LINGUIST 592B Week 12: Classification and model comparison

Kristine Yu

Department of Linguistics, UMass Amherst

April 17, 2014
1 Administrative, etc.

2 Practice with training/testing
Notes and reminders: Projects

- Project presentations: Tuesday, 4/29 or finals week?
Notes and reminders: Projects

- Project presentations: Tuesday, 4/29 or finals week?
- Project write-ups: due Friday, 5/2 (end of finals week)
Notes and reminders: Projects

- Project presentations: Tuesday, 4/29 or finals week?
- Project write-ups: due Friday, 5/2 (end of finals week)
- Instructions on project presentations and write-ups
Notes and reminders: Projects

- Project presentations: Tuesday, 4/29 or finals week?
- Project write-ups: due Friday, 5/2 (end of finals week)
- Instructions on project presentations and write-ups
- Seek help!!
- Data set: Hillenbrand vowels
- Classification: LDA
Prepping the vowel data (Hillenbrand’s vowel data)

```r
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"))
```
The research question

What’s your research question? That determines what classes and feature set you choose for the classification problem.

Important first step: explore the data.
Visualizing vowel data: code

```r
library(ggplot2)
plot.m.f1.f2 <- ggplot(subset(vow.dat, sex == "male"), aes(x = f1.ss, y = f2.ss, 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_bw()
print(plot.m.f1.f2.annot)
```
## Loading required package: methods
Subsetting data

You pick 2-3 vowels, and one level of sex.

vow <- droplevels(subset(vow.dat, sex == "male" & (vowel == "ah" | vowel == "aw" | vowel == "ae")))
Visualizing the data: pairs plots

```r
pairs(~f1.20per + f1.80per + f2.20per + f2.80per, data = vow, 
    main = "Scatterplots for vowels", pch = 21, bg = c("red", 
        "green3", "blue")[vow$vowel])
legend(0.1, 0.9, as.vector(levels(vow$vowel)), fill = c("red", 
    "green3", "blue"))
```
Visualizing the data: pairs plots

Scatterplot for vowels

- f1.20per
- f1.80per
- f2.00per
- f2.80per

...
Visualizing data: ellipse plots code

```r
plot(vow[, c("f1.80per", "f2.80per")], pch = 21, bg = c("red", "green3", "blue")[vow$vowel])
legend(900, 2200, as.vector(levels(vow$vowel)), fill = c("red", "green3", "blue"))
library(ellipse)
lines(ellipse(cov(subset(vow, vowel == "ae")[, c("f1.80per", "f2.80per")]), centre = colMeans(subset(vow, vowel == "ae")[, c("f1.80per", "f2.80per")]), level = 0.5), col = "red")
lines(ellipse(cov(subset(vow, vowel == "ah")[, c("f1.80per", "f2.80per")]), centre = colMeans(subset(vow, vowel == "ah")[, c("f1.80per", "f2.80per")]), level = 0.5), col = "green3")
lines(ellipse(cov(subset(vow, vowel == "aw")[, c("f1.80per", "f2.80per")]), centre = colMeans(subset(vow, vowel == "aw")[, c("f1.80per", "f2.80per")]), level = 0.5), col = "blue")
```
Visualizing data: ellipse plots for Gaussian density
What’s your research question? That determines what variables you include in the classification:

```r
library(MASS)
vow.lda <- lda(vowel ~ f1.80per + f2.80per + dur.ms, data = vow)
# vow.lda <- lda(vowel ~ ., data = vow) # oops! what # went wrong?
pred <- predict(vow.lda, vow)
names(pred)
```

```r
## [1] "class"       "posterior" "x"
```
Visualizing LDA: code

```r
vows <- data.frame(vowels = vow$vowel, lda = pred$x)
prop.lda <- round(vow.lda$svd^2/sum(vow.lda$svd^2), 3)
ggplot(vows, aes(lda.LD1, lda.LD2, color = vowels, shape = vowels)) +
  geom_point() + scale_x_continuous(name = paste("LD1 (", prop.lda[1], ", ", sep = "")) +
  scale_y_continuous(name = paste("LD2 (", prop.lda[2], ", ", sep = ""))
```

592B W12  592B Spring 2014 Week 12  16/ 44
Visualizing LDA: plot

![Graph showing LD1 and LD2 dimensions for vowels: ae, ah, aw]
How well are the vowels separated in each dimension?

```r
prop.lda <- round(vow.lda$svd^2/sum(vow.lda$svd^2), 3)
ggplot(vows, aes(x = lda.LD1, y = lda.LD2, shape = vowels, color = vowels)) +
  # set the locations of the x-axis labels as Tukey's five numbers
  scale_x_continuous(limit = c(min(vows$lda.LD1), max(vows$lda.LD1)),
                     breaks = round(fivenum(vows$lda.LD1), 1), name = paste("LD1 (",
                     prop.lda[1], ", "", sep = "")) +
  # ditto for y-axis labels
  scale_y_continuous(limit = c(min(vows$lda.LD2), max(vows$lda.LD2)),
                     breaks = round(fivenum(vows$lda.LD2), 1), name = paste("LD2 (",
                     prop.lda[2], ", "", sep = "")) +
  # specify points
  geom_point() +
  # specify that we want the rug plot
  geom_rug(size = 0.5) +
  # improve the data/ink ratio
  theme_set(theme_minimal(base_size = 12))
```
How well are the vowels separated in each dimension?

![Graph showing separation of vowels in two dimensions](image-url)
source("multiplot.R")  # Run code to define function
p1 <- ggplot(vows, aes(x = lda.LD1, color = vowels)) + geom_density()
ggtitle("Posterior densities for LDA 1")

p2 <- ggplot(vows, aes(x = lda.LD2, color = vowels)) + geom_density()
ggtitle("Posterior densities for LDA 2")

multiplot(p1, p2)
## Loading required package: grid

```
# Load required package
library(grid)
```

---

### Conditional density plots

#### Posterior densities for LDA 1

- **lda.LD1**
- **Density**
- **Vowels**: ae, ah, aw

#### Posterior densities for LDA 2

- **lda.LD2**
- **Density**
- **Vowels**: ae, ah, aw
Decision boundaries in 2-D LDA space

Remember LDA “spheres” the variables, so we can use an equal-scaled plot. (Below: based on Venables and Ripley (2002, p. 332-3)

```r
eqscplot(pred$x, type = "n")  # equal scale plot
text(pred$x, labels = as.character(pred$class), col = c("red", "green3", "blue")[vow$vowel])
# Now classify in this 2-D LDA space, plot means as crosses
vow.lda.2 <- lda(pred$x, vow$vowel)
means.vow <- vow.lda.2$means
points(means.vow[, -3], pch = 3, cex = 1.5, lwd = 2)
```
Decision boundaries in 2-D LDA space

# Plot decision boundaries

```r
perp <- function(x, y) {
  m <- (x + y)/2
  abline(c(m[2] - s * m[1], s))
  invisible()
}
perp(means.vow[1, ], means.vow[2, ]) # ae/ah
perp(means.vow[2, ], means.vow[3, ]) # ah/aw
perp(means.vow[1, ], means.vow[3, ]) # ae/aw
```
Decision boundaries in 2-D LDA space
To interpret relative coefficient weights, we need to standardize the data.

```r
vow$f1.80per.z <- (vow$f1.80per - mean(vow$f1.80per))/sd(vow$f1.80per)
z.score <- function(x) {
  as.numeric(scale(x, scale = TRUE))
}  # calculate z-score
vow$f1.80per.z2 <- z.score(vow$f1.80per)
```
Example: Khouw and Ciocca 2007

- What classes? What feature set?
Example: Khouw and Ciocca 2007

- What classes? What feature set?
- How many linear discriminants? How much proportion of variance accounted for?
Example: Khouw and Ciocca 2007

- What classes? What feature set?
- How many linear discriminants? How much proportion of variance accounted for?
- What coefficient weights? What group means?
Cowabunga! But wait a minute—we didn’t check for generalization: we didn’t withhold any test data.

```r
print(tt <- table(vow$vowel, pred$class))

##
## ae ah aw
## ae 43 2 0
## ah  4 35 6
## aw  1  6 38

print(error <- sum(tt[row(tt) != col(tt)])/sum(tt))

## [1] 0.1407
```
Generating latex tables

Try packages `xtable`, Hmisc’s `latex()`.

```r
library(xtable)
xtable(table(pred$class, vow$vowel))
```

```latex
\begin{table}[ht]
\centering
\begin{tabular}{rrrr}
\hline
 & ae & ah & aw \\ 
\hline
ae & 43 & 4 & 1 \\ 
ah & 2 & 35 & 6 \\ 
av & 0 & 6 & 38 \\ 
\hline
\end{tabular}
\end{table}
```
Remember **holding out** data?
Partition the dataset into a training set and a test set:

```r
train <- sample(1:nrow(Iris), nrow(Iris)/2)
table(Iris$Sp[train])
```
Problems with holding out data

- If we have very little data, we might not be able to have a sufficiently large training set if we hold out data.
Problems with holding out data

- If we have very little data, we might not be able to have a sufficiently large training set if we hold out data.
- We only do one train/test split, so our estimation of the error rate is highly dependent on the particular split that was randomly chosen.
Problems with holding out data

- If we have very little data, we might not be able to have a sufficiently large training set if we hold out data.
- We only do one train/test split, so our estimation of the error rate is highly dependent on the particular split that was randomly chosen.

Idea: resample instead!
Resampling methods

- Random subsampling
Resampling methods

- Random subsampling
- $\nu$-fold cross-validation
Resampling methods

- Random subsampling
- \(\nu\)-fold cross-validation
- Leave-one-out cross-validation
Resampling methods

- Random subsampling
- $\nu$-fold cross-validation
- Leave-one-out cross-validation

Idea: estimate error rate as average error rate over all experiments.
Split data $K$ times where each split randomly selects some fixed number $n$ of test examples to be held out.
$\nu/K$-fold cross-validation

Partition data into $K - 1$ folds; for each fold $f$, train up classifier on all other folds and then test on fold $f$. 

v-fold cross-validation

What proportion of data is held out for test data as a function of v?

```r
vlda <- function(v, formula, data, cl) {
  require(MASS)
  grps <- cut(1:nrow(data), v, labels = FALSE)[sample(1:nrow(data))]
  pred <- lapply(1:v, function(i, formula, data) {
    omit <- which(grps == i)
    z <- lda(formula, data = data[-omit, ])
    predict(z, data[omit, ])
  }, formula, data)
  wh <- unlist(lapply(pred, function(pp) pp$class))
  table(wh, cl[order(grps)])
}

http://www.stat.berkeley.edu/classes/s133/Class2a.html
Test error

```r
print(tt <- vlda(5, vowel ~ f1.80per + f2.80per + dur.ms, 
    vow, vow$vowel))

##
## wh  ae  ah  aw
## ae  43  5  1
## ah  2  34  7
## aw  0   6  37

print(error <- sum(tt[row(tt) != col(tt)])/sum(tt))

## [1] 0.1556
```
Leave-one-out cross-validation

For each example $y_i$, train on all $y_j$ where $i \neq j$, test on $y_i$ (special case of $\nu$-fold cross-validation):

vow.cv.lda <- lda(vowel ~ f1.80per + f2.80per + dur.ms, 
  data = vow, CV = TRUE)

print(cv.tab <- table(vow$vowel, vow.cv.lda$class))

##
## ae ah aw
## ae 43  2  0
## ah  4 35  6
## aw  1  7 37

print(per.correct <- sum(diag(cv.tab))/sum(cv.tab))

# [1] 0.8519
Visualizing cross-validation performance

```r
plot(vow[, c("f1.80per", "f2.80per")], col = as.factor(vow$vowel),
pch = as.numeric(vow.cv.lda$class))
```
Visualizing cross-validation performance
Feature selection via model comparison

<table>
<thead>
<tr>
<th>Feature removed</th>
<th>Δ MPCorr</th>
<th>Δ Acc (%)</th>
<th>Δ MeanF</th>
</tr>
</thead>
<tbody>
<tr>
<td>all 21 features</td>
<td>0.3823</td>
<td>54.64</td>
<td>0.4222</td>
</tr>
<tr>
<td>f0 stdv</td>
<td>0.0104</td>
<td>1.1986</td>
<td>0.0108</td>
</tr>
<tr>
<td>D(f0) 3:5</td>
<td>0.0023</td>
<td>0.1226</td>
<td>0.0010</td>
</tr>
<tr>
<td>f0 range</td>
<td>0.0016</td>
<td>0.2574</td>
<td>0.0029</td>
</tr>
<tr>
<td>f0 grad54</td>
<td>0.0008</td>
<td>0.0711</td>
<td>0.0013</td>
</tr>
<tr>
<td>D(f0) 4:5</td>
<td>0.0007</td>
<td>-0.0074</td>
<td>-0.0001</td>
</tr>
<tr>
<td>f0 median</td>
<td>0.0006</td>
<td>0.0735</td>
<td>0.0007</td>
</tr>
<tr>
<td>D(f0) 2:5</td>
<td>0.0005</td>
<td>0.0221</td>
<td>0.0002</td>
</tr>
<tr>
<td>f0 min</td>
<td>0.0005</td>
<td>0.0711</td>
<td>0.0006</td>
</tr>
<tr>
<td>f0 mean</td>
<td>0.0003</td>
<td>-0.0123</td>
<td>-0.0003</td>
</tr>
<tr>
<td>f0 grad12</td>
<td>0.0002</td>
<td>0.0123</td>
<td>0.0005</td>
</tr>
<tr>
<td>f0 4:6</td>
<td>0.0002</td>
<td>-0.0245</td>
<td>-0.0003</td>
</tr>
<tr>
<td>f0 max</td>
<td>0.0002</td>
<td>-0.0074</td>
<td>0.0003</td>
</tr>
</tbody>
</table>

Figure: Surendran (2007): features for automatic recognition of Mandarin tones
Feature selection via model comparison: performance metrics

- **Accuracy**: percentage of test examples correctly classified

---

1. The harmonic mean of two numbers is never higher than the geometrical mean. It also tends towards the least number, minimizing the impact of large outliers and maximizing the impact of small ones. The F-measure therefore tends to privilege balanced systems. (http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean)
Feature selection via model comparison: performance metrics

- **Accuracy**: percentage of test examples correctly classified
- **MPCorr**: average probability of the correct label being predicted over test examples

---

1 The harmonic mean of two numbers is never higher than the geometrical mean. It also tends towards the least number, minimizing the impact of large outliers and maximizing the impact of small ones. The F-measure therefore tends to privilege balanced systems. (http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean)
Feature selection via model comparison: performance metrics

- **Accuracy**: percentage of test examples correctly classified
- **MPCorr**: average probability of the correct label being predicted over test examples
- **MeanF**: average of per-class F score, the harmonic mean\(^1\) of precision and recall

---

\(^1\) The harmonic mean of two numbers is never higher than the geometrical mean. It also tends towards the least number, minimizing the impact of large outliers and maximizing the impact of small ones. The F-measure therefore tends to privilege balanced systems. (http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean)
Feature selection via model comparison: performance metrics

- **Accuracy**: percentage of test examples correctly classified
- **MPCorr**: average probability of the correct label being predicted over test examples
- **MeanF**: average of per-class F score, the harmonic mean\(^1\) of precision and recall

### F-measure and precision and recall

- **Precision**: \( P = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \)

\(^1\) The harmonic mean of two numbers is never higher than the geometrical mean. It also tends towards the least number, minimizing the impact of large outliers and maximizing the impact of small ones. The F-measure therefore tends to privilege balanced systems. (http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean)
Feature selection via model comparison: performance metrics

- **Accuracy**: percentage of test examples correctly classified
- **MPCorr**: average probability of the correct label being predicted over test examples
- **MeanF**: average of per-class F score, the harmonic mean\(^1\) of precision and recall

### F-measure and precision and recall

- **Precision**: \[ P = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \]
- **Recall**: \[ R = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \]

\(^1\) The harmonic mean of two numbers is never higher than the geometrical mean. It also tends towards the least number, minimizing the impact of large outliers and maximizing the impact of small ones. The F-measure therefore tends to privilege balanced systems. (http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean)
Feature selection via model comparison: performance metrics

- **Accuracy**: percentage of test examples correctly classified
- **MPCorr**: average probability of the correct label being predicted over test examples
- **MeanF**: average of per-class F score, the harmonic mean\(^1\) of precision and recall

---

**F-measure and precision and recall**

- **Precision**: \[ P = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \]
- **Recall**: \[ R = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \]
- **F-measure**: \[ \frac{1}{F} = \frac{1}{2} \left( \frac{1}{R} + \frac{1}{P} \right) \]

\(^1\)The harmonic mean of two numbers is never higher than the geometrical mean. It also tends towards the least number, minimizing the impact of large outliers and maximizing the impact of small ones. The F-measure therefore tends to privilege balanced systems. (http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean)
Precision and recall: red fishes as targets

**Precision**
$$P = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

**Recall**
$$R = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

How high is precision? Recall?

Precision and recall: red fishes as targets

**Precision and recall**

- **Precision:** \( P = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \)
- **Recall:** \( R = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \)


1. How high is precision? Recall?
2. How high is precision? Recall?
Feature selection via model comparison: performance metrics

- **Accuracy**: percentage of test examples correctly classified

---

1 The harmonic mean of two numbers is never higher than the geometrical mean. It also tends towards the least number, minimizing the impact of large outliers and maximizing the impact of small ones. The F-measure therefore tends to privilege balanced systems. (http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean)
Feature selection via model comparison: performance metrics

- **Accuracy**: percentage of test examples correctly classified
- **MPCorr**: average probability of the correct label being predicted over test examples

---

1. The harmonic mean of two numbers is never higher than the geometrical mean. It also tends towards the least number, minimizing the impact of large outliers and maximizing the impact of small ones. The F-measure therefore tends to privilege balanced systems. (http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean)
Feature selection via model comparison: performance metrics

- **Accuracy**: percentage of test examples correctly classified
- **MPCorr**: average probability of the correct label being predicted over test examples
- **MeanF**: average of per-class F score, the harmonic mean\(^1\) of precision and recall

---

\(^1\)The harmonic mean of two numbers is never higher than the geometrical mean. It also tends towards the least number, minimizing the impact of large outliers and maximizing the impact of small ones. The F-measure therefore tends to privilege balanced systems. (http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean)
Feature selection via model comparison: performance metrics

- **Accuracy**: percentage of test examples correctly classified
- **MPCorr**: average probability of the correct label being predicted over test examples
- **MeanF**: average of per-class F score, the harmonic mean¹ of precision and recall

**F-measure and precision and recall**

- **Precision**: \( P = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \)

¹The harmonic mean of two numbers is never higher than the geometrical mean. It also tends towards the least number, minimizing the impact of large outliers and maximizing the impact of small ones. The F-measure therefore tends to privilege balanced systems. (http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean)
Feature selection via model comparison: performance metrics

- **Accuracy**: percentage of test examples correctly classified
- **MPCorr**: average probability of the correct label being predicted over test examples
- **MeanF**: average of per-class F score, the harmonic mean\(^1\) of precision and recall

### F-measure and precision and recall

- **Precision**: \( P = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \)
- **Recall**: \( R = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \)

---

\(^1\)The harmonic mean of two numbers is never higher than the geometrical mean. It also tends towards the least number, minimizing the impact of large outliers and maximizing the impact of small ones. The F-measure therefore tends to privilege balanced systems. ([http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean](http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean))
Feature selection via model comparison: performance metrics

- **Accuracy**: percentage of test examples correctly classified
- **MPCorr**: average probability of the correct label being predicted over test examples
- **MeanF**: average of per-class F score, the harmonic mean of precision and recall

**F-measure and precision and recall**

- **Precision**: \( P = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \)
- **Recall**: \( R = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \)
- **F-measure**: \( \frac{1}{F} = \frac{1}{2} \left( \frac{1}{R} + \frac{1}{P} \right) \)

1 The harmonic mean of two numbers is never higher than the geometrical mean. It also tends towards the least number, minimizing the impact of large outliers and maximizing the impact of small ones. The F-measure therefore tends to privilege balanced systems. (http://stats.stackexchange.com/questions/37590/in-calculating-the-f-measure-with-precision-and-recall-why-is-the-harmonic-mean)
### Feature selection via model comparison

<table>
<thead>
<tr>
<th>Feature removed</th>
<th>Δ MPcorr</th>
<th>Δ Acc (%)</th>
<th>Δ MeanF</th>
</tr>
</thead>
<tbody>
<tr>
<td>all 21 features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f0 stdv</td>
<td>0.0104</td>
<td>1.1986</td>
<td>0.0108</td>
</tr>
<tr>
<td>D(f0) 3:5</td>
<td>0.0023</td>
<td>0.1226</td>
<td>0.0010</td>
</tr>
<tr>
<td>f0 range</td>
<td>0.0016</td>
<td>0.2574</td>
<td>0.0029</td>
</tr>
<tr>
<td>f0 grad54</td>
<td>0.0008</td>
<td>0.0711</td>
<td>0.0013</td>
</tr>
<tr>
<td>D(f0) 4:5</td>
<td>0.0007</td>
<td>-0.0074</td>
<td>-0.0001</td>
</tr>
<tr>
<td>f0 median</td>
<td>0.0006</td>
<td>0.0735</td>
<td>0.0007</td>
</tr>
<tr>
<td>D(f0) 2:5</td>
<td>0.0005</td>
<td>0.0221</td>
<td>0.0002</td>
</tr>
<tr>
<td>f0 min</td>
<td>0.0005</td>
<td>0.0711</td>
<td>0.0006</td>
</tr>
<tr>
<td>f0 mean</td>
<td>0.0003</td>
<td>-0.0123</td>
<td>-0.0003</td>
</tr>
<tr>
<td>f0 grad12</td>
<td>0.0002</td>
<td>0.0123</td>
<td>0.0005</td>
</tr>
<tr>
<td>f0 4:6</td>
<td>0.0002</td>
<td>-0.0245</td>
<td>-0.0003</td>
</tr>
<tr>
<td>f0 max</td>
<td>0.0002</td>
<td>-0.0074</td>
<td>0.0003</td>
</tr>
</tbody>
</table>

**Figure**: Surendran (2007): features for automatic recognition of Mandarin tones