Pattern Recognition applied to Biomedical Signals

Dr Philip de Chazal
Chief Technical Officer
BiancaMed, Dublin, Ireland
Outline

1st Hour: Focus on Pattern Recognition basics
1. Pattern Recognition Overview
2. Classifiers: LDA, QDA, FFNN, HMM etc
3. Performance Assessment
   - Data splitting
   - Performance measures (sensitivity, specificity etc)
4. Features
   - Transformations
   - Missing values
   - Feature Selection
Outline

2nd hour: Case study on Sleep apnea detection from the Electrocardiogram
- Database
- Expert Annotations
- Features
- Performance assessment
- Practical tips
1. Pattern Recognition Overview

Pattern Recognition

- Supervised Learning
  - Parametric
    - Plug-in parameters
      - Discriminant Analysis
      - Neural Networks (feed forward etc)
      - HMM
  - Distributed parameters
    - Non-Parametric
      - Nearest neighbour (kNN)
      - Decision Trees
      - Bayesian Classifiers

- Unsupervised learning
  - Self organising feature maps
    - Cluster Analysis
    - Hebbian Learning
    - Vector Quantisation
Classifier Types: Parametric models with plug-in parameters

- Define a parametric model between the input features and the output classes
- The model has adjustable parameters which are set using training data
2. Classification Methods

- Bayes theorem
- Gaussian models
- Linear discriminant analysis
  - Derivation
  - Covariance inversion
  - Training Equations
  - Classifying Equations
- Quadratic discriminant analysis
- Feedforward neural networks
Bayes theorem

\[ p(x|C_1)P(C_1) \quad p(x|C_2)P(C_2) \]
Bayes theorem

- Optimally classify an object into one of \( c \) mutually exclusive classes given *priors* and *class densities*.

- For a \( c \)-class problem Bayes’ rule states that the posterior probability of the \( k \)th class is related to its prior probability and its class density function by

\[
p_k = \frac{\pi_k f_k(x, \theta_k)}{\sum_{l=1}^{c} \pi_l f_l(x, \theta_l)}
\]
Likelihood function

If we have \( N = \sum_{n=1}^{c} N_k \) labelled data \((d \times l)\) vectors, \(x_n^{(k)}, n = 1..N_k\), then a likelihood can be formed as follows:

\[
I(\theta) = \prod_{k=1}^{c} \prod_{n=1}^{N_k} \pi_k f_k(x_n^{(k)}, \theta_k)
\]

Think of it as combined probabilities

Our aim is to find the values of \( \theta \) for each class that maximise the value of the \( I(\theta) \) likelihood. Equivalently we can find the values of \( \theta \) that maximise the value of the log-likelihood:

\[
L(\theta) = \log(I(\theta)) = \sum_{k=1}^{c} \sum_{n=1}^{N_k} \log(\pi_k f_k(x_n^{(k)}, \theta_k))
= \sum_{k=1}^{c} \sum_{n=1}^{N_k} \log(f_k(x_n^{(k)}, \theta_k)) + \sum_{k=1}^{c} N_k \log(\pi_k)
\]
LDA: Gaussian parametric model for the class densities

- The class densities are modelled with a Gaussian model (\(d\)-dimensional) with common covariance across all classes:

\[
f_k(x, \theta_k = \mu_k, \Sigma) = (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} (x - \mu_k)^T \Sigma^{-1} (x - \mu_k) \right]
\]
LDA: Log-likelihood

For a training example
\[ \log \left( f_k \left( x_n^{(k)}, \mu_k, \Sigma \right) \right) = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log(|\Sigma|) - \frac{1}{2} \left( x_n^{(k)} - \mu_k \right)^T \Sigma^{-1} \left( x_n^{(k)} - \mu_k \right) \]

Hence
\[ \sum_{k=1}^{c} \sum_{n=1}^{N_k} \log \left( f_k \left( x_n^{(k)}, \theta_k \right) \right) = -\frac{dN}{2} \log(2\pi) - \frac{N}{2} \log(|\Sigma|) - \frac{1}{2} \sum_{k=1}^{c} \sum_{n=1}^{N_k} \left( x_n^{(k)} - \mu_k \right)^T \Sigma^{-1} \left( x_n^{(k)} - \mu_k \right) \]

And the log-likelihood over all training examples
\[ L \left( \theta = \mu_1, \ldots, \mu_c, \Sigma \right) = -\frac{dN}{2} \log(2\pi) - \frac{N}{2} \log(|\Sigma|) - \frac{1}{2} \sum_{k=1}^{c} \sum_{n=1}^{N_k} \left( x_n^{(k)} - \mu_k \right)^T \Sigma^{-1} \left( x_n^{(k)} - \mu_k \right) + \sum_{k=1}^{c} N_k \log(\pi_k) \]
LDA: Maximising the log-likelihood function

1. \[
\frac{\partial L}{\partial \mu_k} = \sum_{n=1}^{N_k} \left( x_n^{(k)} - \mu_k \right) = 0
\]
\[
\mu_k = \frac{1}{N_k} \sum_{n=1}^{N_k} x_n^{(k)}
\]

2. \[
\frac{\partial L}{\partial \Sigma} = \sum_{k=1}^{c} \sum_{n=1}^{N_k} \left( x_n^{(k)} - \mu_k \right) \left( x_n^{(k)} - \mu_k \right)^T
\]
\[
\Sigma = \sum_{k=1}^{c} \sum_{n=1}^{N_k} \left( x_n^{(k)} - \mu_k \right) \left( x_n^{(k)} - \mu_k \right)^T / N
\]
QDA: Separate Gaussian parametric model for the class densities

- The class densities are modelled with a Gaussian model \((d\)-dimensional\) with separate covariance across all classes:

\[
f_k(x, \theta_k = \mu_k, \Sigma^{(k)}) = \left(\frac{1}{2\pi}\right)^{\frac{d}{2}} |\Sigma^{(k)}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(x - \mu_k)^T \Sigma^{(k)^{-1}} (x - \mu_k)\right]
\]
QDA: Log-likelihood

For a training example

\[
\log(f_k(x_n^{(k)}, \mu_k, \Sigma^{(k)})) = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log(|\Sigma^{(k)}|) - \frac{1}{2} (x_n^{(k)} - \mu_k)^T \Sigma^{(k)-1} (x_n^{(k)} - \mu_k)
\]

Hence

\[
\sum_{k=1}^{c} \sum_{n=1}^{N_k} \log(f_k(x_n^{(k)}, \theta_k)) = -\frac{dN}{2} \log(2\pi) - \frac{1}{2} \sum_{k=1}^{c} N_k \log(|\Sigma^{(k)}|) - \frac{1}{2} \sum_{k=1}^{c} \sum_{n=1}^{N_k} (x_n^{(k)} - \mu_k)^T \Sigma^{(k)-1} (x_n^{(k)} - \mu_k)
\]

And the log-likelihood over all training examples

\[
L(\theta = \mu_1, \ldots, \mu_c, \Sigma^{(k)}) = -\frac{dN}{2} \log(2\pi) - \frac{1}{2} \sum_{k=1}^{c} N_k \log(|\Sigma^{(k)}|) - \frac{1}{2} \sum_{k=1}^{c} \sum_{n=1}^{N_k} (x_n^{(k)} - \mu_k)^T \Sigma^{(k)-1} (x_n^{(k)} - \mu_k) + \sum_{k=1}^{c} N_k \log(\pi_k)
\]
QDA: Maximising the log-likelihood function

1. \[
\frac{\partial L}{\partial \mu_k} = [\text{many steps omitted!}] = \Sigma_k^{-1} \left( \sum_{n=1}^{N_k} x_n^{(k)} - \mu_k \right) N_k = 0
\]
   \[
   \mu_k = \frac{\sum_{n=1}^{N_k} x_n^{(k)}}{N_k}
   \]

2. \[
\frac{\partial L}{\partial \Sigma_k^{-1}} = [\text{many steps omitted!}] = H \times \left( N_k \Sigma_k - \sum_{n=1}^{N_k} (x_n^{(k)} - \mu_k) (x_n^{(k)} - \mu_k)^T \right)
\]
   \[
   H = \begin{bmatrix}
   0.5 & 1 & \cdots & 1 \\
   1 & 0.5 & \vdots \\
   \vdots & \ddots & \ddots & 1 \\
   1 & \cdots & 1 & 0.5
   \end{bmatrix}
   \]
   \[
   \Sigma_k = \sum_{n=1}^{N_k} (x_n^{(k)} - \mu_k) (x_n^{(k)} - \mu_k)^T / N_k
   \]
Covariance inversion

- If a covariance matrix does not have full rank then cannot invert matrix
- Work around
  - Identify columns of CV matrix with zero eigenvectors
  - Remove these columns for CV (equivalent to remove corresponding features)
  - Invert submatrix
Training Equations

Define an associated \((c \times I)\) target vector \(t_n\) which has one element set to 1 and all other elements set to zero for each of the \(N (d \times I)\) training feature vector \(x_n\). The position of the element with value 1 indicates the class e.g. for a four class problem a target vector indicating that the associated training feature vector belongs to class 3 is

\[
t = [0 \ 0 \ 1 \ 0]^T.
\]

Form a \((d \times N)\) matrix of feature vectors,

\[
X = \begin{bmatrix} x_1 & x_2 & \ldots & x_N \end{bmatrix}
\]

and a \((c \times N)\) matrix of target vectors

\[
T = \begin{bmatrix} t_1 & t_2 & \ldots & t_N \end{bmatrix}
\]

and a \((d \times c)\) matrix of mean vectors

\[
M = \begin{bmatrix} \mu_1 & \mu_2 & \ldots & \mu_c \end{bmatrix}
\]

and a \((c \times I)\) vector of prior probabilities

\[
\pi = \begin{bmatrix} \pi_1 & \pi_2 & \ldots & \pi_c \end{bmatrix}^T
\]
Matlab implementation – Target vectors

<table>
<thead>
<tr>
<th>Vector elements</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
<th>Class 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t_3$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t_4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$t_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Training Equations

\[ M = X T^T \left( T T^T \right)^{-1} \]

\[ \Sigma = X \left( X - M T \right)^T / N \quad \text{LDA} \]

\[ \Sigma^{(k)} = X \left( \left( X \times 1_{[d,1]} T_k. \right) - \mu_k T_k. \right)^T / N_k \quad \text{QDA} \]

\[ T_k. \text{ means the kth row of matrix } T \]
Processing – general form

\[ p_k = \frac{\pi_k f_k(x, \theta_k)}{\sum_{l=1}^{c} \pi_l f_l(x, \theta_l)} = \frac{\exp(K)\pi_k f_k(x, \theta_k)}{\exp(K)\sum_{l=1}^{c} \pi_l f_l(x, \theta_l)} \]

\[ = \frac{\exp(K)\exp(\log(\pi_k f_k(x, \theta_k)))}{\sum_{l=1}^{c} \exp(K)\exp(\log(\pi_l f_l(x, \theta_l)))} = \frac{\exp(\log(\pi_k f_k(x, \theta_k)) + K)}{\sum_{l=1}^{c} \exp(\log(\pi_l f_l(x, \theta_l)) + K)} \]

\[ = \frac{\exp(y_k)}{\sum_{l=1}^{c} \exp(y_l)}, \text{ where } y_k = \log(\pi_k f_k(x, \theta_k)) + K \]

Now \[ \sum_{k=1}^{c} p_k = \sum_{k=1}^{c} \frac{\pi_k f_k(x, \theta_k)}{\sum_{l=1}^{c} \pi_l f_l(x, \theta_l)} = \frac{\sum_{k=1}^{c} \pi_k f_k(x, \theta_k)}{\sum_{l=1}^{c} \pi_l f_l(x, \theta_l)} = 1 \]

\[ \therefore p_1 = 1 - \sum_{l=2}^{c} p_l, \quad p_k = \frac{\exp(y_k)}{\sum_{l=1}^{c} \exp(y_l)} \]
Processing - LDA

\[
\log \left( \pi_k f_k (x, \theta_k) \right) = \log (\pi_k) - \frac{d}{2} \log (2\pi) - \frac{1}{2} \log (|\Sigma|) - \frac{1}{2} (x - \mu_k)^T \Sigma^{-1} (x - \mu_k)
\]

\[
= \log (\pi_k) - \frac{1}{2} \left( x^T \Sigma^{-1} x - 2 \mu_k^T \Sigma^{-1} x + \mu_k^T \Sigma^{-1} \mu_k \right) - \frac{d}{2} \log (2\pi) - \frac{1}{2} \log (|\Sigma|)
\]

\[
= \log (\pi_k) + \mu_k^T \Sigma^{-1} x - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + K, \quad \text{where } K = -\frac{d}{2} \log (2\pi) - \frac{1}{2} \log (|\Sigma|) + x^T \Sigma^{-1} x
\]

\[
y_k = \log (\pi_k) + \mu_k^T \Sigma^{-1} x - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k
\]

\[
y_k = \log (\pi_k) + a_k x + b_k \quad \text{ Linear equation}
\]

\[
a_k = \mu_k^T \Sigma^{-1}
\]

\[
b_k = -\frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k
\]
Processing - QDA

\[
\log\left( \pi_k f_k(x, \theta_k) \right) = \log(\pi_k) - \frac{d}{2} \log(2\pi) - \frac{1}{2} \log(|\Sigma_k|) - \frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)
\]

\[
= \log(\pi_k) - \frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) - \frac{d}{2} \log(2\pi) - \frac{1}{2} \log(|\Sigma_k|)
\]

\[
= \log(\pi_k) - \frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) - \frac{1}{2} \log(|\Sigma_k|) - K, \text{ where } K = \frac{d}{2} \log(2\pi)
\]

\[
y_k = 2 \log(\pi_k) - (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) - \log(|\Sigma_k|)
\]

Quadratic equation
Feedforward Neural Networks

- Multilayer perceptron artificial neural network
- 0 or more hidden layers
Feedforward Neural Networks

- Flexible linear or nonlinear mapping from features to classes
- A feedforward neural network is a ‘universal function approximator’
- Except for linear networks, training requires numerical optimisation
- Back propagation algorithm used for efficient training

\[
F: \mathbb{R}^d \rightarrow \mathbb{R}^n
\]

\[
y_m^{(1)} = \phi^{(1)} \left( b_m^{(1)} + \sum_{i=1}^{d} w_{mi} x_i \right)
\]
3. Performance Measurement Methods

- **Twoway classification**
  - Prior probabilities
  - Sensitivity
  - Specificity
  - Positive and negative predictivity
  - Accuracy
- **Multiway classification**
  - Prior probabilities
  - Sensitivity
  - Specificity
  - Positive and negative predictivity
  - Accuracy
- **Data splitting**
# Two classification

<table>
<thead>
<tr>
<th>Diagnostic Allocation</th>
<th>True status</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Abnormal ($D$)</td>
<td>Normal ($\sim D$)</td>
<td>Total</td>
<td></td>
</tr>
<tr>
<td>Abnormal ($S$)</td>
<td>$a$</td>
<td>$b$</td>
<td></td>
<td>$a+b$</td>
</tr>
<tr>
<td>Normal ($\sim S$)</td>
<td>$c$</td>
<td>$d$</td>
<td></td>
<td>$c+d$</td>
</tr>
<tr>
<td>Total</td>
<td>$a+c$</td>
<td>$b+d$</td>
<td></td>
<td>$N$</td>
</tr>
</tbody>
</table>

$a$ is the number of cases which were classified abnormal and were truly abnormal.  
$b$ is the number of cases which were classified abnormal but were in fact normal.  
$c$ is the number of cases which were classified normal but were in fact abnormal.  
$d$ is the number of cases which were classified normal and were truly normal.  
$N$ is the total number of cases.

Probability of having the disease ($P$) = \[
\frac{a + c}{N} 
\]

Probability of not having the disease = \[
\frac{b + d}{N} = 1 - P
\]
Two way classification

<table>
<thead>
<tr>
<th>Diagnostic Allocation</th>
<th>True status</th>
<th>Abnormal (D)</th>
<th>Normal (~D)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abnormal (S)</td>
<td>a</td>
<td>b</td>
<td>a+b</td>
<td></td>
</tr>
<tr>
<td>Normal (~S)</td>
<td>c</td>
<td>d</td>
<td>c+d</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>a+c</td>
<td>b+d</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

Sensitivity (Se) = \( p(S|D) = \text{Percentage of well classified abnormalities} = \frac{a}{a+c} \)

Specificity (Sp) = \( p(\sim S|\sim D) = \text{Percentage of well classified normals} = \frac{d}{b+d} \)

Accuracy (A) = \( \text{Percentage of well classified cases} = \frac{a+d}{N} \)
Two way classification

<table>
<thead>
<tr>
<th>Diagnostic Allocation</th>
<th>True status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Abnormal (D)</td>
</tr>
<tr>
<td>Abnormal (S)</td>
<td>a</td>
</tr>
<tr>
<td>Normal (~S)</td>
<td>c</td>
</tr>
<tr>
<td>Total</td>
<td>a+c</td>
</tr>
</tbody>
</table>

Predictive value of positive test (PV+) = \( p(D|S) \)

\[
= \frac{Percentage \ of \ well \ classified \ positives = \frac{a}{a+b}}{P. Se + (1-P).(1-Sp)}
\]

Predictive value of a negative test (PV-) = \( p(\sim D|\sim S) \)

\[
= \frac{Percentage \ of \ well \ classified \ negatives = \frac{d}{c+d}}{P. (1-Se) + (1-P).Sp}
\]
## Multiway classification

<table>
<thead>
<tr>
<th>True Status</th>
<th>No disease</th>
<th>Disease 1</th>
<th>Disease 2</th>
<th>...</th>
<th>Disease n</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>No disease</td>
<td>N₀₀</td>
<td>N₀₁</td>
<td>N₀₂</td>
<td></td>
<td>N₀ₙ</td>
<td>N₀</td>
</tr>
<tr>
<td>Disease 1</td>
<td>N₁₀</td>
<td>N₀₀</td>
<td>N₀₀</td>
<td></td>
<td>N₁ₙ</td>
<td>N₁</td>
</tr>
<tr>
<td>Disease 2</td>
<td>N₂₀</td>
<td>N₀₀</td>
<td>N₀₀</td>
<td></td>
<td>N₂ₙ</td>
<td>N₂</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disease n</td>
<td>Nᵢ₀</td>
<td>Nᵢ₁</td>
<td>Nᵢ₂</td>
<td></td>
<td>Nᵢₙ</td>
<td>Nᵢ</td>
</tr>
<tr>
<td>Sum</td>
<td>N₀</td>
<td>N₁</td>
<td>N₂</td>
<td></td>
<td>Nᵢ</td>
<td>Nᵢ</td>
</tr>
</tbody>
</table>

Prevalence of disease i = probability of having disease i: \( P_i = \frac{N_i}{N_{..}} \)
# Multiway classification

<table>
<thead>
<tr>
<th>True Status</th>
<th>No disease</th>
<th>Disease 1</th>
<th>Disease 2</th>
<th>...</th>
<th>Disease n</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>No disease</td>
<td>N_{00}</td>
<td>N_{01}</td>
<td>N_{02}</td>
<td></td>
<td>N_{0n}</td>
<td>N_0</td>
</tr>
<tr>
<td>Disease 1</td>
<td>N_{10}</td>
<td>N_{00}</td>
<td>N_{00}</td>
<td></td>
<td>N_{1n}</td>
<td>N_1</td>
</tr>
<tr>
<td>Disease 2</td>
<td>N_{20}</td>
<td>N_{00}</td>
<td>N_{00}</td>
<td></td>
<td>N_{2n}</td>
<td>N_2</td>
</tr>
<tr>
<td>:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disease n</td>
<td>N_{n0}</td>
<td>N_{11}</td>
<td>N_{12}</td>
<td></td>
<td>N_{nn}</td>
<td>N_n</td>
</tr>
<tr>
<td>Sum</td>
<td>N_0</td>
<td>N_1</td>
<td>N_2</td>
<td></td>
<td>N_n</td>
<td>N_..</td>
</tr>
</tbody>
</table>

Sensitivity for disease $i = \frac{N_{ii}}{N_i}$:

Specificity = Proportion of correctly classified normals $= \frac{N_{00}}{N_0}$.

Accuracy (A) = Proportion of correctly classified cases $= \frac{\sum N_{ii}}{N_..}$.
### Multiway classification

<table>
<thead>
<tr>
<th>True Status</th>
<th>No disease</th>
<th>Disease 1</th>
<th>Disease 2</th>
<th>...</th>
<th>Disease n</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>No disease</td>
<td>N_00</td>
<td>N_01</td>
<td>N_02</td>
<td></td>
<td>N_0n</td>
<td>N_0.</td>
</tr>
<tr>
<td>Disease 1</td>
<td>N_10</td>
<td>N_00</td>
<td>N_00</td>
<td></td>
<td>N_1n</td>
<td>N_1.</td>
</tr>
<tr>
<td>Disease 2</td>
<td>N_20</td>
<td>N_00</td>
<td>N_00</td>
<td></td>
<td>N_2n</td>
<td>N_2.</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disease n</td>
<td>N_n0</td>
<td>N_11</td>
<td>N_12</td>
<td></td>
<td>N_nn</td>
<td>N_n.</td>
</tr>
<tr>
<td>Sum</td>
<td>N_0</td>
<td>N_1</td>
<td>N_2</td>
<td></td>
<td>N_n</td>
<td>N_n.</td>
</tr>
</tbody>
</table>

Predictive value for disease $i = \text{proportion of cases classified disease } i \text{ which are correct}$: $PV_i = \frac{N_{ii}}{N_i}$
Data splitting

- **Ideal**
  - Maximum data for training
  - Maximum data for testing
  - Training and test data independent
  - Conflicting requirements

- **Resubstitution**
  - Train and test on same recordings
  - Positively biased results

- **Holdout**
  - Train on one sample, test on remaining sample

- **Cross fold validation**
Illustration of 5-fold cross validation using 10 ECG records. The data is divided into 5 mutually exclusive folds and the classifier is trained and tested 5 times. Each time a different test fold is used and the remainder of the data used for training.

Unbiased, computationally intensive
4. Features

- Transformations
- Missing values
- Feature Selection
Transformations

- Look at the histogram of features and try applying a transformation if a skewed distribution resulting in a less skewed distribution.
Missing Values

- Practical data sets often have missing feature values due to faulty measurements etc
- A majority of classifier models require all feature values to present
- What to do?
  - Delete all cases with one of more feature values
  - Estimate the missing feature values
    - replace with average (bad choice as skews distribution)
    - Replace with random value
    - Replace with value from another case that is “similar” and has all features
  - See academic.uprm.edu/~eacuna/IFCS04r.pdf for a good summary
The aim is to find a subset of the available features that provides “acceptable” performance

- Fewer features means easier implementation
- There may exist subsets of the available features that provide higher classification performance than the full feature set
  - Irrelevant and redundant features in general reduce classifier performance
- Methods to look for include “filter” and “wrapper” methods, forward selection, backward elimination, stepwise, exhaustive, beam search
Part 2:
Case Study: Sleep Apnea Detection using the Electrocardiogram
Case Study: Sleep Apnea Detection using the Electrocardiogram

- Obstructive sleep apnea: 2-4% prevalence, disrupted sleep, treated with Continuous Positive Airway Pressure (CPAP) mask.
- Diagnosis using polysomnogram (multiple signals).
- Diagnostic test costs $1500 - carried out in hospital.
- Only 15% with disease have been diagnosed.
Study Objective

- See if can determine a method that can reliably detect sleep apnea using the Electrocardiogram (ECG)

- Benefits
  - Can do the test at home
  - Low cost
  - Reduce waiting lists in hospitals
Sleep Apnoea ECG database

- *Computers in Cardiology Conference 2000 Challenge*
  - Automated ECG apnoea detection.
- Uses modified lead V2 ECG from PSG database from patients at Philipps University in Germany (T. Penzel) which had been scored by sleep physiologists using complete polysomnogram.
- Supplied raw ECG waveform from a single lead and QRS detection times (unverified).
- 70 records total (about 8 hrs each); 35 released for training, 35 for independent testing
Epoch based scoring

- PSG scored on epoch-by-epoch basis.
- Goal was to mimic human scorer
  - Each epoch labelled as Normal (NR) or Sleep disordered respiration (SDR)
Splitting up of the Data

- We have 70 overnight recordings of ECG
  - Every minute annotated as ‘normal’ of ‘sleep disorder breathing’ by an expert
  - Over 32000 labels
  - 35 recordings available for training, 35 withheld for testing
Bradycardia/tachycardia patterns

- Guilleminault et al. were first to report on characteristic bradycardia/tachycardia pattern associated with obstructive apnoeas (Guilleminault C et al. Cyclical variation of heart rate in sleep apnea syndrome. Lancet 1984).

Brady/tachy patterns

Stein et al.,
J. Cardiovasc
Electrophysiol.,
2003

Figure 1. Full page of a 1-hour tachogram, beginning at 22:16, from a patient with severe sleep apnea. Tachogram has been separated into 10-minute segments for greater clarity. Lights out was at 22:23:24. Arousal from the first obstructive apnea occurs at 22:27:07 and is visible on the tachogram. Each subsequent heart rate arousal is associated with a respiratory event.
EDR signal

- Modulation of chest lead ECG signal amplitude by respiration

EDR(n)=area enclosed by the QRS complex(n)
ECG derived respiration (EDR)


Figure 7. The upper is the EDR waveform; the lower is the respiration reference waveform. The first waves represent breaths done moderately slow. It’s possible to distinguish two different apneas. The first (20s) is preceded by total expiration and the second (33s) is preceded by a deep inspiration. Duration: 200 seconds.

Travaglini et al
## Features

<table>
<thead>
<tr>
<th>RR interval time domain</th>
<th>RR interval frequency domain</th>
<th>EDR frequency domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial correlation</td>
<td>32 PSD features</td>
<td>32 PSD features</td>
</tr>
<tr>
<td>SDNN</td>
<td>Record-based mean RR interval</td>
<td>Record-based mean EDR amplitude</td>
</tr>
<tr>
<td>NN50</td>
<td>Record-based stdev RR interval</td>
<td>Record-based stdev EDR amplitude</td>
</tr>
<tr>
<td>pNN50</td>
<td>Epoch-based mean RR interval</td>
<td>Epoch-based mean EDR amplitude</td>
</tr>
<tr>
<td>Allan Factor at 5-25 secs time scale</td>
<td>Epoch-based st.dev. RR interval</td>
<td>Epoch-based st.dev. EDR amplitude</td>
</tr>
</tbody>
</table>

88 Features in all
Experiments

- Linear and quadratic discriminant classifiers
  - Quick to train
  - Wanted to focus study on the features not the classifiers
  - Wanted a system that is readily implemented on microprocessors
- Different combination of feature groups:
  - RR time domain and frequency domain
  - EDR frequency domain
- Feature Selection
- Covariance regularisation (not discussed here)
Training – all features

- 35-fold cross validation. Each fold contained 1 record
  - Removed intra-record bias
Training – Feature Selection

- Use the best first feature selection strategy
- Outer loop: 35-fold cross validation
- Inner loop: 34 fold cross validation.
- As before each fold contained 1 record
  - Removed intra-record bias
Performance Assessment 1

- During Feature selection classifiers compared using ‘Accuracy’

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Expert</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NR</td>
<td>NR</td>
<td>TP</td>
<td>FN</td>
</tr>
<tr>
<td>SDR</td>
<td>SDR</td>
<td>FP</td>
<td>TN</td>
</tr>
</tbody>
</table>

Accuracy = \( \frac{(TP+TN)}{(TP+FN+FP+TN)} \)
In addition classifiers assessed using Sensitivity and Specificity

<table>
<thead>
<tr>
<th></th>
<th>NR</th>
<th>SDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NR</td>
<td>TP</td>
<td>FN</td>
</tr>
<tr>
<td>SDR</td>
<td>FP</td>
<td>TN</td>
</tr>
</tbody>
</table>

Sensitivity = TP / (TP+FP)

Specificity = TN / (TN+FN)
### Results - cross validation

(a) All features, (b) Feature selection

- LDA better than QDA
- Feature selection improved QDA
- Best features were RR and EDR combined

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Feature Set</th>
<th>Number of Features</th>
<th>Regularisation Value (α)</th>
<th>Average of training set results</th>
<th>Average of testing set results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Accuracy (%)</td>
<td>Sensitivity (%)</td>
</tr>
<tr>
<td>LD</td>
<td>RR</td>
<td>52</td>
<td>0</td>
<td>89.2</td>
<td>83.2</td>
</tr>
<tr>
<td>LD</td>
<td>EDR</td>
<td>36</td>
<td>0</td>
<td>87.7</td>
<td>81.9</td>
</tr>
<tr>
<td>LD</td>
<td>RR+EDR</td>
<td>88</td>
<td>0</td>
<td>92.3</td>
<td>89.6</td>
</tr>
<tr>
<td>QD</td>
<td>RR</td>
<td>52</td>
<td>0</td>
<td>83.0</td>
<td>94.1</td>
</tr>
<tr>
<td>QD</td>
<td>EDR</td>
<td>36</td>
<td>0</td>
<td>89.1</td>
<td>86.2</td>
</tr>
<tr>
<td>QD</td>
<td>RR+EDR</td>
<td>88</td>
<td>0</td>
<td>89.9</td>
<td>94.6</td>
</tr>
<tr>
<td>LD</td>
<td>RR</td>
<td>23.1*</td>
<td>0</td>
<td>89.3</td>
<td>84.2</td>
</tr>
<tr>
<td>LD</td>
<td>EDR</td>
<td>9.7*</td>
<td>0</td>
<td>87.7</td>
<td>82.6</td>
</tr>
<tr>
<td>LD</td>
<td>RR+EDR</td>
<td>27.0*</td>
<td>0</td>
<td>92.2</td>
<td>89.7</td>
</tr>
<tr>
<td>QD</td>
<td>RR</td>
<td>18.4*</td>
<td>0</td>
<td>90.0</td>
<td>90.0</td>
</tr>
<tr>
<td>QD</td>
<td>EDR</td>
<td>13.9*</td>
<td>0</td>
<td>88.8</td>
<td>84.9</td>
</tr>
<tr>
<td>QD</td>
<td>RR+EDR</td>
<td>22.7*</td>
<td>0</td>
<td>93.2</td>
<td>92.8</td>
</tr>
</tbody>
</table>
Results – withheld set

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Feature Set</th>
<th>Optimisation technique</th>
<th>Number of Features</th>
<th>Regularisation Value (α)</th>
<th>Released-set Accuracy (%)</th>
<th>Released-set Sensitivity (%)</th>
<th>Released-set Specificity (%)</th>
<th>Withheld-set Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD</td>
<td>RR+EDR*</td>
<td>Unoptimised</td>
<td>88</td>
<td>0</td>
<td>92.3</td>
<td>89.6</td>
<td>94.0</td>
<td>90.4</td>
</tr>
<tr>
<td>LD</td>
<td>RR+EDR*</td>
<td>Feature selection</td>
<td>31</td>
<td>0</td>
<td>92.2</td>
<td>89.8</td>
<td>93.7</td>
<td>90.5</td>
</tr>
</tbody>
</table>

• Results on new data (withheld set) similar to the cross-validation results which is an encouraging sign!
Results - feature separation across classes

- **RR PSD features**
  - Good separation at low frequencies

- **EDR PSD features**
  - Good separation particularly at low frequencies
Practical tips

- Always start with a simple system and add complexity if needed. E.g., start with LDA, progress to NN if needed.
- Focus on finding good discriminating features. If your features are poor, then no fancy classifier will help.
- As best as possible make sure your performance assessment is unbiased:
  - Be careful of training and testing with features from the same record.
  - Never make many performance assessments on the same data set, then report the best as this will be a positively biased result. Be particularly careful with feature selection where many thousands of comparisons may be made.
- Consider what is the target device for the pattern recognition system:
  - E.g., low power, low computational device will influence your choice of features and classifier.
Bibliography

- **Data splitting**

- **Performance estimating**

- **Classifiers**
  - Mike James, *Classification Algorithms*, John Wiley and Sons, 1985

- **Sleep Apnea**
Email – contact me

- philip.dechazal@biancamed.com
- philip@ee.ucd.ie