LINGUIST 592B Week 11: Classification and model comparison

Kristine Yu

Department of Linguistics, UMass Amherst

April 10, 2014
1. Administrative, etc.

2. Classification and generalization

3. Practice with training/testing
Notes and reminders: Projects

Literature report due on Thursday

- Algorithm: use paper to help you write up a note on how the algorithm works
  - What parameters need to be extracted from the signal? Why these parameters?
  - Any particular difficulties and/or special tricks for parameter extraction
  - If presented in paper, applications of algorithm, e.g. why would we want to measure harmonics-to-noise ratio? What can that do for us?
Notes and reminders: Projects

Literature report due on Thursday

- Paper focused on application: use paper to help you understand how speech processing was used to answer the research question at hand
  - What’s the research question and hypotheses?
  - What computational methods/speech processing methods were used and why?
  - How were the results of the methods analyzed, e.g. model comparison, deciding which features are important, and how were these used to support the conclusions of the study (answering the research question?)
Classification algorithm example: support vector machines

1. Given labeled training data, e.g. 
   \[ \langle 200, 210, 224, 7 \rangle \]

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

2. Draw convex hull around data from a given category

---

**Classification algorithm example: support vector machines**

Classification algorithm example: support vector machines

1. Given labeled **training data**, e.g. \[ \langle 200, 210, 224 \rangle, \widehat{1} \rangle 
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. 
\[ \langle 200, 210, 224, 7 \rangle \]

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, Tibshirani and Friedman 2009)
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
4. Use separating hyperplane to classify **test data** (unseen data): train on 4 speakers, test on 5th, average results
A difference between LDA and SVM approaches:

- LDA: finds an approximation to density functions of classes (remember: assumption of Gaussian densities with equal covariances)
- SVM: just looks for optimal separating hyperplane as decision boundary
Interlude: Baye’s theorem

- Intuition behind chain rule (Venn diagrams + population)
Interlude: Baye’s theorem

- Intuition behind chain rule (Venn diagrams + population)
- Derivation of Baye’s theorem
Interlude: Baye’s theorem

- Intuition behind chain rule (Venn diagrams + population)
- Derivation of Baye’s theorem
- Derivation of log posterior (exercise!)
Density functions and classification

Why do we care about ratios between density?

\[ p(x|\omega_i) \]

\[ \omega_1 \]
\[ \omega_2 \]

\[ P(\omega_1) = 0.7 \quad P(\omega_2) = 0.3 \]

\[ R_1 \quad R_2 \]

\[ P(\omega_1) = 0.9 \quad P(\omega_2) = 0.1 \]

\[ P(\omega_1) = 0.8 \quad P(\omega_2) = 0.2 \]

\[ P(\omega_1) = 0.99 \quad P(\omega_2) = 0.01 \]

\[ R_1 \quad R_2 \]
Interlude: Baye’s theorem

- Intuition behind chain rule (Venn diagrams + population)
Interlude: Baye’s theorem

- Intuition behind chain rule (Venn diagrams + population)
- Derivation of Baye’s theorem
Interlude: Baye’s theorem

- Intuition behind chain rule (Venn diagrams + population)
- Derivation of Baye’s theorem
- Derivation of log posterior (exercise!)
Let $f_k(x)$ be the class-conditional density of $X$ in class $G = k$ and $\pi_k$ be the prior probability of class $k$, with priors summing to 1. Then the posterior density can be found as:

$$\Pr(G = k \mid X = x) = \frac{f_k(x) \pi_k}{\sum_{\ell=1}^{K} f_\ell(x) \pi_\ell}.$$
Maximum a Posteriori Classification

How do we determine what the classification decision for a data point is?

In **maximum a posteriori (MAP) classification**, we choose the class with the **highest posterior density** for the data.
Maximum a Posteriori Classification

How do we determine what the classification decision for a data point is?

In **maximum a posteriori (MAP) classification**, we choose the class with the **highest posterior density** for the data.

- Compute the **log odds** ratio, the log-transformed ratio of posterior class densities.
Maximum a Posteriori Classification

How do we determine what the classification decision for a data point is?

In **maximum a posteriori (MAP) classification**, we choose the class with the **highest posterior density** for the data.

- Compute the **log odds** ratio, the log-transformed ratio of posterior class densities
- Taking the ratio allows us to forego the calculation of the normalization term
Maximum a Posteriori Classification

How do we determine what the classification decision for a data point is?

In **maximum a posteriori (MAP) classification**, we choose the class with the **highest posterior density** for the data.

- Compute the **log odds** ratio, the log-transformed ratio of posterior class densities
- Taking the ratio allows us to forego the calculation of the normalization term
- Taking the log transform allows us to turn multiplication into addition
Maximum a Posteriori Classification

How do we determine what the classification decision for a data point is?

In **maximum a posteriori (MAP) classification**, we choose the class with the **highest posterior density** for the data.

- Compute the **log odds** ratio, the log-transformed ratio of posterior class densities
- Taking the ratio allows us to forego the calculation of the normalization term
- Taking the log transform allows us to turn multiplication into addition
- Note: log is often assumed to mean log$_e$, i.e. ln
The Gaussian assumption

Model each class density as a **multivariate normal (Gaussian)**, e.g. in 1-D, with mean $\mu_k$ and variance $\sigma^2$ for class $k$:

$$f_k(x) = \frac{1}{\sqrt{2\pi \sigma^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma^2}}$$  \hspace{1cm} (1)

Take **log-odds ratio**. If $\mu_k = \mu_l = \mu$ (equal covariances):

$$\log \frac{Pr(G = k|X = x)}{Pr(G = l|X = x)}$$  \hspace{1cm} (2)

$$= \log \frac{f_k(x)}{f_l(x)} + \log \frac{\pi_k}{\pi_l}$$  \hspace{1cm} (3)

$$= \log \frac{\pi_k}{\pi_l} + \log \frac{e^{-\frac{(x-\mu_k)^2}{2\sigma^2}}}{e^{-\frac{(x-\mu_l)^2}{2\sigma^2}}}$$  \hspace{1cm} (4)
Computing the log odds ratio

\[
\log \frac{Pr(G = k|X = x)}{Pr(G = l|X = x)} = \log \frac{f_k(x)}{f_l(x)} + \log \frac{\pi_k}{\pi_l}
\]

(5)

(6)

(7)

(8)

(9)

(10)
Linear discriminant analysis

- The assumption of equal covariances eliminates the term that is quadratic in $x$
Linear discriminant analysis

- The assumption of equal covariances eliminates the term that is quadratic in $x$
- This is why the computed decision boundaries are linear in LDA
Linear discriminant analysis

- The assumption of equal covariances eliminates the term that is quadratic in $x$
- This is why the computed decision boundaries are linear in LDA
- The general case where covariance matrices are arbitrary results in a quadratic decision boundary
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.
library(ggplot2)
plot.m.f1.f2 <- ggplot(subset(vow.dat, 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")
print(plot.m.f1.f2.annot)
## Loading required package: methods
Subsetting data

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

```r
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 = "Scatterplot for vowels")
```
Visualizing the data: pairs plots

Scatterplot for vowels

f1.20per

f1.80per

f2.20per

f2.80per
Deciding on the feature set and classes

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)
```
Visualizing LDA: plot

- LD1 (0.912)
- LD2 (0.088)
- Vowels: ae, ah, aw
Cowabunga!

```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 \texttt{latex()}.

\begin{verbatim}
library(xtable)
xtable(table(pred$\texttt{class}, vow$\texttt{vowel}))
\end{verbatim}

\begin{verbatim}
\# \texttt{% latex\ table\ generated\ in\ R\ 3.0.2\ by\ xtable\ 1.7-1\ package}
\# \texttt{% Thu\ Apr\ 10\ 09:21:40\ 2014}
\# \texttt{\begin{verbatim}
begin\{table\}[ht]\n\centering\n\begin{tabular}{rrrr}
\hline
& ae & ah & aw \\
\hline
ae & 43 & 4 & 1 \\
ah & 2 & 35 & 6 \\
aw & 0 & 6 & 38 \\
\hline
\end{tabular}
\end{verbatim}\}
\end{table}
\end{verbatim}
\end{verbatim}
LDA example: iris dataset

Remember this?
Partition the dataset into a training set and a test set:

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

Let's automate this!
$\nu$-fold cross-validation

What proportion of data is held out for test data as a function of $\nu$?

```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

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

##
## wh ae ah aw
## ae 43 4 1
## ah 2 34 8
## aw 0 7 36

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

## [1] 0.163
Leave-one-out cross-validation

For each example $y_i$, train on all $y_j$ where $i \neq j$, test on $y_i$:

```r
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))
```

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

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

```r
## [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](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**

Precision: 
\[ P = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \]

**Recall**

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 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](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)
## 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