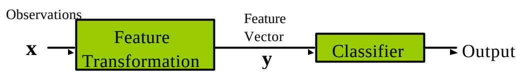
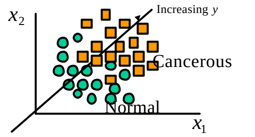
Classification (II)

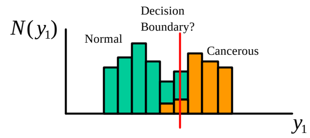


- Feature Transformation
  - Generates features **y** from **x**
  - **y** usually lower dimension than **x**
- Classifier
  - Partitions feature space into different regions

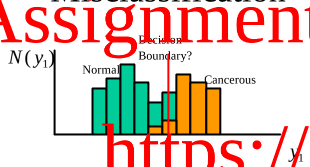
Example

$y_1 = x_1 + x_2$



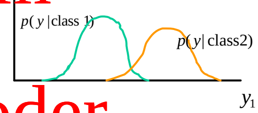


Misclassification



- Impossible to completely separate classes
- Some will always be misclassified
- Good classifier will make fewest mistakes
- Need probability theory to analyse performance

Statistical Decisions



Decision Rules

- Best decision rule should make fewest mistakes
- Need to quantify probability of error
- *Optimal* decision rule is one which minimises the probability of error

Likelihood Ratio Test

- Classify **y** by choosing the class, c_i which has the highest conditional probability, $P(c_i\,|\,\mathbf{y})$

$$P(c_i\,|\,\mathbf{y}) = \frac{p(\mathbf{y}\,|\,c_i)P(c_i)}{p(\mathbf{y})}$$

Likelihood Ratio Test

- For two classes we have:
  - Choose class 1 if

$$\frac{p(\mathbf{y}\,|\,c_1)P(c_1)}{p(\mathbf{y})} > \frac{p(\mathbf{y}\,|\,c_2)P(c_2)}{p(\mathbf{y})}$$

  - Choose class 1 if $L(\mathbf{y}) = \frac{p(\mathbf{y}\,|\,c_1)}{p(\mathbf{y}\,|\,c_2)} > \frac{P(c_2)}{P(c_1)}$

  where L(y) is called the *likelihood ratio*

Example

- Suppose we wish to decide if a cell is cancerous by measuring how red (r) it is.
  - Cancerous cells have

$$p(r\,|\,c) = \frac{1}{\sqrt{2p}}\exp(-0.5(r-5)^2)$$

  - Normal cells have

$$p(r\,|\,n) = \frac{1}{\sqrt{2p}}\exp(-0.5(r-3)^2)$$

  - If cancerous cells and normal cells are equally likely, what is the best classification of a cell with redness r?

Example (cont)

- The likelihood ratio is $L(r) = \frac{\exp(-0.5(r-5)^2)}{\exp(-0.5(r-3)^2)}$
- Cell is cancerous if $L(r) > \frac{P(n)}{P(c)} = 1$

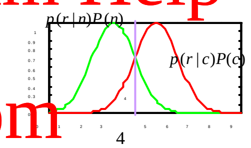
$$\frac{\exp(-0.5(r-5)^2)}{\exp(-0.5(r-3)^2)} > 1$$

$$-0.5(r-5)^2 + 0.5(r-3)^2 > 0$$

$$4r - 16 > 0$$

$$\boxed{r > 4}$$

Example



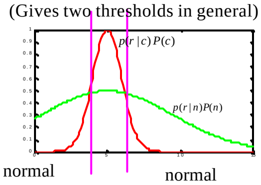
- This is a special case (both distributions have equal variance)

Differing variances

$$-\frac{0.5(r-m_n)^2}{s_n^2} + \frac{0.5(r-m_c)^2}{s_c^2} > 0$$

$$ar^2 + br + c > 0$$

(Gives two thresholds in general)



1D Classifier

- Given examples $\{a_i\}$ from class A, $\{b_i\}$ from class B.
- Estimate distributions *p(x|A), p(x|B)*
  - For normal pdf, compute mean and covariance
- Select priors *P(A),P(B).*
- To classify new example x:
- Select class A if *p(x|A)p(A)>p(x|B)p(B)*

Modifying the threshold

Choose class 1 if $L(\mathbf{y}) = \frac{p(\mathbf{y} \mid A)}{p(\mathbf{y} \mid B)} > t$

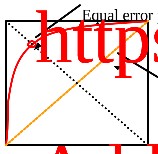
- If $t=P(B)/P(A)$ we make fewest errors
- If t< $P(B)/P(A)$ we classify more A correctly, but make more mistakes on B
- If t> $P(B)/P(A)$ we classify more B correctly, but make more mistakes on A

ROC Curves

- "Receiver Operating Characteristic"
- Summarises performance of classifier as threshold is changed
- Plot true positives (A's correctly classified) against false positives (B's misclassified as A) for different thresholds.
- Allow choice of threshold to achieve particular error performance

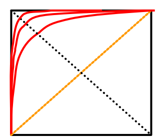
ROC Curves

- True positives (A's correctly classified)
- False positives (B's misclassified as A)



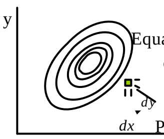
ROC Curves

- Closer curve is to top-left, the better



Multi-variate Distributions

- PDFs extend to n dimensions
- In 1D $P([x, x+dx]) \approx p(x)dx$
- In 2D $P([x, x+dx][y, y+dy]) \approx p(x,y)dxdy$



Multivariate Normal PDF

- In n dimensions, the normal distribution with mean $\mathbf{m}$ and covariance $\mathbf{S}$ has pdf:

$$p(\mathbf{x}:\mathbf{m},\mathbf{S}) = c\exp(-0.5M)$$

$$M = (\mathbf{x}-\mathbf{m})^T \mathbf{S}^{-1} (\mathbf{x}-\mathbf{m}) \qquad c = \frac{1}{(2p)^{n/2} |\mathbf{S}|^{1/2}}$$

- The covariance of N samples is

$$S = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$$

Quadratic Classifiers

- Suppose we have training vectors from several different classes
- For each class, compute the mean and covariance to generate a normal distribution, $p(\mathbf{x} \mid c_i)$
- To classify a new vector, choose the class which maximises $p(\mathbf{x} \mid c_i)P(c_i)$

Nearest Neighbour Classifiers

- Useful *non-linear* classifier
- Retain all training set
- Select class of new example as that of \`closest\` vector in training set
- Require a distance metric $d(\mathbf{x}_1, \mathbf{x}_2)$
- Common metric is Euclidean distance,

$$d(\mathbf{x}_1, \mathbf{x}_2) = |\mathbf{x}_1 - \mathbf{x}_2|^2$$

k-NN Classifier

- Rather than choose single closest,
- Find *k* closest. (*k* odd)
  - If k_A from class A and k_B from class B
  - Choose class A if $k_A > k_B$
- More robust than single nearest neighbour

Support Vector Machines

- A powerful new type of (2-class) classifier
- Designed to minimise expected error over an *unseen* test set:
  - "Structural Risk Minimisation"
  - Avoids problems of overspecialisation on training set
- Particularly useful for small training sets

Example of (Non-Linear) SVM Classification Space