Statistics 202: Data Mining

Linear Discriminant Analysis

Based in part on slides from textbook, slides of Susan Holmes

©Jonathan Taylor

November 9, 2012

Discriminant analysis

Nearest centroid rule

- Suppose we break down our data matrix as by the labels yielding $(\boldsymbol{X}^j)_{1 \leq j \leq k}$ with sizes $\texttt{nrow}(\boldsymbol{X}^j) = n_j$.
- A simple rule for classification is:

  Assign a new observation with features $\boldsymbol{x}$ *to*

$$\widehat{f}(\boldsymbol{x}) = \operatorname*{argmin}_{1 \leq j \leq k} d(\boldsymbol{x}, \boldsymbol{X}^j)$$

- What do we mean by distance here?

Discriminant analysis

Nearest centroid rule

- If we can assign a *central point* or *centroid* $\widehat{\mu}_j$ to each $\boldsymbol{X}^j$, then we can define the distance above as distance to the centroid $\widehat{\mu}_j$.
- This yields the nearest centroid rule

$$\widehat{f}(\boldsymbol{x}) = \underset{1 \leq j \leq k}{\operatorname{argmin}} \, d(\boldsymbol{x}, \widehat{\mu}_j)$$

Discriminant analysis

Nearest centroid rule

- This rule is described completely by the functions

$$h_{ij}(\boldsymbol{x}) = \frac{d(\boldsymbol{x}, \widehat{\mu}_j)}{d(\boldsymbol{x}, \widehat{\mu}_i)}$$

with $\widehat{f}(\boldsymbol{x})$ being any i such that

$$h_{ij}(\boldsymbol{x}) \geq 1 \ \forall j.$$

Discriminant analysis

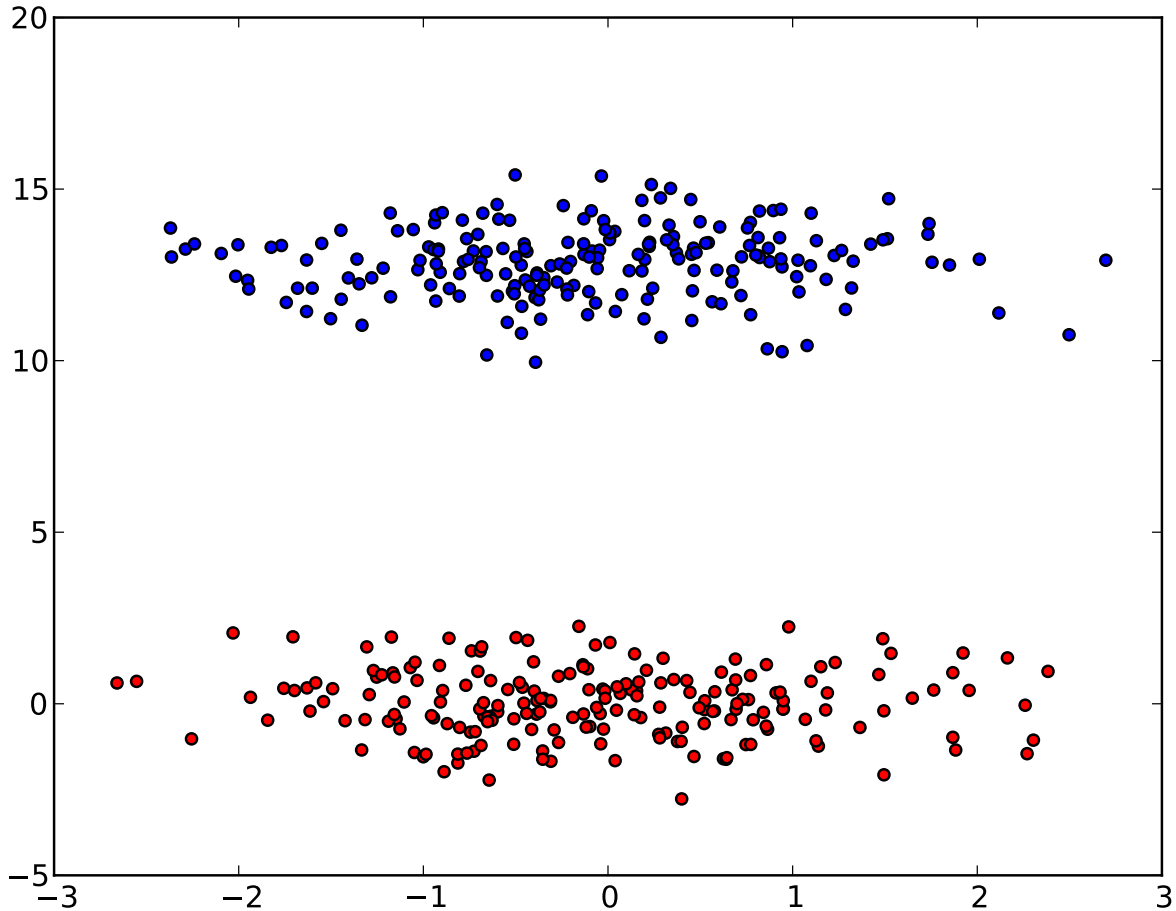
Nearest centroid rule

- If $d(\boldsymbol{x}, \boldsymbol{y}) = \|x - y\|_2$ then the natural centroid is

$$\widehat{\mu}_j = \frac{1}{n_j} \sum_{l=1}^{n_j} \boldsymbol{x}_l^j$$

- This rule just classifies points to the nearest $\widehat{\mu}_j$.
- But, if the covariance matrix of our data is not I, this rule ignores structure in the data . . .

Why we should use covariance



Discriminant analysis

Choice of distance

- Often, there is some background *model* for our data that is equivalent to a given procedure.
- For this nearest centroid rule, using the Euclidean distance effectively assumes that within the set of points $\boldsymbol{X}^j$, the rows are multivariate Gaussian with covariance matrix proportional to I.
- It also implicitly assumes that the n_j's are roughly equal.
- For instance, if one n_j was very small just because a data point is close to that $\widehat{\mu}_j$ doesn't necessarily mean that we should conclude it has label j because there might be a huge number of points of label i near that $\widehat{\mu}_j$.

Discriminant analysis

Gaussian discriminant functions

- Suppose each group with label j had its own mean μ_j and covariance matrix Σ_j, as well as proportion π_j.
- The Gaussian discriminant functions are defined as

$$h_{ij}(x) = h_i(x) - h_j(x)$$
$$h_i(x) = \log \pi_i - \log |\Sigma_i|/2 - (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)/2$$

- The first term weights the prior probability, the second two terms are $\log \phi_{\mu_i, \Sigma_i}(x) = \log L(\mu_i, \Sigma_i | x)$.
- Ignoring the first two terms, the h_i's are essentially within-group Mahalanobis distances . . .

Discriminant analysis

Gaussian discriminant functions

- The classifier assigns $\boldsymbol{x}$ label i if $h_i(\boldsymbol{x}) \geq h_j(\boldsymbol{x}) \forall j$.
- Or,

$$f(\boldsymbol{x}) = \underset{1 \leq i \leq k}{\operatorname{argmax}}\, h_i(\boldsymbol{x})$$

- This is equivalent to a Bayesian rule. We'll see more Bayesian rules when we talk about naïve Bayes . . .
- When all Σ_i and π_i's are identical, the classifier is just nearest centroid using Mahalanobis distance instead of Euclidean distance.

Discriminant analysis

Estimating discriminant functions

- In practice, we will have to estimate π_j, μ_j, Σ_j.
- Obvious estimates:

$$\widehat{\pi}_j = \frac{n_j}{\sum_{j=1}^k n_j}$$

$$\widehat{\mu}_j = \frac{1}{n_j} \sum_{l=1}^{n_j} X_l^j$$

$$\widehat{\Sigma}_j = \frac{1}{n_j - 1} (X^j - \widehat{\mu}_j \mathbf{1})^T (X^j - \widehat{\mu}_j \mathbf{1})$$

Discriminant analysis

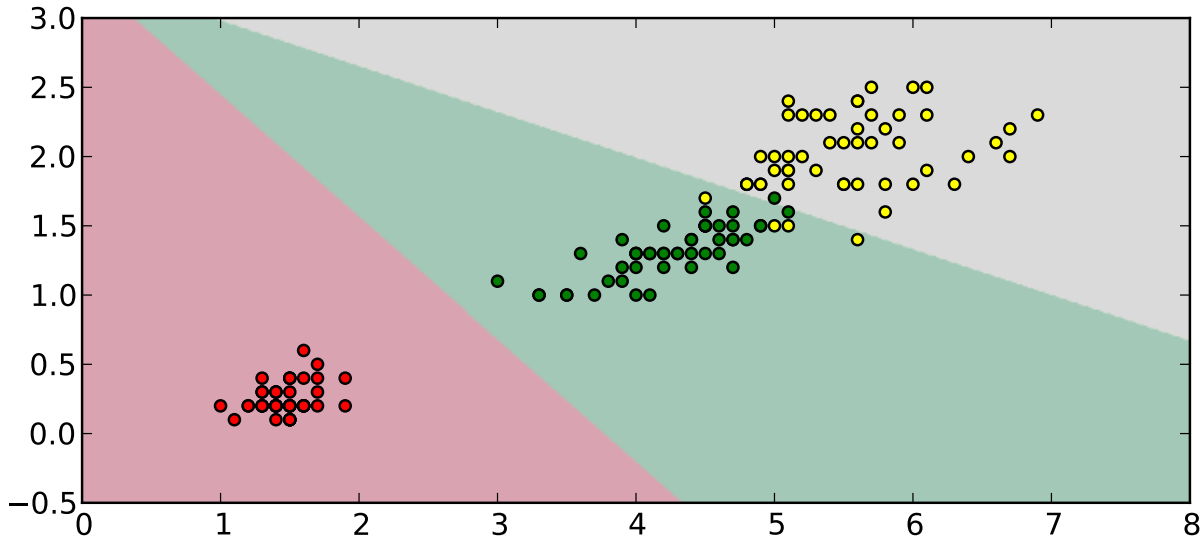
Estimating discriminant functions

- If we assume that the covariance matrix is the same within groups, then we might also form the pooled estimate

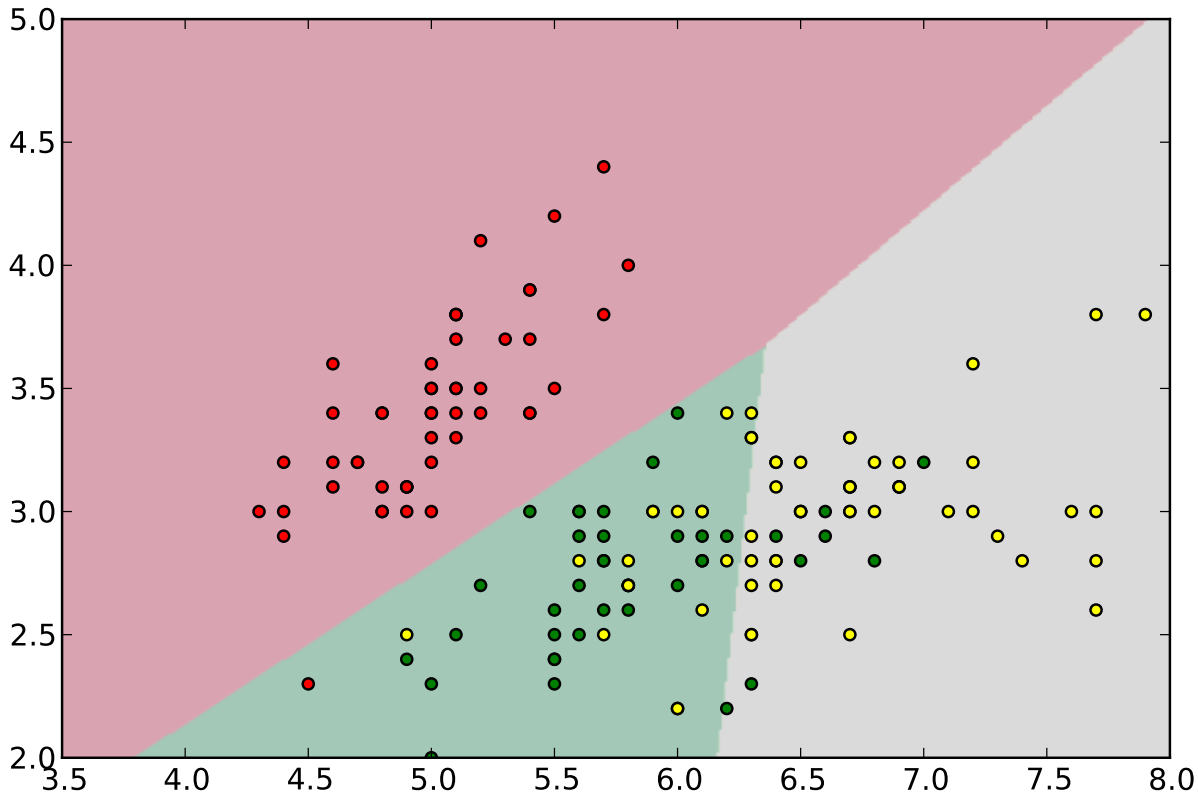
$$\widehat{\Sigma}_P = \frac{\sum_{j=1}^{k}(n_j - 1)\widehat{\Sigma}_j}{\sum_{j=1}^{k} n_j - 1}$$

- If we use the pooled estimate $\Sigma_j = \widehat{\Sigma}_P$ and plug these into the Gaussian discrimants, the functions $h_{ij}(\boldsymbol{x})$ are *linear* (or affine) functions of $\boldsymbol{x}$.
- This is called Linear Discriminant Analysis (LDA).
- Not to be confused with the other LDA (Latent Dirichlet Allocation) . . .

Linear Discriminant Analysis using (`petal.width`, `petal.length`)



Linear Discriminant Analysis using (`sepal.width`, `sepal.length`)

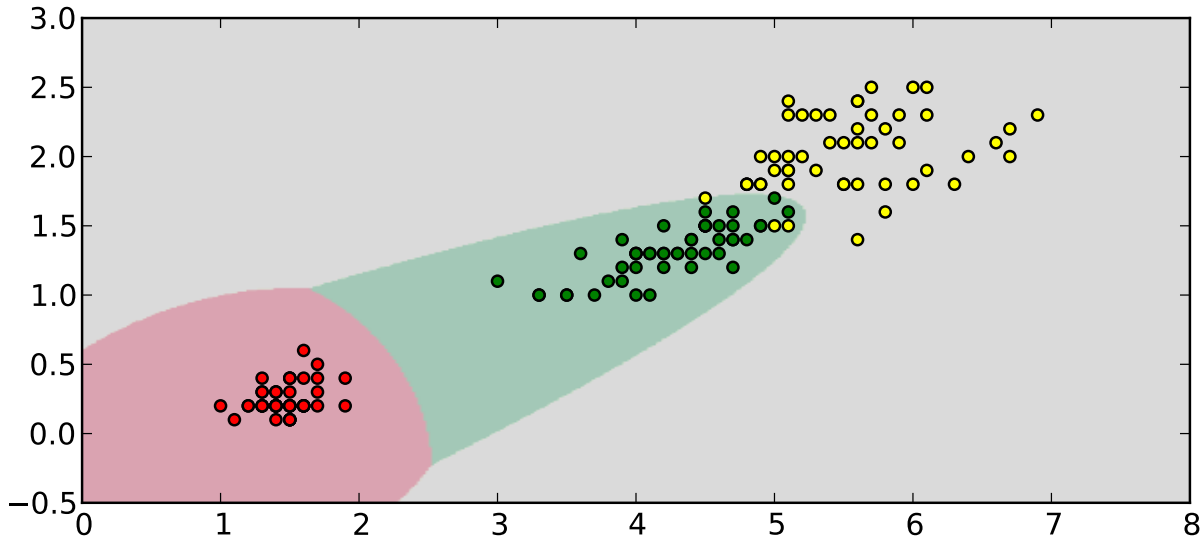


Discriminant analysis

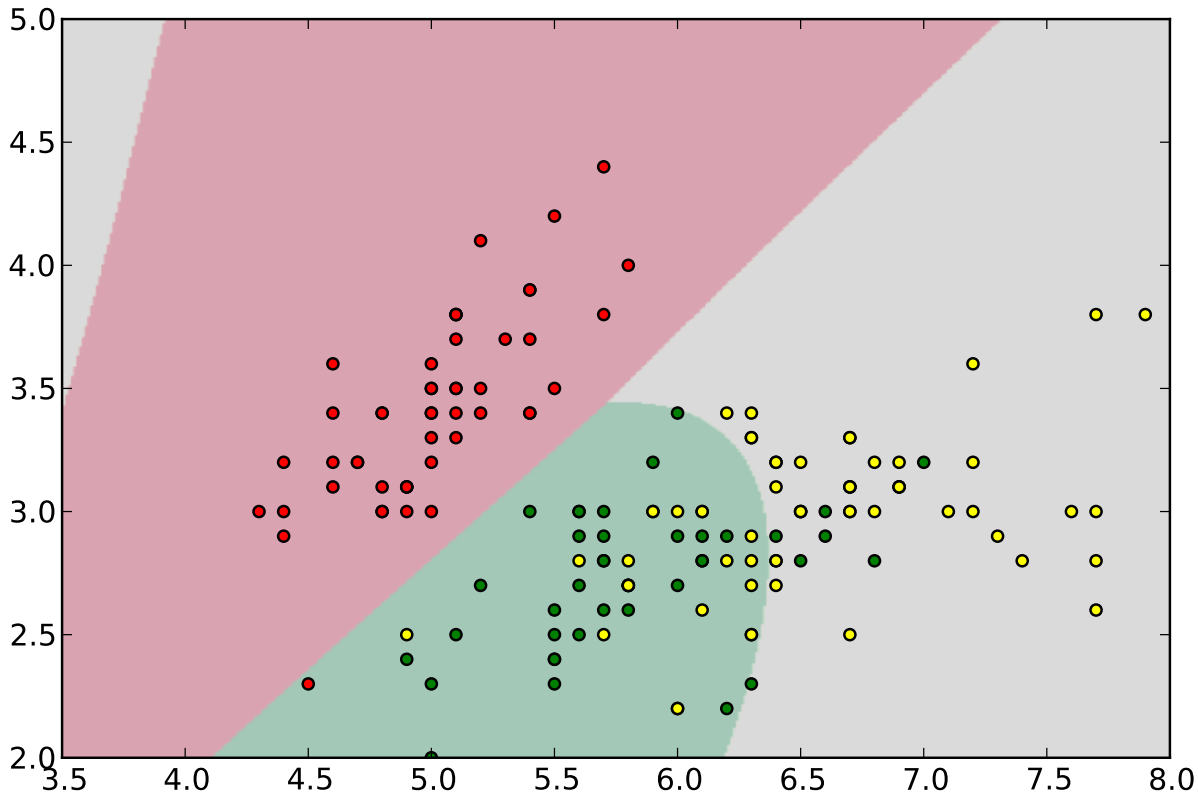
Quadratic Discriminant Analysis

- If we use don't use pooled estimate $\Sigma_j = \widehat{\Sigma}_j$ and plug these into the Gaussian discrimants, the functions $h_{ij}(\boldsymbol{x})$ are *quadratic* functions of $\boldsymbol{x}$.
- This is called Quadratic Discriminant Analysis (QDA).

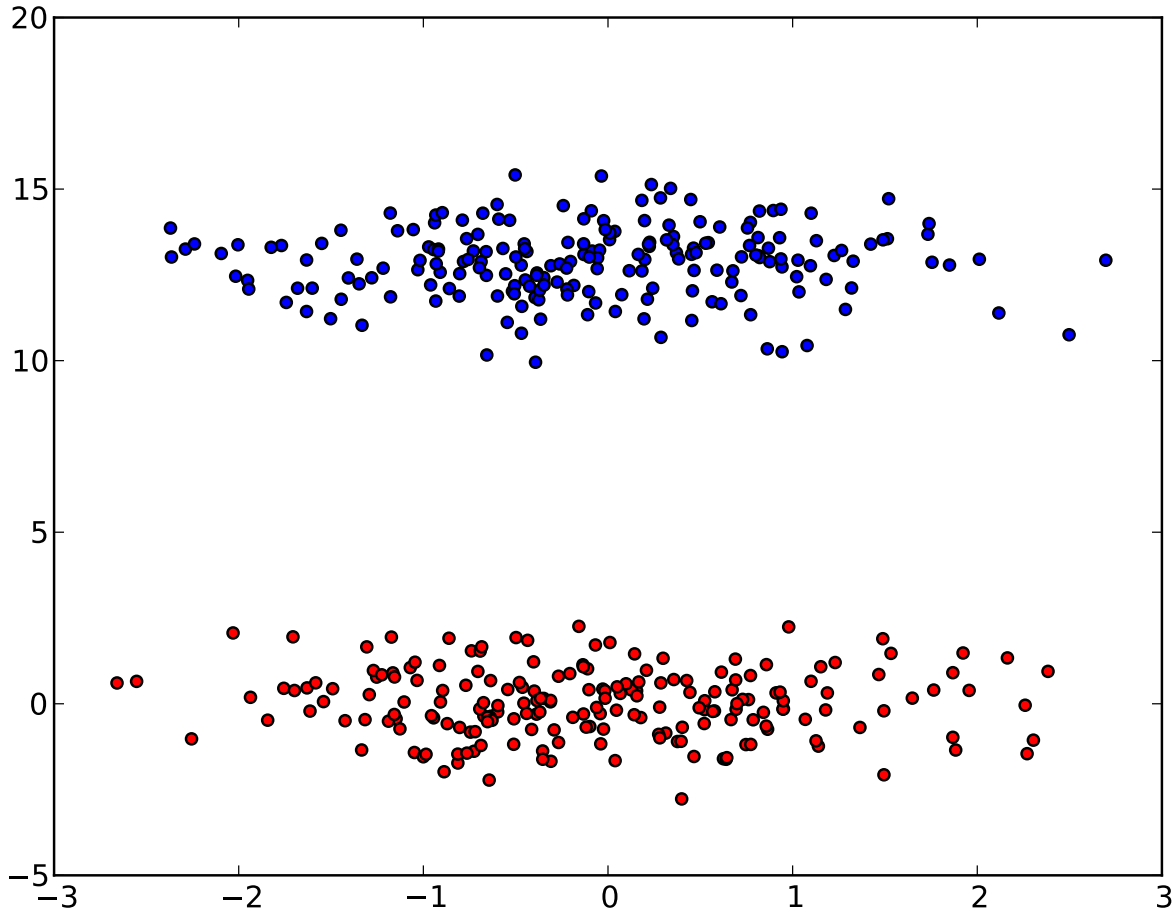
Quadratic Discriminant Analysis using (`petal.width, petal.length`)



Quadratic Discriminant Analysis using (`sepal.width, sepal.length`)



Motivation for Fisher's rule



Discriminant analysis

Fisher's discriminant function

- Fisher proposed to classify using a linear rule.
- He first decomposed

$$(\boldsymbol{X} - \hat{\mu}\mathbf{1})^T(\boldsymbol{X} - \hat{\mu}\mathbf{1}) = \widehat{SS}_B + \widehat{SS}_W$$

- Then, he proposed,

$$\hat{v} = \text{argmax}_{v: v^T \widehat{SS}_W v = 1} v^T \widehat{SS}_B v$$

- Having found $\hat{v}$, form $\boldsymbol{X}^j\hat{v}$ and the centroid $\eta_j = \text{mean}(\boldsymbol{X}^j\hat{v})$
- In the two-class problem $k = 2$, this is the same as LDA.

Discriminant analysis

Fisher's discriminant functions

- The direction $\widehat{v}_1$ is an eigenvector of some matrix. There are others, up to $k-2$ more.
- Suppose we find all $k-1$ vectors and form $\boldsymbol{X}_j V^T$, each one an $n_j \times (k-1)$ matrix with centroid $\eta_j \in \mathbb{R}^{k-1}$.
- The matrix V determines a map from $\mathbb{R}^p$ to $\mathbb{R}^{k-1}$, so given a new data point we can compute $V\boldsymbol{x} \in \mathbb{R}^{k-1}$.
- This gives rise to a classifier

$$\widehat{f}(\boldsymbol{x}) = \text{nearest centroid}(V\boldsymbol{x})$$

- This is LDA (assuming $\pi_j = \frac{1}{k}$) ...

Discriminant analysis

Discriminant models in general

- A discriminant model is generally a model that estimates

$$P(Y = j|\boldsymbol{x}), 1 \leq j \leq k$$

- That is, given that the features I observe are $\boldsymbol{x}$, the probability I think this label is j ...
- LDA and QDA are actually *generative models* since they specify

$$P(X = \boldsymbol{x}|Y = j).$$

- There are lots of discriminant models . . .

Discriminant analysis

Logistic regression

- The logistic regression model is ubiquitous in binary classification (two-class) problems
- Model:

$$P(Y = 1|\boldsymbol{x}) = \frac{\alpha + e^{\boldsymbol{x}^T\beta}}{1 + e^{\alpha+\boldsymbol{x}^T\beta}} = \pi(\alpha, \beta, \boldsymbol{x})$$

Discriminant analysis

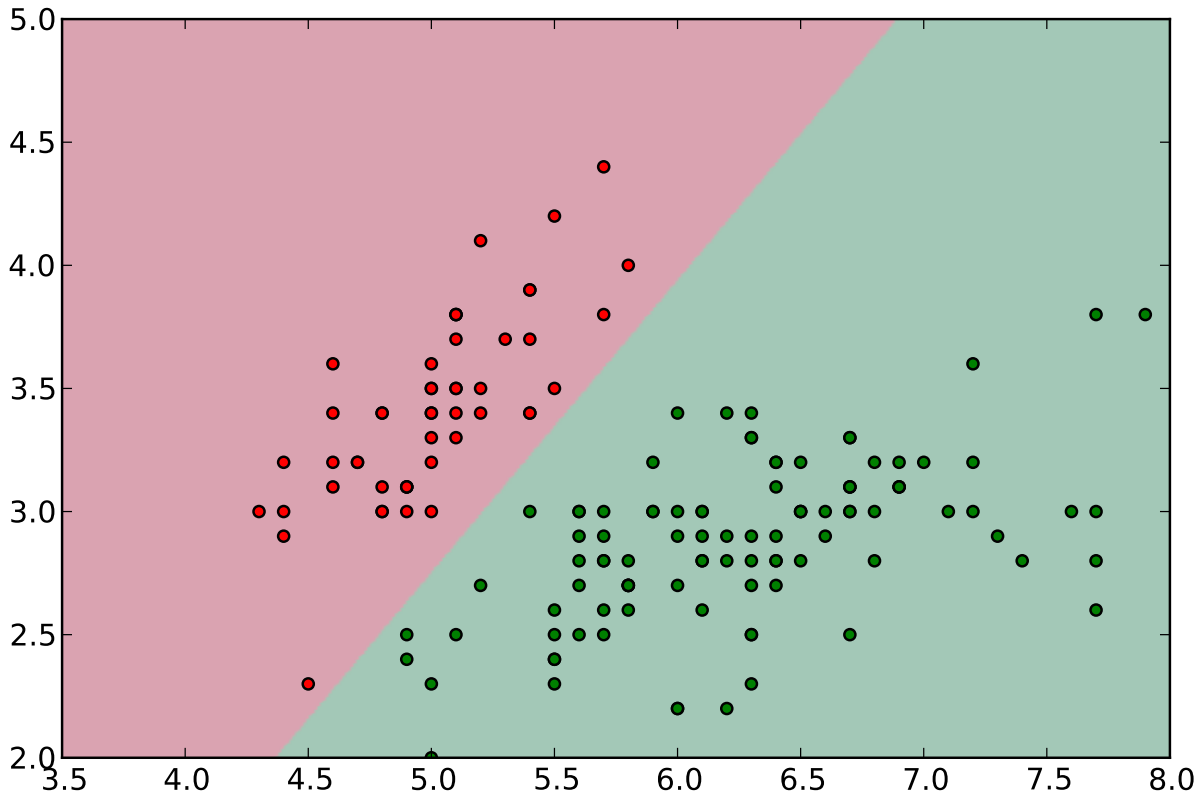
Logistic regression

- Software that fits a logistic regression model produces an estimate of β based on a data matrix $\boldsymbol{X}_{n \times p}$ and binary labels $\boldsymbol{Y}_{n \times 1} \in \{0,1\}^n$
- It fits the model minimizing what we call the *deviance*

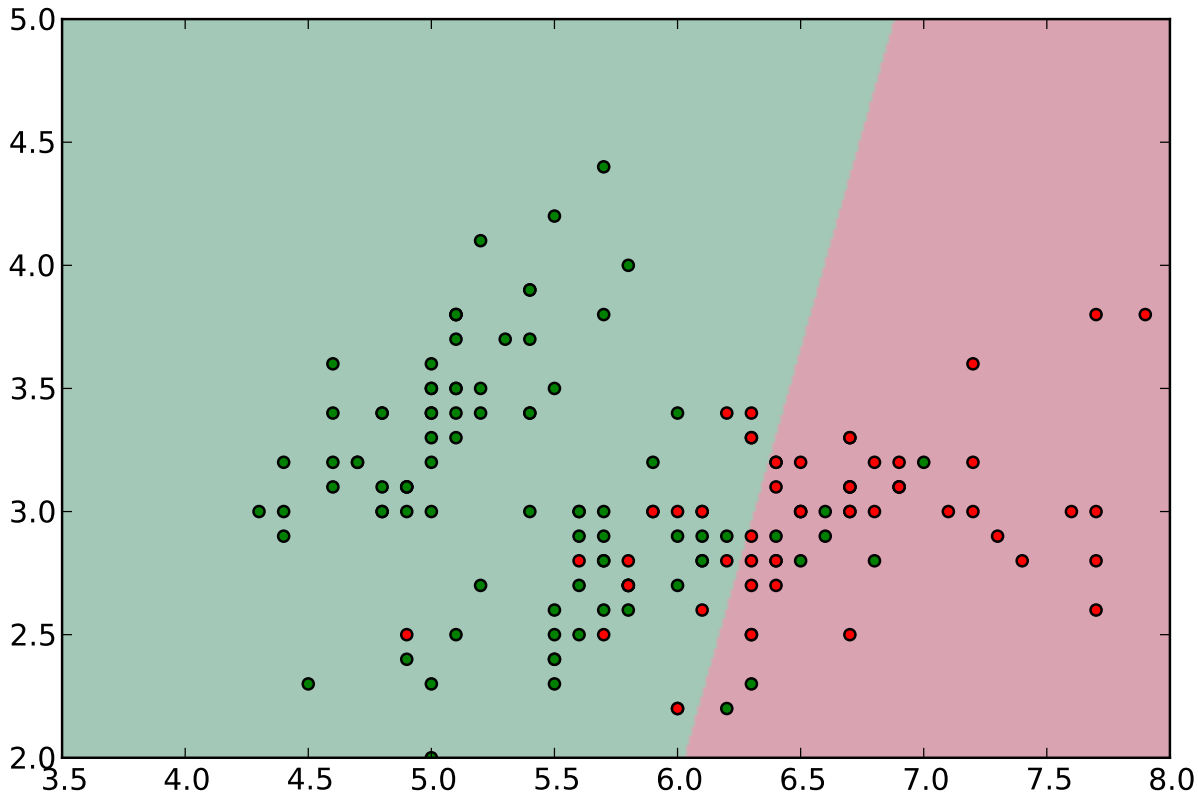
$$\text{DEV}(\beta) = -2 \sum_{i=1}^{n} (\boldsymbol{Y}_i \log \pi(\beta, \boldsymbol{X}_i) + (1 - \boldsymbol{Y}_i) \log(1 - \pi(\beta, \boldsymbol{X}_i))$$

- While not immediately obvious, this is a convex minimization problem, hence is fairly easy to solve.
- Unlike trees, the convexity yields a globally optimal solution.

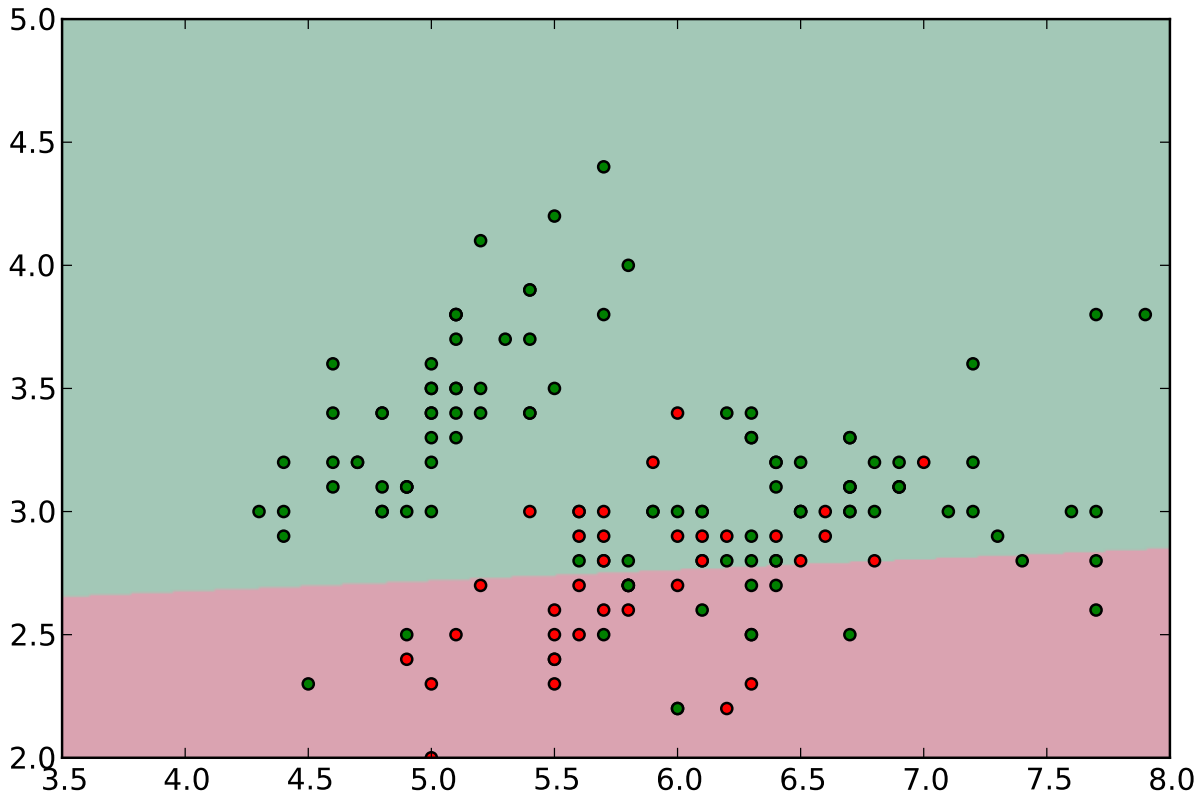
Logistic regression, *setosa* vs *virginica*, *versicolor* using (`sepal.width, sepal.length`)



Logistic regression, *virginica* vs *setosa*, *versicolor* using (`sepal.width, sepal.length`)



Logistic regression, *versicolor* vs *setosa*, *virginica* using (`sepal.width, sepal.length`)



Discriminant analysis

Logistic regression

- Logistic regression produces an estimate of

$$P(\boldsymbol{y} = 1|\boldsymbol{x}) = \widehat{\pi}(\boldsymbol{x})$$

- Typically, we classify as 1 if $\widehat{\pi}(\boldsymbol{x}) > 0.5$.
- This yields a 2×2 confusion matrix

	Predicted: 0	Predicted: 1
Actual : 0	*TN*	*FP*
Actual : 1	*FN*	*TP*

- From the 2×2 confusion matrix, we can compute Sensitivity, Specificity, etc.

Discriminant analysis

Logistic regression

- However, we could choose the threshold differently, perhaps related to estimates of the prior probabilities of 0's and 1's.
- Now, each threshold $0 \leq t \leq 1$ yields a new confusion matrix
- This yields a 2×2 confusion matrix

	Predicted: 0	Predicted: 1
Actual : 0	$TN(t)$	$FP(t)$
Actual : 1	$FN(t)$	$TP(t)$

Discriminant analysis

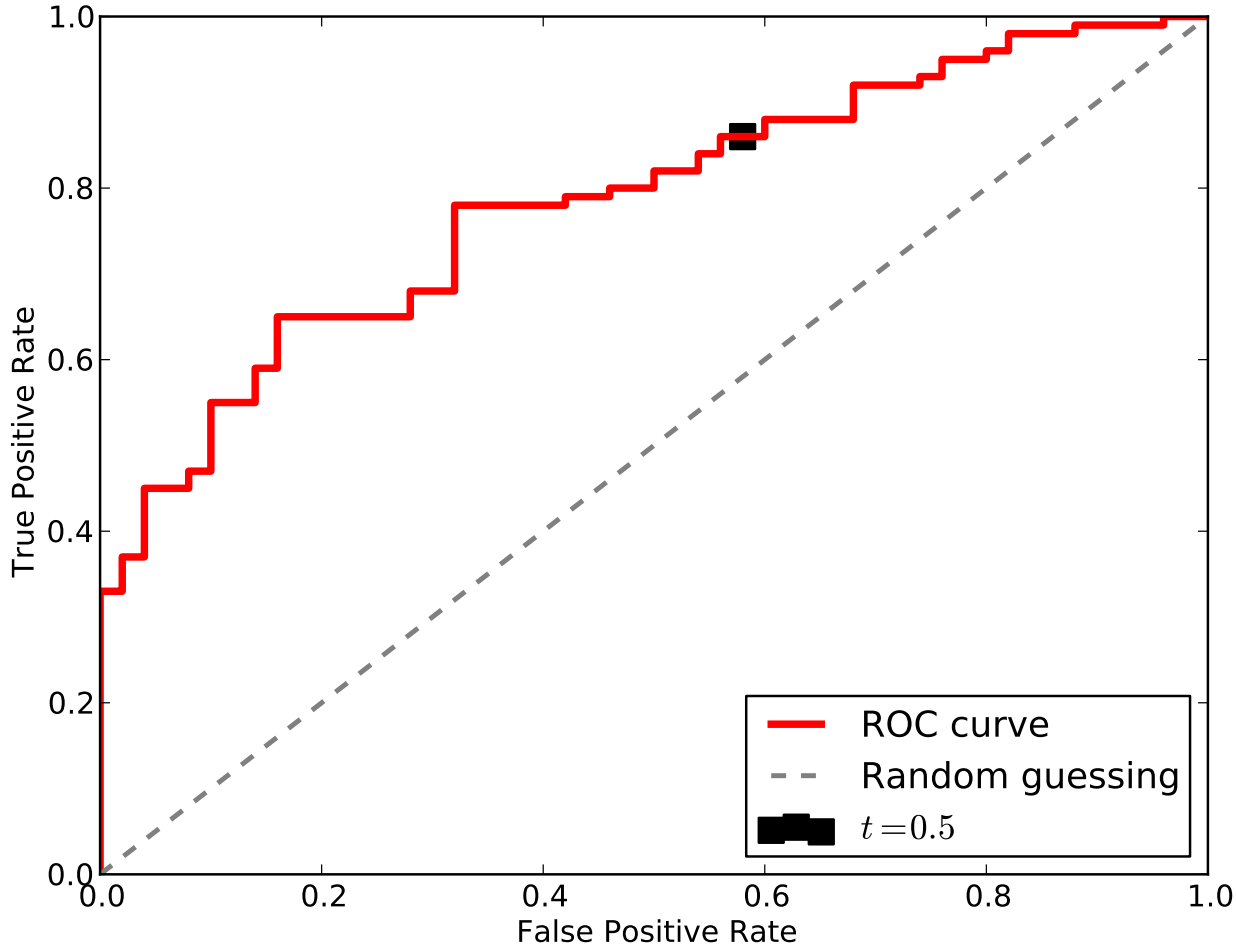
ROC curve

- Generally speaking, we prefer classifiers that are both highly sensitive and highly specific.
- These confusion matrices can be summarized using an ROC (Receiver Operating Characteristic) curve.
- This is a plot of the curve

$$(1 - \text{Specificity}(t), \text{Sensitivity}(t))_{0 \leq t \leq 1}$$

- Often, Specificity is referred to as *TNR* (True Negative Rate), and $(1 - \text{Specificity}(t))$ as *FPR*.
- Often, Sensitivity is referred to as *TPR* (True Positive Rate), and $(1 - \text{Sensitivity}(t))$ as *FNR*.

AUC: Area under ROC curve



Discriminant analysis

ROC curve

- Points in the upper left of the ROC curve are good.
- Any point on the diagonal line represents a classifier that is guessing randomly.
- A tree classifier as we've discussed seems to correspond to only one point in the ROC plot.
- *But* one can estimate probabilities based on frequencies in terminal nodes.

Discriminant analysis

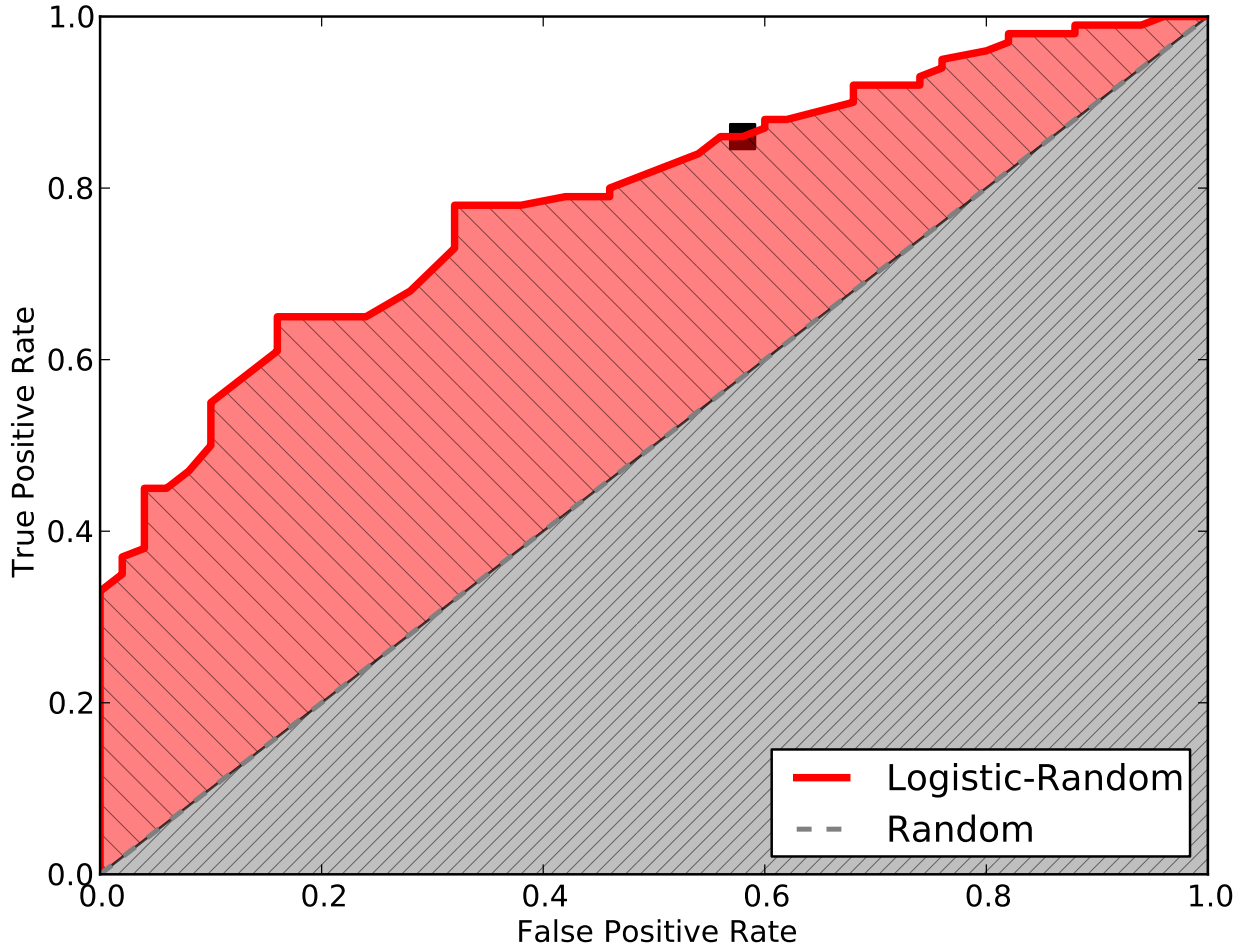
ROC curve

- A common numeric summary of the ROC curve is

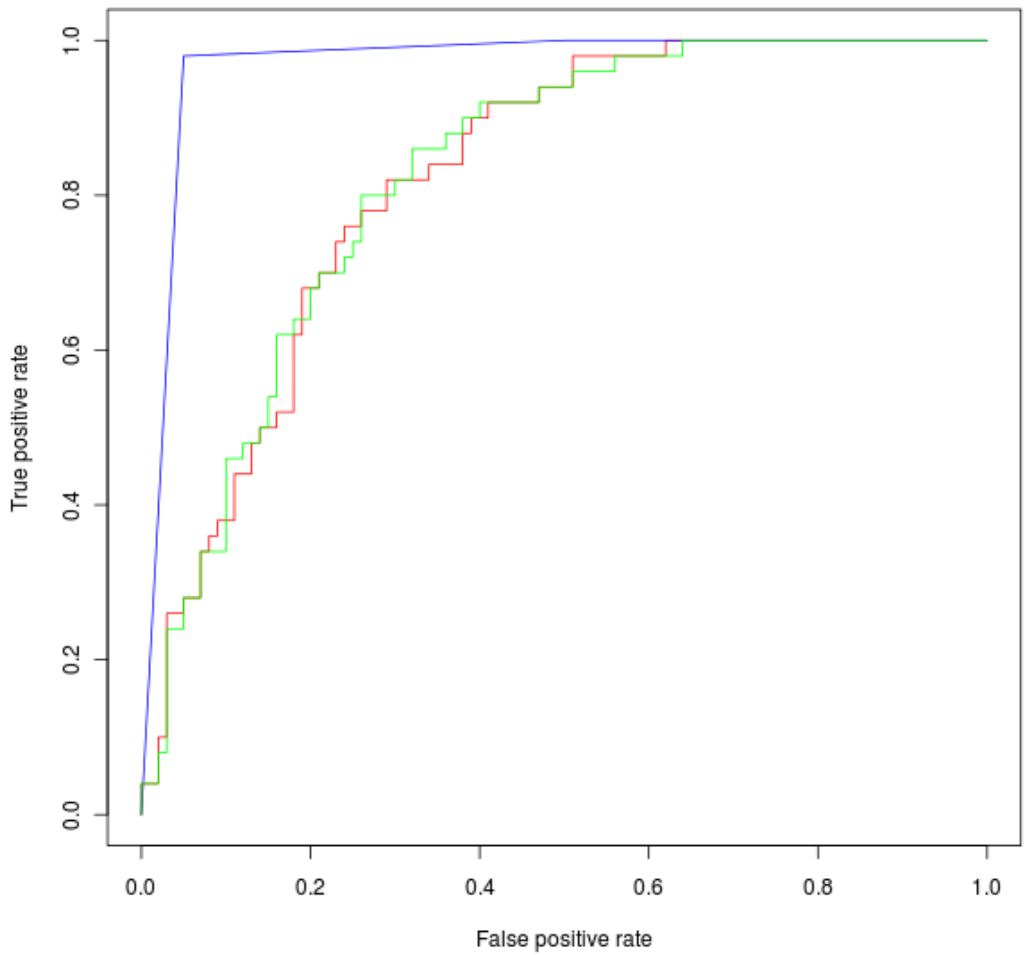
$$AUC(ROC \text{ curve}) = \text{Area under ROC curve.}$$

  - Can be interpreted as an estimate of the probability that the classifier will give a random positive instance a higher score than a random negative instance.
  - Maximum value is 1.
  - For a random guesser, AUC is 0.5

AUC: Area under ROC curve



ROC curve: logistic, rpart, lda



Statistics 202:
Data Mining