.733333</td>
<td>3.029167</td>
<td>5.633333</td>
<td>2.116667</td>
</tr>
</tbody>
</table>
LDA object for iris dataset: coefficients of linear discriminants

Coefficients of linear discriminants: gives us the new feature set, where each new feature is a linear combination of the original features. Weight tells us something about importance of each original feature in the discriminant function (decision boundary).

Coefficients of linear discriminants:

<table>
<thead>
<tr>
<th></th>
<th>LD1</th>
<th>LD2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sepal.L.</td>
<td>0.8661028</td>
<td>-0.1529818</td>
</tr>
<tr>
<td>Sepal.W.</td>
<td>1.2761710</td>
<td>-1.8118482</td>
</tr>
<tr>
<td>Petal.L.</td>
<td>-1.6332415</td>
<td>1.4558098</td>
</tr>
<tr>
<td>Petal.W.</td>
<td>-3.9026661</td>
<td>-3.5863053</td>
</tr>
</tbody>
</table>
Proportion of trace: tells us how much variance each new dimension (feature) accounts for—opportunity to reduce dimensionality if we keep only some of the new features.

Proportion of trace:
LD1  LD2
0.9861  0.0139
Sicking the classifier on the test set!

```r
predict(z, Iris[-train, ])$class

## [1] s s s s s s s s s s s s s s s s s s s s s s s s s
## [26] s s c c c c c c c c c c c c v c c c c c c c c c c
## [51] c c c c c v v v v v v v v v v v v v v v v v v v v
## Levels: c s v

lda.iris.test <- predict(z, Iris[-train, ])$class
(z1 <- update(z, . ~ . ~ . - Petal.W.))

## Call:
## lda(Sp ~ Sepal.L. + Sepal.W. + Petal.L., data = Iris, prior = c(1, 1)/3, subset = train)
##
## Prior probabilities of groups:
## c s v
## 0.3333 0.3333 0.3333
```
Machine classification: generalization

Generalization

- **Training**: build classifier based on a set of labeled training examples \( \langle x_m, y_m \rangle \), where \( x_m \in \mathbb{R}^n \) is the feature vector and \( y_m \in C = \{ \text{Class 1, Class 2, \ldots} \} \) is the class label.
Machine classification: generalization

Generalization

- **Training**: build classifier based on a set of labeled training examples \( \langle x_m, y_m \rangle \), where \( x_m \in \mathbb{R}^n \) is the feature vector and \( y_m \in C = \{ \text{Class 1, Class 2, \ldots} \} \) is the class label.

- **Testing**: test classifier performance on novel test data not included in training data set. (Often: withheld data)
Machine classification: generalization

Generalization

- **Training**: build classifier based on a set of labeled training examples \( \langle x_m, y_m \rangle \), where \( x_m \in \mathbb{R}^n \) is the feature vector and \( y_m \in C = \{ \text{Class 1, Class 2, \ldots} \} \) is the class label.

- **Testing**: test classifier performance on novel test data not included in training data set. (Often: withheld data)

- Classifier performance assessed based on generalization (test) error.
Machine classification output

Classification

Given an **test example** $\langle x_m, y^*_m \rangle$, where $x_m \in \mathbb{R}^n$ is the **feature vector** and $y^*_m \in C = \{\text{Class 1, Class 2, \ldots}\}$ is the (actual) **label**, a classifier $K$:
Classification

Given an **test example** $\langle x_m, y_m^* \rangle$, where $x_m \in \mathbb{R}^n$ is the **feature vector** and $y_m^* \in \mathcal{C} = \{\text{Class 1, Class 2, \ldots}\}$ is the (actual) **label**, a classifier $K$:

- Assigns a label $y_m^K$ to $x_m$, or
Machine classification output

Classification

Given an test example $\langle x_m, y^*_m \rangle$, where $x_m \in \mathbb{R}^n$ is the feature vector and $y^*_m \in C = \{\text{Class 1, Class 2, \ldots}\}$ is the (actual) label, a classifier $K$:

- Assigns a label $y^K_m$ to $x_m$, or
- Assigns a probability distribution $\text{Prob}(x_m \text{ in class } c)$ for $c \in C$, with $y^K_m = \arg \max_{c \in C} \text{Prob}(x_m \text{ in class } c)$
Assessing the classifier performance: accuracy

# Classification accuracy

```r
sum(Iris[-train, ]$Sp == lda.iris.test)/length(lda.iris.test) * 100
```

```r
## [1] 98.67
```

## [1] 98.67
Assessing the classifier performance: confusion matrix

```r
confus.data <- data.frame(Iris[-train, ]$Sp, lda.iris.test)
names(confus.data) <- c("Actual", "Predicted")
# compute frequency of actual categories
actual <- as.data.frame(table(confus.data$Actual))
names(actual) <- c("Actual", "ActualFreq")
# build confusion matrix
confusion <- as.data.frame(table(confus.data$Actual, confus.data$Predicted))
names(confusion) <- c("Actual", "Predicted", "Freq")
# calculate percentage of test cases based on actual frequency
confusion <- merge(confusion, actual, by = c("Actual"))
confusion$Percent <- confusion$Freq/confusion$ActualFreq * 100
```
Your turn: LDA on Hillenbrand vowels!

Now you try building an LDA classifier for some of the Hillenbrand vowels!
Classification algorithm example: support vector machines

1. Given labeled training data, e.g.

\[
\langle \langle 200, 210, 224 \rangle, 1 \rangle
\]

Classification algorithm example: support vector machines

1. Given labeled **training data**, e.g. \( \langle 200, 210, 224 \rangle, 1 \rangle 

2. Draw convex hull around data from a given category

---

Classification algorithm example: support vector machines

1. Given labeled training data, e.g., \(<\langle 200, 210, 224 \rangle, 1 >\)

2. Draw convex hull around data from a given category

3. Find separating hyperplane maximizing margin between convex hulls

Classification algorithm example: support vector machines

1. Given labeled **training data**, e.g. 
   \[ \langle \langle 200, 210, 224 \rangle, 1 \rangle \]
2. Draw convex hull around data from a given category
3. Find **separating hyperplane** maximizing margin between convex hulls

\[ x^T \beta + \beta_0 = 0 \]

\[ M = \frac{1}{\| \beta \|} \]

(Hastie, Tibsharani and Friedman 2009)
Classification algorithm example: support vector machines

1. Given labeled **training data**, e.g.
   \[\langle 200, 210, 224 \rangle, 1 \rangle\]
2. Draw convex hull around data from a given category
3. Find **separating hyperplane** maximizing margin between convex hulls
4. Use separating hyperplane to classify **test data** (unseen data): train on 4 speakers, test on 5th, average results

(Hastie, Tibshirani and Friedman 2009)