

Linear classification methods

Lecture 2a

The top of the slide features a blue background with a pattern of binary code (0s and 1s). A magnifying glass is positioned over the right side of the background, focusing on the text.

Overview

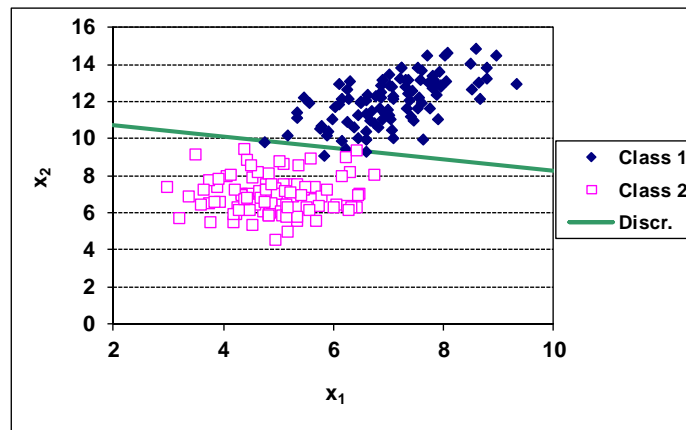
- Elements of decision theory
- Logistic regression
- Discriminant Analysis models

Classification

- Given data $D = ((X_i, Y_i), i = 1 \dots N)$
 - $Y_i = Y(X_i) = C_j \in \mathcal{C}$
 - Class set $\mathcal{C} = (C_1, \dots, C_K)$

Classification problem:

- Decide $\hat{Y}(x)$ that maps **any** x into some class C_K
 - Decision boundary



Classifiers

- **Deterministic**: decide a rule that directly maps X into \hat{Y}
- **Probabilistic**: define a model for $P(Y = C_i | X), i = 1 \dots K$

Disadvantages of deterministic classifiers:

- Sometimes simple mapping is not enough (risk of cancer)
- Difficult to embed loss \rightarrow rerun of optimizer is often needed
- Combining several classifiers into one is more problematic
 - Algorithm A classifies as spam, Algorithm B classifies as not spam \rightarrow ???
 - $P(\text{Spam} | A) = 0.99, P(\text{Spam} | B) = 0.45 \rightarrow$ better decision can be made

Bayesian decision theory

- Machine learning models estimate $p(y|x)$ or $p(y|x, \hat{w})$
- Transform probability into action \rightarrow which value to predict? \rightarrow decision step
 - $p(Y = Spam|x) = 0.83 \rightarrow$ do we move the mail to Junk?
 - What is more dangerous: deleting 1 non-spam mail or letting 1 spam mail enter Inbox?
- \rightarrow **Loss function** or **Loss matrix**

Loss matrix

- Costs of classifying $Y = C_k$ to C_j :
 - Rows: true, columns: predicted

$$L = \|L_{ij}\|, i = 1, \dots, n, j = 1, \dots, n$$

- **Example 1:** 0/1-loss

$$L = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

- **Example 2:** Spam

$$L = \begin{pmatrix} 0 & 100 \\ 1 & 0 \end{pmatrix}$$

Loss and decision

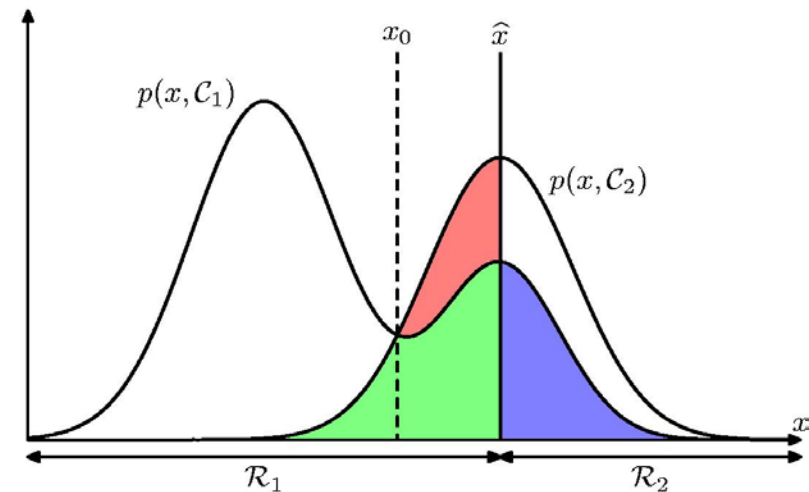
- Expected loss minimization

- R_j : classify to C_j

$$EL = \sum_k \sum_j \int_{R_j} L_{kj} p(\mathbf{x}, C_k) d\mathbf{x}$$

- Choose such R_j that EL is minimized
- Two classes

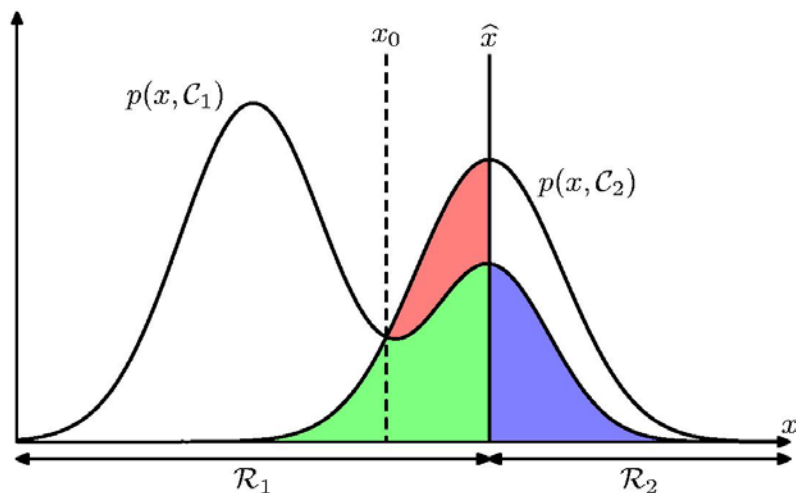
$$EL = \int_{R_1} L_{21} p(x, C_2) dx + \int_{R_2} L_{12} p(x, C_1) dx$$



Loss and decision

- Loss minimization

$$\min_{\hat{f}} EL(y, \hat{f}) = \min_{\hat{f}} \int L(y, \hat{f}) p(y, x|w) dx dy$$



When loss is

$\begin{cases} 1, \text{wrongly classified} \\ 0, \text{correctly classified} \end{cases}$

Classify Y as

$$\hat{Y} = \arg \max_c p(Y = c|X)$$

Loss and decision

- How to minimize *EL with two classes?*
- Rule:
 - $L_{12}p(x, C_1) > L_{21}p(x, C_2) \rightarrow \text{predict } y \text{ as } C_1$
- 0/1 Loss: **classify to the class which is more probable!**

$$\frac{p(C_1|x)}{p(C_2|x)} > \frac{L_{21}}{L_{12}} \rightarrow \text{predict } y \text{ as } C_1$$

Loss and decision

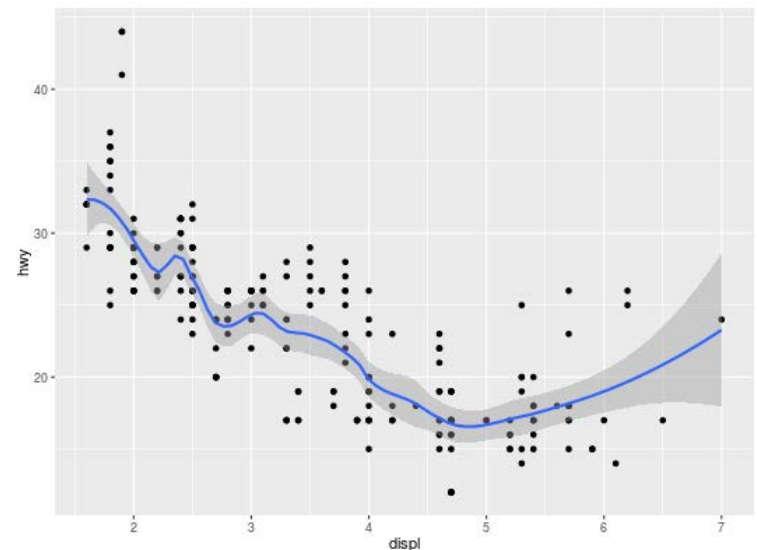
- Continuous targets: squared loss

- Given a model $p(x, y)$, minimize

$$EL = \int L(y, \hat{Y}(x)) p(x, y) dx dy$$

- Using **square loss**, the optimal is posterior mean

$$\hat{Y}(x) = \int y p(y|x) dy$$



ROC curves

- Binary classification
- The choice of the threshold $\hat{x} = \frac{L_{21}}{L_{12}}$ affects prediction \rightarrow what if we don't know the loss? Which classifier is better?
- **Confusion matrix**

	PREDICTED			
T R U E		1	0	Total
	1	TP	FN	N_+
	0	FP	TN	N_-

ROC curves

- **True Positive Rates (TPR) = sensitivity = recall**

- Probability of detection of positives: TPR=1 positives are correctly detected

$$TPR = TP/N_+$$

- **False Positive Rates (FPR)**

- Probability of false alarm: system alarms (1) when nothing happens (true=0)

$$FPR = FP/N_-$$

- **Specificity**

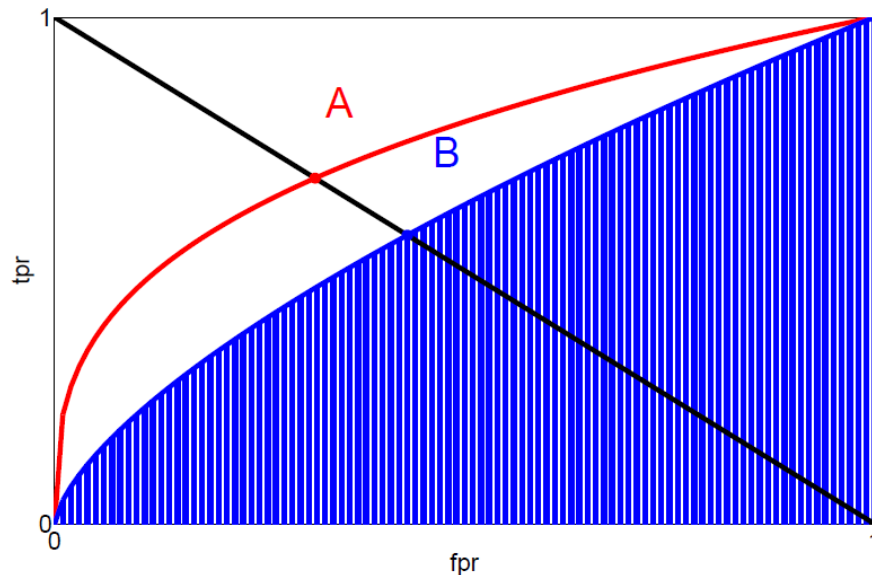
$$Specificity = 1 - FPR$$

- **Precision**

$$Precision = \frac{TP}{TP + FP}$$

ROC curves

- **ROC**=Receiver operating characteristics
- Use various thresholds, measure TPR and FPR
- Same FPR, higher TPR → better classifier
- Best classifier = greatest Area Under Curve (**AUC**)



Types of supervised models

- **Generative models:** model $p(X|Y, w)$ and $p(Y|w)$

- **Example:** k-NN classification

$$p(X = x|Y = C_i, K) = \frac{K_i}{N_i V}, p(C_i|K) = \frac{N_i}{N}$$

From Bayes Theorem,

$$p(Y = C_i|x, K) = \frac{K_i}{K}$$

- **Discriminative models:** model $p(Y|X, w)$, X constant

- **Example:** logistic regression

- $p(Y = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}}$

Generative vs Discriminative

- Generative can be used to generate new data
- Generative normally easier to fit (check Logistic vs K-NN)
- Generative: each class estimated separately → do not need to retrain when a new class added
- Discriminative models: can replace X with $\phi(X)$ (preprocessing), method will still work
 - Not generative, distribution will change
- Generative: often make too strong assumptions about $p(X|Y, w)$ → bad performance

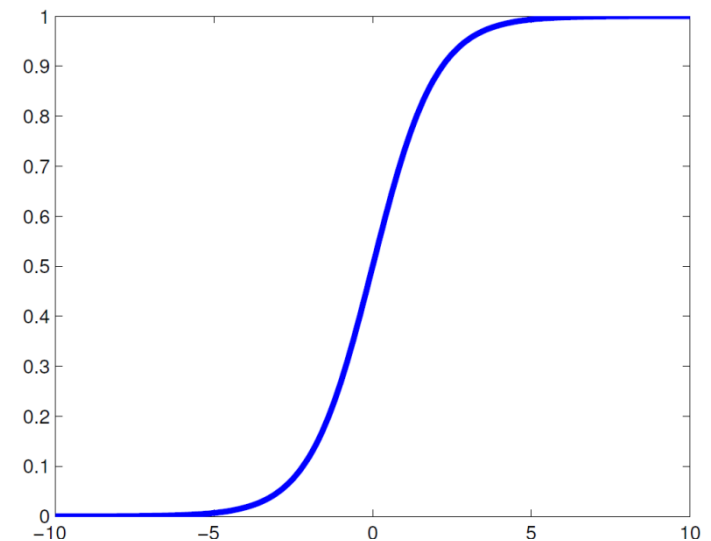
Logistic regression

- Discriminative model
- Model for binary output
 - $C = \{C_1 = 1, C_2 = 0\}$
 $p(Y = C_1|X) = \text{sigm}(\mathbf{w}^T \mathbf{x})$

$$\text{sigm}(a) = \frac{1}{1 + e^{-a}}$$

- Alternatively
 $Y \sim \text{Bernoulli}(\text{sigm}(a)), a = \mathbf{w}^T \mathbf{x}$
$$\text{sigm}(a) = \frac{1}{1 + e^{-a}}$$

What is $P(Y = C_2|X)$?



Logistic regression

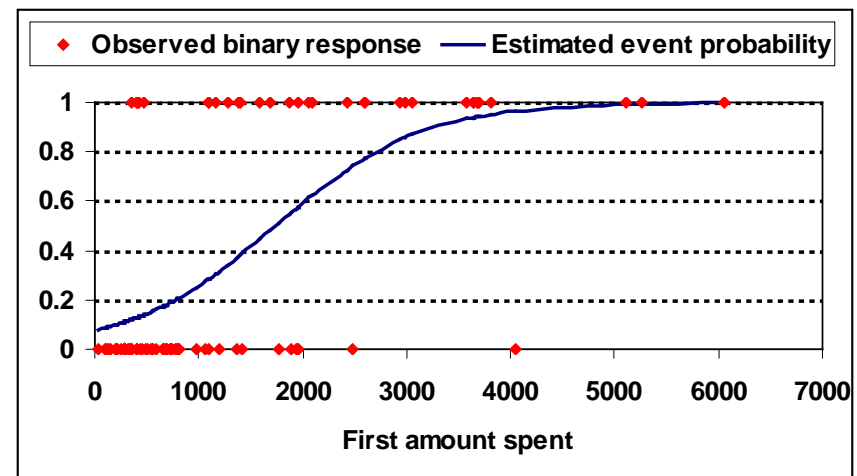
- Logistic model- yet another form

$$\ln \frac{p(Y = 1|X = x)}{P(Y = 0|X = x)} = \ln \frac{p(Y = 1|X = x)}{1 - P(Y = 1|X = x)} = \text{logit}(p(Y = 1|X = x)) = \mathbf{w}^T \mathbf{x}$$

**The log of the odds
is linear in \mathbf{x}**

- Here $\text{logit}(t) = \ln \left(\frac{t}{1-t} \right)$
- Note $p(Y|X)$ is connected to $\mathbf{w}^T \mathbf{x}$ via logit link

Example: Probability to buy
more than once as function of
First Amount Spend



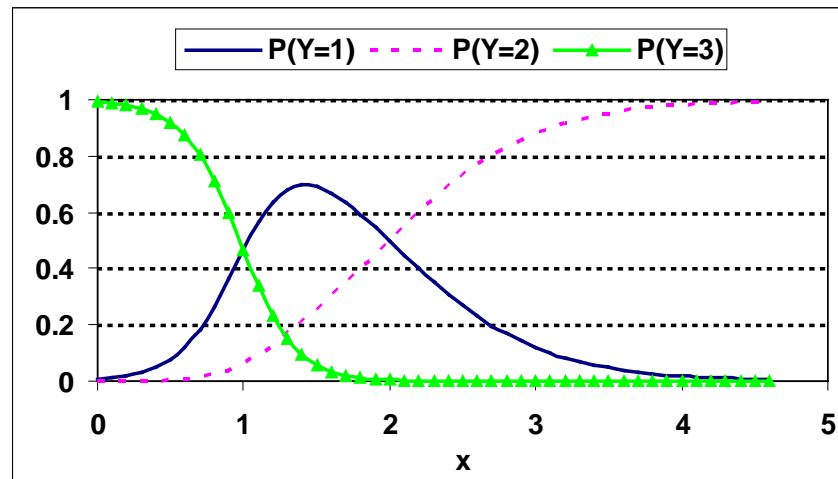
Logistic regression

- When Y is categorical,

$$p(Y = C_i | x) = \frac{e^{w_i^T x}}{\sum_{j=1}^K e^{w_j^T x}} = \text{softmax}(w_i^T x)$$

- Alternatively

$$Y \sim \text{Multinoulli}(\text{softmax}(w_1^T x), \dots, \text{softmax}(w_K^T x))$$



Logistic regression

Fitting logistic regression

- In binary case,

$$\log P(D|w) = \sum_{i=1}^N y_i \log(\text{sigm}(w^T x_i)) + (1 - y_i) \log(1 - \text{sigm}(w^T x_i))$$

- Can not be maximized analytically, but unique maximizer exists
- To maximize loglikelihood, optimization used
 - Newton's method traditionally used (Iterative Reweighted Least Squares)
 - Steepest descent, Quasi-newton methods...

Estimation:

For new x , estimate $p(y) = [p_1, \dots, p_c]$ and classify as $\arg \max_i p_i$

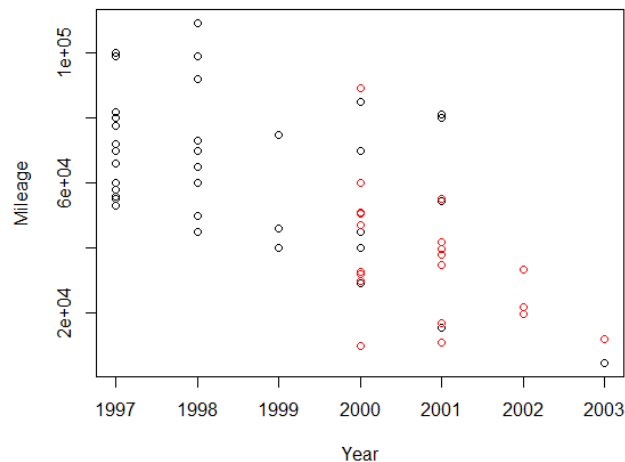
Decision boundaries of logistic regression are linear

Logistic regression

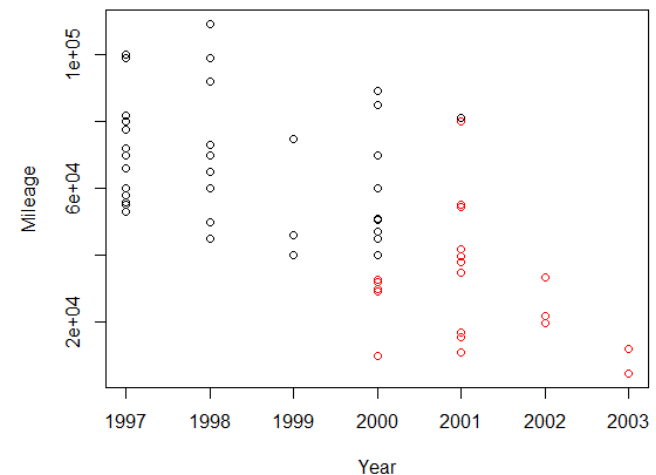
- In R, use `glm()` with `family="binomial"`
 - Predicted probabilities: `predict(fit,newdata, type="response")`

Example Equipment=f(Year, mileage)

Original data



Classified data

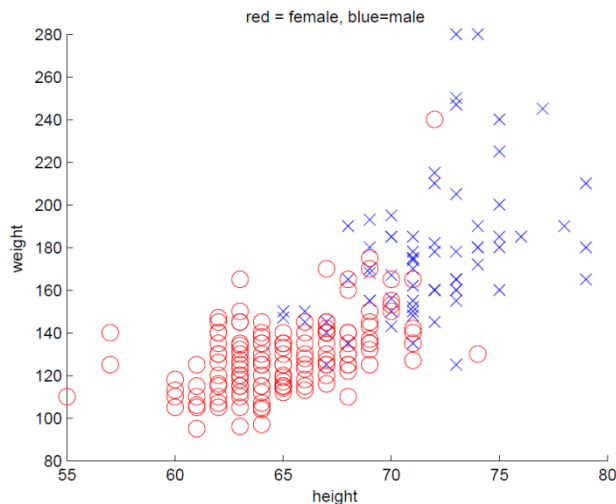
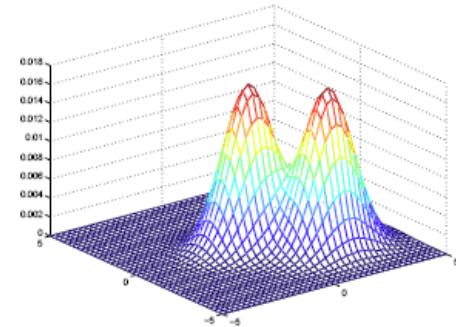


Quadratic discriminant analysis

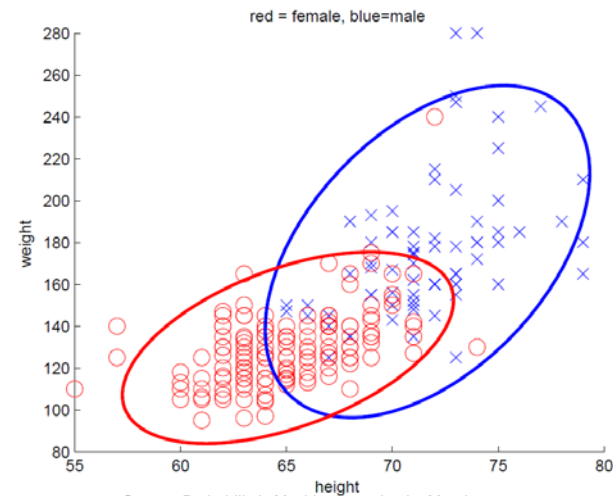
- Generative classifier
- Main assumptions:
 - \mathbf{x} is now **random** as well as y

$$p(\mathbf{x}|y = C_i, \theta) = N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

Unknown parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$



Source: Probabilistic Machine Learning by Murphy

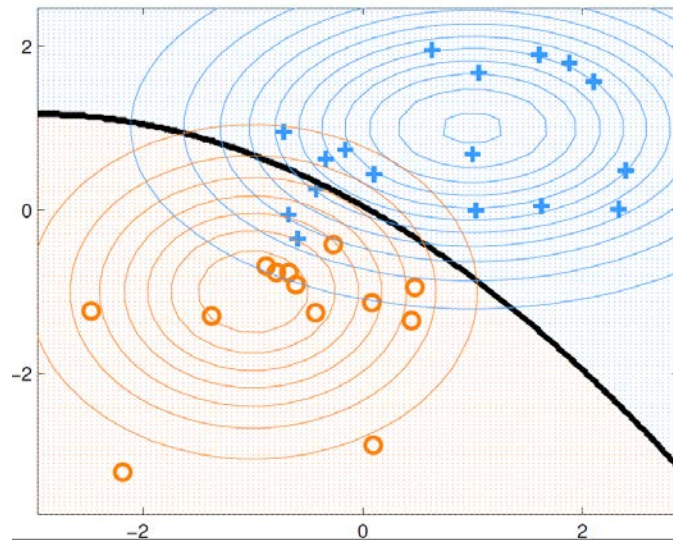


Source: Probabilistic Machine Learning by Murphy

Quadratic discriminant analysis

- If parameters are estimated, classify:

$$\hat{y}(\mathbf{x}) = \arg \max_c p(y = c | \mathbf{x}, \theta)$$

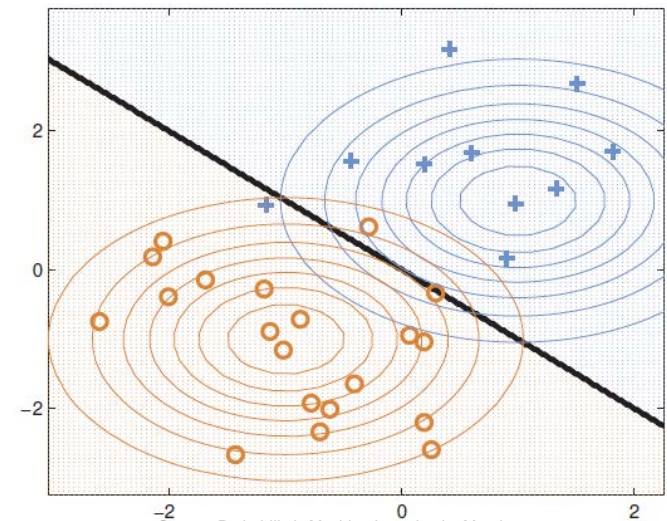


Source: Probabilistic Machine Learning by Murphy

Linear discriminant analysis (LDA)

- Assumption $\Sigma_i = \Sigma, i = 1, \dots, K$
- Then $p(y = c_i | x) = \text{softmax}(w_i^T x + w_{0i}) \rightarrow$ exactly the same form as the logistic regression
 - $w_{0i} = -\frac{1}{2} \mu_i^T \Sigma^{-1} \mu_i + \log \pi_i$
 - $w_i = \Sigma^{-1} \mu_i$
- Decision boundaries are linear
 - **Discriminant function:**

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$



Linear discriminant analysis (LDA)

- Difference LDA vs logistic regression??
 - Coefficients will be estimated differently! (models are different)
- How to estimate coefficients
 - find MLE.

$$\hat{\mu}_c = \frac{1}{N_c} \sum_{i:y_i=c} \mathbf{x}_i, \quad \hat{\Sigma}_c = \frac{1}{N_c} \sum_{i:y_i=c} (\mathbf{x}_i - \hat{\mu}_c)(\mathbf{x}_i - \hat{\mu}_c)^T$$

$$\hat{\Sigma} = \frac{1}{N} \sum_{c=1}^k N_c \hat{\Sigma}_c$$

- Sample mean and sample covariance are MLE!
- If class priors are parameters (**proportional priors**),

$$\hat{\pi}_c = \frac{N_c}{N}$$

LDA and QDA: code

- Syntax in R, library **MASS**

`lda(formula, data, ..., subset, na.action)`

- Prior – class probabilities
- Subset – indices, if training data should be used

`qda(formula, data, ..., subset, na.action)`

`predict(..)`

LDA: output

```
resLDA=lda(Equipment~Mileage+Year, data=mydata)
print(resLDA)
```

```
> print(resLDA)
Call:
lda(Equipment ~ Mileage + Year, data = mydata)

Prior probabilities of groups:
      0      1
0.6440678 0.3559322

Group means:
      Mileage      Year
0 63539.21 1998.447
1 36857.62 2000.762

Coefficients of linear discriminants:
              LD1
Mileage -1.500069e-05
Year      5.745893e-01
```

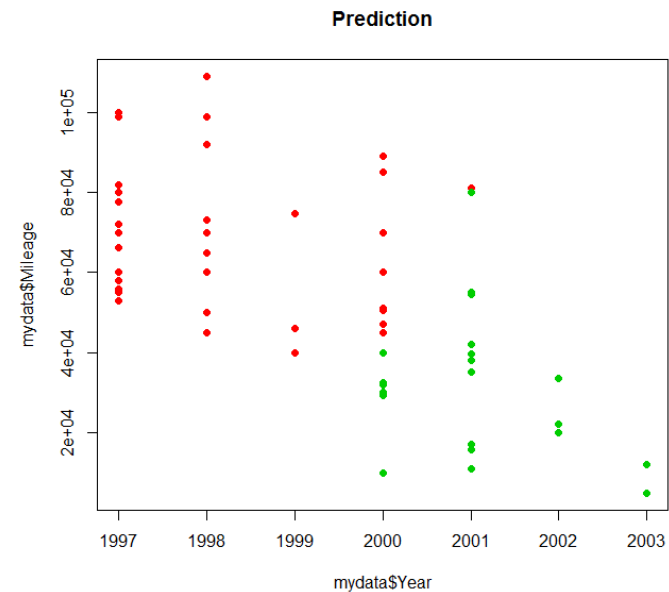
LDA: output

- Misclassified items

```
plot(mydata$Year, mydata$Mileage,  
col=as.numeric(Pred$class)+1, pch=21,  
bg=as.numeric(Pred$class)+1,  
main="Prediction")
```

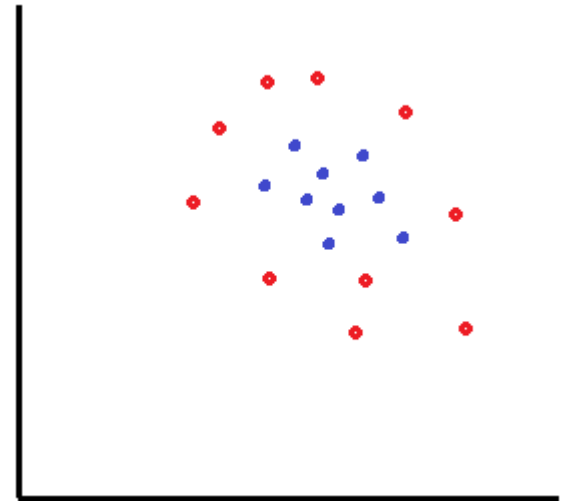
```
> table(Pred$class, mydata$Equipment)
```

```
      0  1  
0 31  6  
1  7 15
```



LDA versus Logistic regression

- Generative classifiers are easier to fit, discriminative involve numeric optimization
- LDA and Logistic have same model form but are fit differently
- LDA has stronger assumptions than Logistic, some other generative classifiers lead also to logistic expression
- New class in the data?
 - Logistic: fit model again
 - LDA: estimate new parameters from the new data
- Logistic and LDA: complex data fits badly unless interactions are included



LDA versus Logistic regression

- LDA (and other generative classifiers) handle missing data easier
- Standardization and generated inputs:
 - Not a problem for Logistic
 - May affect the performance of the LDA in a complex way
- Outliers affect $\Sigma \rightarrow$ LDA is not robust to gross outliers
- LDA is often a good classification method even if the assumption of normality and common covariance matrix are not satisfied.