Linear Methods for Classification

Based on Chapter 4 of Hastie, Tibshirani, and Friedman

David Madigan
Predictive Modeling

Goal: learn a mapping: \( y = f(x; \theta) \)

Need:  
1. A model structure  
2. A score function  
3. An optimization strategy

Categorical \( y \in \{ c_1, \ldots, c_m \} \): classification

Real-valued \( y \): regression

Note: usually assume \( \{ c_1, \ldots, c_m \} \) are mutually exclusive and exhaustive
Probabilistic Classification

Let $p(c_k) = \text{prob. that a randomly chosen object comes from } c_k$

Objects from $c_k$ have: $p(x \mid c_k, \theta_k)$ (e.g., MVN)

Then:

$$p(c_k \mid x) \propto p(x \mid c_k, \theta_k) p(c_k)$$

**Bayes Error Rate**: 

$$p_b^* = \int (1 - \max_k p(c_k \mid x)) p(x) dx$$

• Lower bound on the best possible error rate
Bayes error rate about 6%
Classifier Types

**Discriminative**: model $p(c_k | \mathbf{x})$
- e.g. logistic regression, CART

**Generative**: model $p(\mathbf{x} | c_k, \theta_k)$
- e.g. “Bayesian classifiers”, LDA
Regression for Binary Classification

• Can fit a linear regression model to a 0/1 response

• Predicted values are not necessarily between zero and one

• With p>1, the decision boundary is linear
  e.g. \(0.5 = b_0 + b_1 x_1 + b_2 x_2\)
Linear Discriminant Analysis

$K$ classes, $X \ n \times \ p \ data \ matrix.$

$$p(c_k \mid \mathbf{x}) \propto p(\mathbf{x} \mid c_k, \theta_k) \ p(c_k)$$

Could model each class density as multivariate normal:

$$p(\mathbf{x} \mid c_k) = \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1} (\mathbf{x} - \mu_k)}$$

LDA assumes $\Sigma_k \equiv \Sigma$ for all $k$. Then:

$$\log \frac{p(c_k \mid x)}{p(c_l \mid x)} = \log \frac{p(c_k)}{p(c_l)} \ - \frac{1}{2} (\mu_k + \mu_l)^T \Sigma^{-1} (\mu_k - \mu_l) + x^T \Sigma^{-1} (\mu_k - \mu_l)$$

This is linear in $x$. 
Linear Discriminant Analysis (cont.)

It follows that the classifier should predict: \( \arg \max_k \delta_k(x) \)

\[
\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log p(c_k)
\]

“linear discriminant function”

If we don’t assume the \( \Sigma_k \)'s are identical, get Quadratic DA:

\[
\delta_k(x) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + \log p(c_k)
\]
Linear Discriminant Analysis (cont.)

Can estimate the LDA parameters via maximum likelihood:

\[ \hat{p}(c_k) = \frac{N_k}{N} \]

\[ \hat{\mu}_k = \frac{\sum_{i \in k} x_i}{N_k} \]

\[ \hat{\Sigma} = \sum_{k=1}^{K} \sum_{i \in k} (x_i - \mu_k)(x_i - \mu_k)' / (N - K) \]
Figure 4.5: The left panel shows three Gaussian distributions, with the same covariance and different means. Included are the contours of constant density enclosing 95% of the probability in each case. The Bayes decision boundaries between each pair of classes are shown (broken straight lines), and the Bayes decision boundaries separating all three classes are the thicker solid lines (a subset of the former). On the right we see a sample of 30 drawn from each Gaussian distribution, and the fitted LDA decision boundaries.
LDA

QDA
Figure 4.6: Two methods for fitting quadratic boundaries. The left plot shows the quadratic decision boundaries for the data in Figure 4.1 (obtained using LDA in the five-dimensional space $x_1, x_2, x_{12}, x_1^2, x_2^2$). The right plot shows the quadratic decision boundaries found by QDA. The differences are small, as is usually the case.
LDA (cont.)

• Fisher is optimal if the class are MVN with a common covariance matrix

• Computational complexity $O(mp^2n)$
Logistic Regression

Note that LDA is linear in x:

\[
\log \frac{p(c_k | x)}{p(c_0 | x)} = \log \frac{p(c_k)}{p(c_0)} - \frac{1}{2} (\mu_k + \mu_0)^T \Sigma^{-1} (\mu_k - \mu_0) + x^T \Sigma^{-1} (\mu_k - \mu_0)
\]

\[= \alpha_k^0 + \alpha_k^T x\]

Linear logistic regression looks the same:

\[
\log \frac{p(c_k | x)}{p(c_0 | x)} = \beta_k^0 + \beta_k^T x
\]

But the estimation procedure for the co-efficients is different. LDA maximizes joint likelihood \([y, X]\); logistic regression maximizes conditional likelihood \([y | X]\). Usually similar predictions.
Logistic Regression MLE

For the two-class case, the likelihood is:

\[
l(\beta) = \sum_{i=1}^{n} \left\{ y_i \log p(x_i; \beta) + (1 - y_i) \log(1 - p(x_i; \beta)) \right\}
\]

\[
\log \left( \frac{p(x; \beta)}{1 - p(x; \beta)} \right) = \beta^T x \quad \log p(x; \beta) = \beta^T x - \log(1 + \exp(\beta^T x))
\]

\[
\Rightarrow l(\beta) = \sum_{i=1}^{n} \left\{ y_i \beta^T x + \log(1 + \exp(\beta^T x)) \right\}
\]

The maximize need to solve (non-linear) score equations:

\[
\frac{dl(\beta)}{d\beta} = \sum_{i=1}^{n} x_i (y_i - p(x_i; \beta)) = 0
\]
### Logistic Regression Modeling

South African Heart Disease Example (y=MI)

<table>
<thead>
<tr>
<th></th>
<th>Coef.</th>
<th>S.E.</th>
<th>Z score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>-4.130</td>
<td>0.964</td>
<td>-4.285</td>
</tr>
<tr>
<td>sbp</td>
<td>0.006</td>
<td>0.006</td>
<td>1.023</td>
</tr>
<tr>
<td>Tobacco</td>
<td>0.080</td>
<td>0.026</td>
<td>3.034</td>
</tr>
<tr>
<td>ldl</td>
<td>0.185</td>
<td>0.057</td>
<td>3.219</td>
</tr>
<tr>
<td>Famhist</td>
<td>0.939</td>
<td>0.225</td>
<td>4.178</td>
</tr>
<tr>
<td>Obesity</td>
<td>-0.035</td>
<td>0.029</td>
<td>-1.187</td>
</tr>
<tr>
<td>Alcohol</td>
<td>0.001</td>
<td>0.004</td>
<td>0.136</td>
</tr>
<tr>
<td>Age</td>
<td>0.043</td>
<td>0.010</td>
<td>4.184</td>
</tr>
</tbody>
</table>
likelihood

score test

LRT

Wald

$\beta_0$

mle

$\beta$
Data that has a sharp survival cut off point between patients who live or die should have a large value of $\beta$.

Data with a lengthy transition from survival to death should have a low value of $\beta$. 
Regularized Logistic Regression

- Ridge/LASSO logistic regression

\[ \hat{w} = \arg \inf_w \frac{1}{n} \sum_{i=1}^{n} \ln(1 + \exp(-w^T x_i y_i)) + \lambda w^2. \]

\[ \hat{w} = \arg \inf_w \frac{1}{n} \sum_{i=1}^{N} \log(1 + \exp(-w^T x_i y_i)) + \lambda \sum_{j} |w_j| \]

- Successful implementation with over 100,000 predictor variables
- Can also regularize discriminant analysis
Simple Two-Class Perceptron

Define: \[ h(x) = \sum_{j=1}^{p} w_j x_j, \quad 1 \leq j \leq p \]

Classify as class 1 if \( h(x) > 0 \), class 2 otherwise

Score function: \# misclassification errors on training data

For training, replace class 2 \( x_j \)'s by \(-x_j\); now need \( h(x) > 0 \)

Initialize weight vector

Repeat one or more times:

For each training data point \( x_i \):

If point correctly classified, do nothing

Else \( w \leftarrow w + \lambda x_i \)

Guaranteed to converge to a separating hyperplane (if exists)
Figure 4.13: A toy example with two classes separable by a hyperplane. The orange line is the least squares solution, which misclassifies one of the training points. Also shown are two blue separating hyperplanes found by the perceptron learning algorithm with different random starts.
“Optimal” Hyperplane

The “optimal” hyperplane separates the two classes and maximizes the distance to the closest point from either class.

Finding this hyperplane is a convex optimization problem.

This notion plays an important role in support vector machines.
Figure 4.15: The same data as in Figure 4.13. The shaded region delineates the maximum margin separating the two classes. There are three support points indicated, which lie on the boundary of the margin, and the optimal separating hyperplane (blue line) bisects the slab. Included in the figure is the boundary found using logistic regression (red line), which is very close to the optimal separating hyperplane (see Section 12.3.3).
\[ w \cdot x + b = 0 \]

\[ \frac{|1-b|}{\|w\|} \text{ from (0,0)} \]

\[ \frac{|-1-b|}{\|w\|} \text{ from (0,0)} \]

Margin

Origin
Bad Things Can Happen…

DATA

happiness

hours per week on studies
simple regression line

HAP = \beta_0 + \beta_1 \times \text{HOURS}, \ \beta_1 \text{ will be estimated to be negative}
A 2nd Look at the DATA

happiness

hours per week on studies
A 2\textsuperscript{nd} Look at the DATA

happiness

hours per week on studies
Other Odd Things Can Happen…
Other Odd Things Can Happen…
### Table

<table>
<thead>
<tr>
<th></th>
<th>Nausea</th>
<th>No Nausea</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rosinex</td>
<td>81</td>
<td>9</td>
</tr>
<tr>
<td>Rosinex</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>No Rosinex</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>No Rosinex</td>
<td>90</td>
<td>810</td>
</tr>
</tbody>
</table>

### R Code

```r
Call:
glm(formula = N ~ G, family = binomial(link = logit), data = d)

Deviance Residuals:
     Min       1Q   Median       3Q      Max
-1.8519  -0.4800  -0.4800  -0.4800   2.1063

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.1031   0.1065  -19.76  <2e-16 ***
G           3.6195   0.2812   12.87  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 949.78  on 1009  degrees of freedom
Residual deviance: 720.32  on 1008  degrees of freedom
AIC: 724.32
```