# Linear classification methods

Lecture 2a

# Overview

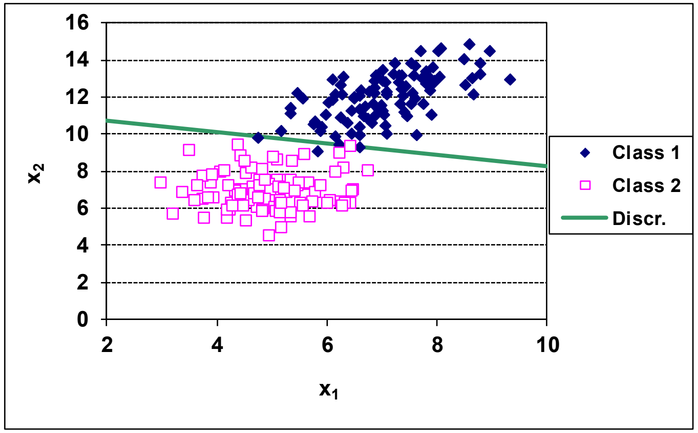
- Elements of decision theory
- Logistic regression
- Discriminant Analysis models

# Classification

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

**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 $P(Y = C_i|X), i = 1 \dots K$

**Disanvantages of deterministic classifiers**:

- Sometimes simple mapping is not enough (risk of cancer)
- Difficult to embed loss-> 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 → ???
  - P(Spam|A)=0.99, P(Spam|B)=0.45 → 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→ which value to predict?→decision step
  - $p(Y = Spam|x) = 0.83$→do we move the mail to Junk?
  - What is more dangerous: deleting 1 non-spam mail or letting 1 spam mail enter Inbox?
- →**Loss function** or **Loss matrix**

# Loss matrix

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

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

- Example 1: 0/1-loss

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

- Example 2: Spam

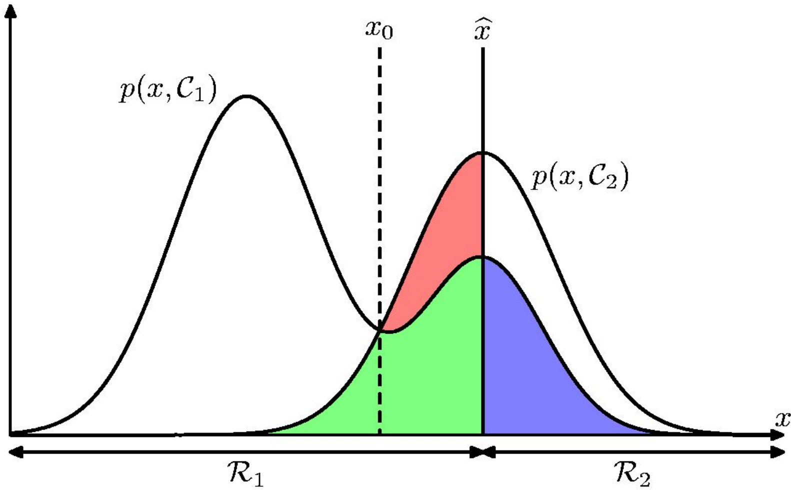
$$L = \begin{pmatrix} 0 & 100 \\ 1 & 0 \end{pmatrix}$$

# Loss and decision

- Expected loss minimization
  - $R_j$ : classify to $C_j$

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

- **Choose such $R_j$ that $EL$ is minimized**
- Two classes

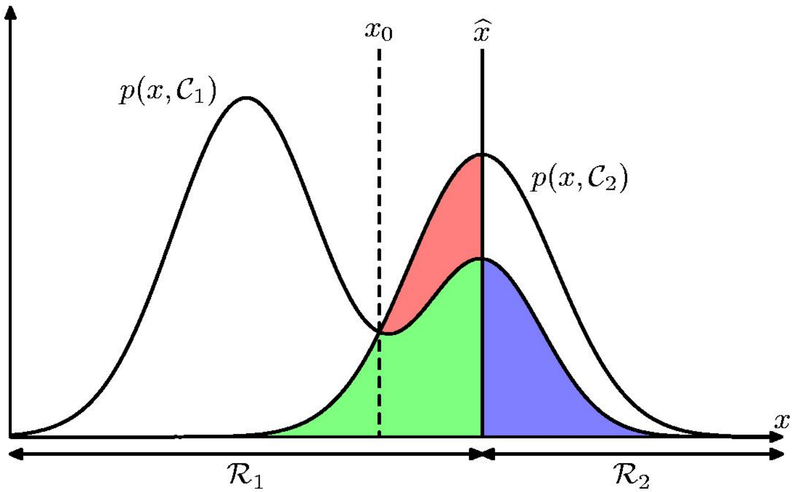



$$EL = \int_{R_1} L_{21} p(x, C_2) dx + \int_{R_2} L_{12} p(x, C_1)\, dx$$

# Loss and decision

- Loss minimization

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




When loss is

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

Classify $Y$ as

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

# Loss and decision

- How to minimize *EL with two classes?*

- Rule:
  - $L_{12}p(x, C_1) > L_{21}p(x, C_2)$ →predict $y$ as $C_1$

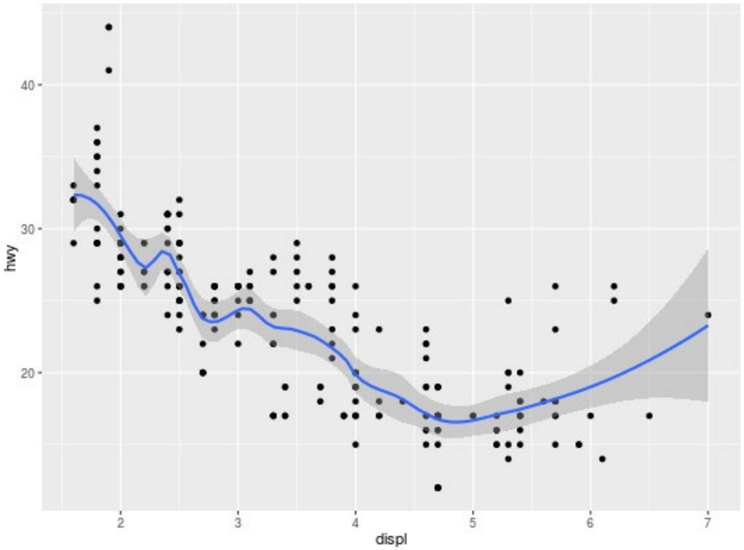
- 0/1 Loss: classify to the class which is more probable!

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

# Loss and decision

- Continuous targets: squared loss
  - Given a model $p(x, y)$, minimize

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




- 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 thershold $\hat{x} = \frac{L_{21}}{L_{12}}$ affects prediction→ what if we don't know the loss? Which classifier is better?
- **Confusion matrix**

| | PREDICTED | | | |
|---|---|---|---|---|
| **TRUE** | | 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**

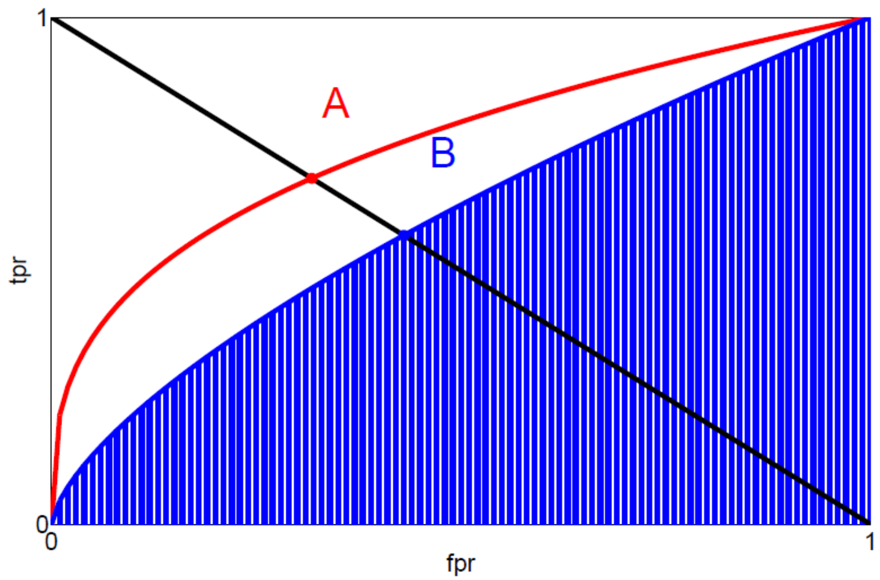
$$Specificity = 1 - FPR$$

- **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

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

# Generative vs Discriminative

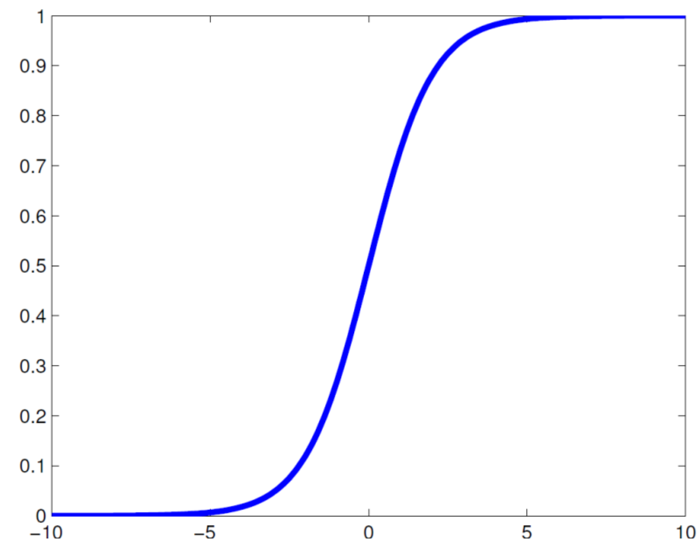
- Generative can be used to generate new data
- Generative normally easier to fit (check Logistic vs K-NN)
- Generative: each class estimated separately→ do not need to retrain when a new class added
- Discriminative models: can replace $X$ with $\phi(X)$ (preprocessing), method will still work
  - Not 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
  - $C = \{C_1 = 1, C_2 = 0\}$
    $$p(Y = C_1|X) = sigm(\boldsymbol{w}^T\boldsymbol{x})$$

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

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




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

# Logistic regression

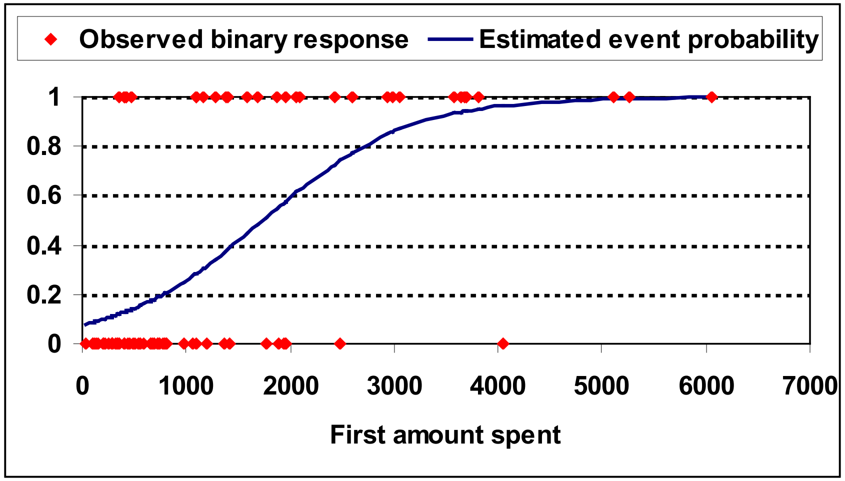
- Logistic model- yet another form

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

**The log of the odds is linear in *x***

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

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



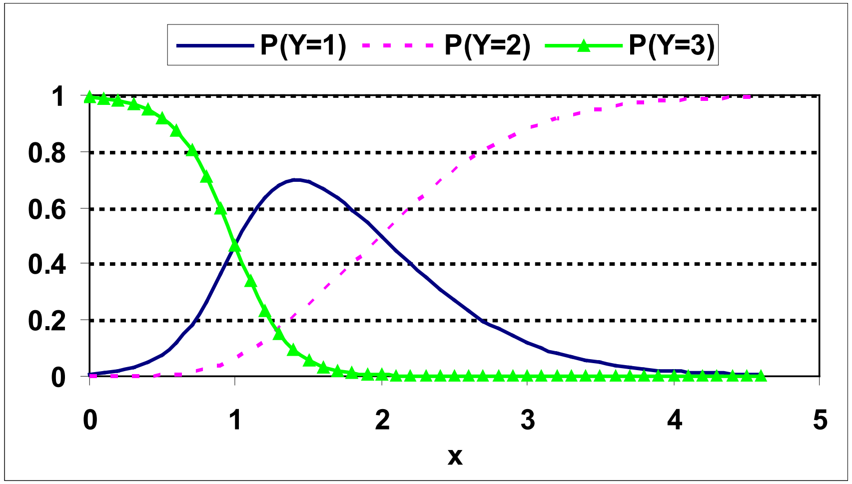
# Logistic regression

- When Y is categorical,

$$p(Y = C_i|x) = \frac{e^{\boldsymbol{w}_i^T \boldsymbol{x}}}{\sum_{j=1}^{K} e^{\boldsymbol{w}_j^T \boldsymbol{x}}} = softmax(\boldsymbol{w}_i^T \boldsymbol{x})$$

- Alternatively

$$Y \sim Multinoulli\left(softmax(\boldsymbol{w}_1^T \boldsymbol{x}), \ldots softmax(\boldsymbol{w}_K^T \boldsymbol{x})\right)$$



# Logistic regression

## Fitting logistic regression

- In binary case,

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

  - 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 $\max_i p_i$

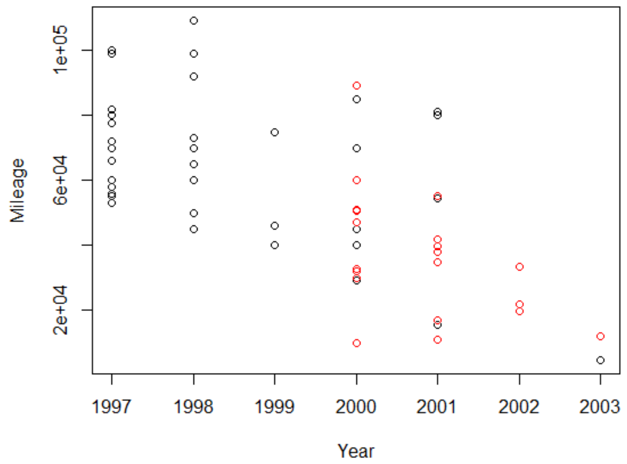
Decision boundaries of logistic regression are linear

# Logistic regression

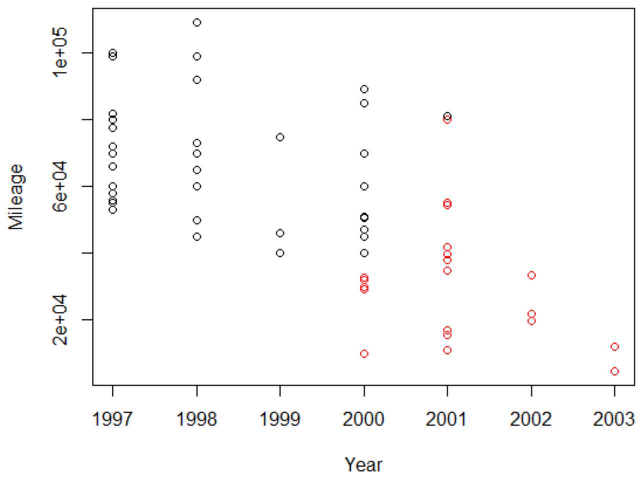
- In R, use glm() with family=”binomial” for two classes
- In R, use multinom () from **nnet** package for more than two classes

Example Equipment=f(Year, mileage)

**Original data**

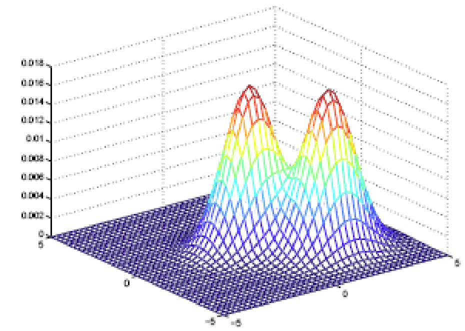



**Classified data**



# Quadratic discriminant analysis

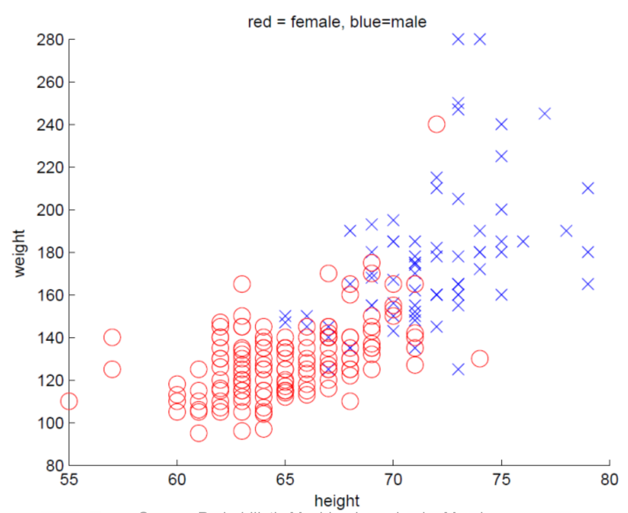
- Generative classifier
- Main assumptions:
  - $\boldsymbol{x}$ is now **random** as well as $y$



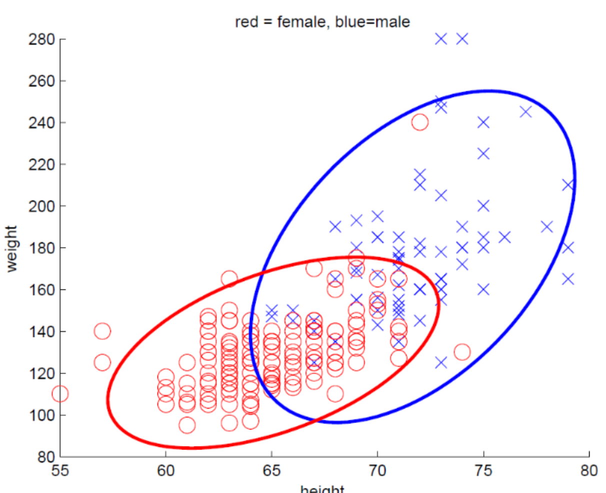
$$p(\boldsymbol{x}|y = C_i, \theta) = N(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$$

$$p(y = C_i) = \pi_i$$

Unknown parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}\}, \boldsymbol{\pi}$




Source: Probabilistic Machine Learning by Murphy

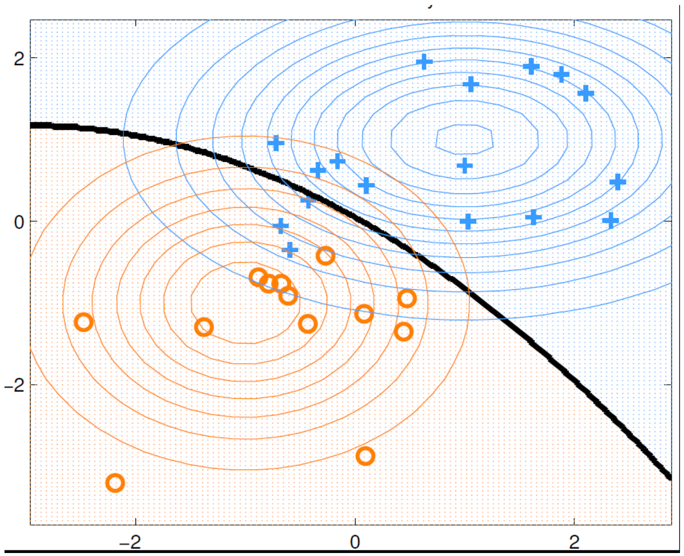



Source: Probabilistic Machine Learning by Murphy

# Quadratic discriminant analysis

- If parameters are estimated, classify:

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




Source: Probabilistic Machine Learning by Murphy

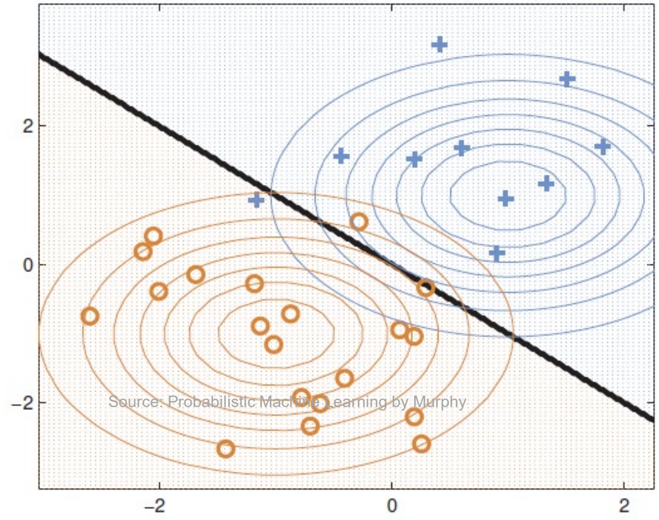
# Linear discriminant analysis (LDA)

- Assumtion $\Sigma_i = \Sigma, i = 1, \dots K$
- **Probabilistic model**

$$\boldsymbol{x}|y = C_i, \mu_i, \Sigma \quad \sim N(\mu_i, \Sigma)$$
$$y|\pi \sim Multinomial(\pi_1, \dots \pi_K\ )$$

- Then $p(y = c_i|x) = softmax(w_i^T x + w_{0i})$ →exactly the same form as the logistic regression
  - $w_{0i} = -\frac{1}{2}\boldsymbol{\mu}_i^T \Sigma^{-1} \boldsymbol{\mu}_i + \log \pi_i$
  - $w_i = \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i$
- 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.

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

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

  - 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 probabiliies
- 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
```

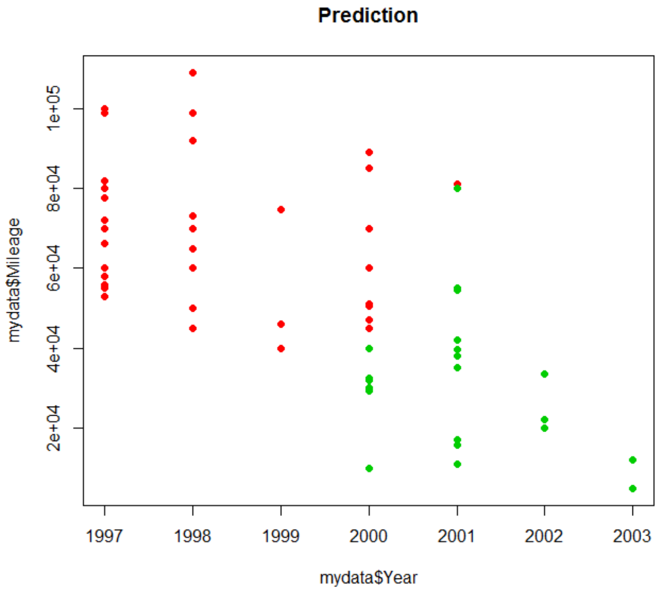
# LDA: output

- Misclassified items

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

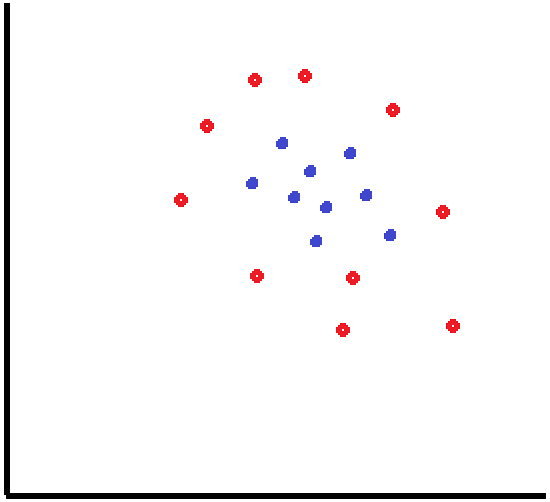
```
> table(Pred$class, mydata$Equipment)

    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.