Classification
Estadística II

Eugenio Sánchez Úbeda
Introduction
Classification problems

Example

- **Email Spam Filtering** involves **classifying** an email as spam or non-spam based on the email contents
  - **Output**: spam or non-spam
  - **Input variables** or features: information obtained from the email

The filter estimates the output from the inputs (i.e., given an email, it is spam or not)
Classification problems
Iris data

- 150 observations
  - 4 inputs: length and width of petals and sepals
  - Output: type of Iris (setosa, versicolor, virginica)
Classification problems
Iris data

• Estimated output by four different classifiers (using two inputs)
  • For each pair of input values the color represents the estimated output

• There exist many classification techniques

• Accuracy, interpretability and learning complexity of the estimation process are key for selecting the appropriate one
Main types of problems
The classification problem

Many of the concepts that we have encountered in regression transfer over to the classification setting with only some modifications due to the fact that the output is no longer numerical.

Supervised learning

Unsupervised learning

Regression problem

Classification problem

Clustering problem
Classification Approach

- The theoretical model consists of two main terms:

\[ Y = f(X) + \epsilon \]

The qualitative output variable \( Y \) can take on \( K \) possible distinct and unordered values (\( K \) different classes or categories).

Usually in classification the deterministic term has not a mathematical compact form, it includes some logical classification rule.
Classification
Main approaches (idea)

• **Direct approach**
  - Focus on the *conditional expected value* of the output (the average class)
  - Estimates directly the output value

\[
E(Y \mid X = x)
\]

• **Probabilistic approach**
  - Focus on the *conditional probability of each class*
  - Estimate the K probabilities to provide an *enhanced output*

\[
Pr(Y = k \mid X = x)
\]
Classification

Classification trees vs linear discriminant

- Two bidimensional classification illustrative examples

**PROBLEM A:** the true decision boundary is linear

**PROBLEM B:** The true decision boundary is non-linear
Classification problem

Classification vs regression

• The main goal is to estimate, from the input values, the output value (as in regression)

• Now the output is qualitative (categorical)
  • Finite set of possible values
  • Unordered values

New ways of assessing the model accuracy
Classification
Assessing model accuracy

- The most common approach for quantifying the accuracy of the classifier is the **error rate**, the proportion of mistakes that are made if we apply our estimate to the observations.
  - 1 means 100% of misclassification
  - 0 means perfect classification

\[
\frac{1}{n} \sum_{i=1}^{n} I(y_i \neq \hat{y}_i)
\]

Dataset for computing the error rate:
\((x_1, y_1), \ldots, (x_n, y_n)\)

Indicator variable
\(I(\text{true}) = 1\quad I(\text{false}) = 0\)

\(y_i \neq \hat{y}_i \rightarrow I(y_i \neq \hat{y}_i) = 0\)
Classification
Assessing model accuracy

• The classification **confusion matrix** allows us to understand how the classifier performed in each class, the types of mistakes

• Example

```
y = [1 1 1 1 1 1 0 0 0]; %true, target
yest = [1 0 1 1 0 0 1 0 1]; %estimated
plotconfusion(y,yest);
```

<table>
<thead>
<tr>
<th></th>
<th>y</th>
<th>yest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Elements on the **diagonal** of the matrix represent observations whose output were **correctly predicted**, while **off-diagonal** elements represent cases that were **misclassified**

**Confusion Matrix**

```
Confusion Matrix
```

<table>
<thead>
<tr>
<th></th>
<th>Output Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20.0%</td>
</tr>
<tr>
<td>1</td>
<td>10.0%</td>
</tr>
<tr>
<td>2</td>
<td>20.0%</td>
</tr>
<tr>
<td>3</td>
<td>30.0%</td>
</tr>
<tr>
<td>4</td>
<td>40.0%</td>
</tr>
<tr>
<td>5</td>
<td>60.0%</td>
</tr>
<tr>
<td>6</td>
<td>66.7%</td>
</tr>
<tr>
<td>7</td>
<td>33.3%</td>
</tr>
<tr>
<td>8</td>
<td>57.1%</td>
</tr>
<tr>
<td>9</td>
<td>42.9%</td>
</tr>
<tr>
<td>10</td>
<td>60.0%</td>
</tr>
</tbody>
</table>

```

Overall error rate = 40.0%
Model complexity vs generalization error
Main difficulties
Overfitting and oversmoothing

- More complex models (such as classification trees) can lead to a phenomenon known as **overfitting** the data, which essentially means they follow the errors, or noise, too closely.

![Diagram showing oversmoothing and overfitting examples](image)

Oversmoothing is typical when very simple models are used.

Overfitting is typical when very complex models are used.
Main difficulties
Overfitting and oversmoothing

• Example (bidimensional \textit{regression} problem)

\begin{align*}
\text{Years of Education} & \quad \text{Salary} \\
\text{Years of Education} & \quad \text{Salary} \\
\text{Years of Education} & \quad \text{Salary}
\end{align*}

Complexity increases

• Example (\textbf{Classification} problem)

\begin{align*}
\text{Years of Education} & \quad \text{Salary} \\
\text{Years of Education} & \quad \text{Salary} \\
\text{Years of Education} & \quad \text{Salary}
\end{align*}

Complexity increases

\textit{Estadística II}
GITI/GITT

\textit{Clasificación 2017-2018}
Main difficulties

Overfitting and oversmoothing

• Trade-off between the deterministic and the random component of the model

\[ Y = f(X) + \epsilon \]

In the **oversmoothing** case part of the deterministic component is considered as noise.

In the **overfitting** case part of the noise is modeled as deterministic.

It is a matter of estimating the **right** complexity of the model.
Model complexity
Training and test sets

• The test set is used to estimate the future prediction error, consisting of unseen observations not used to train the statistical learning method.

• The test set allows fixing the trade-off between model complexity and training error.
Model complexity
Training and test sets

- Example (regression problem)

The U-shape in the Test MSE curve holds regardless of the particular data set and the statistical method being used.
Model complexity
Training and test sets

• Example (regression problem)

The training MSE decreases monotonically. However, because the truth is close to linear, the test MSE only decreases slightly before increasing again.

The training and test MSE curves still exhibit the same general patterns, but now there is a rapid decrease in both curves before the test MSE starts to increase slowly.

U-shape in the Test MSE holds
Model complexity
Training and test sets

• Regression example
  • Estimate the polynomial (complexity and parameters) from the available dataset

• 200 observations
• There is noise
Model complexity
Training and test sets

- Fit 12 polynomials with increasing complexity between 1 and 12
- Use 80% of the data set as training data and 20% for testing
Model complexity
Training and test sets

- Trade-off between test error and model complexity

Choose quadratic
Model complexity
Training and test sets

- More data allows better estimation of the complexity
Direct approach:
Classification trees
Classification trees

Overview

• Characteristics:
  • Classify classes of the output by splitting the input space
  • The resulting white-box model is hierarchical

• Example:

• Types of nodes: test and terminal

• Types of separators:
  • For categorical input variables: Value of $X$?
  • For continuous input variables: Value of $X < $threshold?

• Non-linear models
• Universal approximators
Classification trees
Partition of the input space

- Iris Problem
  - The hierarchy of the tree imposes the input space partitioning

Splits are orthogonal to the axes
Classification trees
Illustrative example

- 2 classes: inside/outside
- 4 inputs (attributes)
  \((x_1, x_2, x_3, x_4)\)

The fourth input is noise

```c
if (x3>100 OR x3<0)
    y = 0; /* outside */
else{
    if (((x3<10*(10-x1)) AND (x3<(1000-x2)/10))
        y = 1; /* inside */
    else
        y = 0; /* outside */
}
```
Classification trees
Illustrative example: input space

(a) Projection along plane $x_1-x_2$

(b) Projection along plane $x_1-x_3$

(c) Projection along plane $x_3-x_2$

(d) Projection along plane $x_3-x_4$

‘Inside’: blue circles
‘Outside’: black dots
Classification trees

Illustrative example: splits

**Learning set:** data.tr  N=2000 Correct_classif. = 1905 (95.25%)

**Test set:** data.tst N=1000 Correct_classif. = 941 (94.10%)

Algorithm: ID3  Hmin: 0.52  Number_of_nodes= 17
Classification trees
Growing learning algorithm

(a) 
Increase the purity of the nodes

(b) 
Top-down building approach

• Idea:
  • Split the input space recursively until the terminal nodes are pure enough

How to assess the purity of a set?
Classification trees
Diversity

- **Entropy**: standard way of assessing the chaos in the data set

\[
H(\text{LS}(n)) = - \sum_{i=1}^{N_C} p(n, c_i) \log[p(n, c_i)]
\]

where
- \( p(n, c_i) \): proportion of points belonging to class \( c_i \)

Mixture of larger chaos

More ordered classes
Classification trees
Growing learning algorithm

**Key points**

- **Splitting criterion**: selection of the best split
- **Stopping criterion**
Classification trees
Growing Learning algorithm: Splitting criterion

- Select the **best split** $T$ that produces the **largest decrease of the chaos** in the $LS(n)$

$$Index(n, T) = H(LS(n)) - \sum_{s=1}^{N_s} p(n_s)H(LS(n_s))$$

Types of splits ($T$):

- $X$ is **categorical**: $T = \text{value } X$?
- $X$ is **continuous**: $T = \text{value } X < \text{threshold}$?
Classification trees
Growing learning algorithm: Splitting criterion

Data set and candidate splits

Evaluation of split S2

24 observations (13 of class c1 and 11 of class c2)

Entropy of the original data set

$H(LS(n)) = -\sum_{i=1}^{N_C} p(n,c_i) \log[p(n,c_i)]$

$\displaystyle I_{ID3}(n,T) = H(LS(n)) - \sum_{s=1}^{N_S} p(n_s)H(LS(n_s))$

Node C

$p(n1) = 14/24$
$p(n2) = 10/24$
$p(n1, c1) = 11/14$
$p(n1, c2) = 3/14$
$p(n2, c1) = 2/10$
$p(n2, c2) = 8/10$
Classification trees
Growing learning algorithm: **Stopping criterion**

- Control the number of nodes (related to the complexity of the tree)
- There exist different stopping criteria (more or less complex)

\[ H(\text{LS}(n)) \leq H_{\text{min}} \]

\( H_{\text{min}} \): Minimum entropy

- Stopping criterion based on entropy:
  - Stop splitting the node \( n \) if the entropy is small
Classification trees
Real problem: Iris classification

96% of training observations are correctly classified (97% of the test set)

98% of training observations are correctly classified (97% of the test set)

Both models have the same test error rate. One should choose the simplest one
Classification trees
Wehenkel representation

The answer to the previous split can be Y (yes) or N (no)

Number of learning examples in this node

Number of test examples traversing this node

Proportion of test examples traversing the node that are correctly classified

Proportion of learning examples of each class in the node

split

atr2 < threshold2 ?

The answer to the previous split can be Y (yes) or N (no)

Proportion #2%

#1

#2% (#3)

atr1 < threshold1 ?

Y

N

atr2 < threshold2 ?

Proportion #2%

#1

#2% (#3)
Probabilistic approach: Linear Discriminant Analysis
Discriminant Analysis

Idea

- Assume that different output classes generate data based on different distributions

The number of observations of each class can be different.
Discriminant Analysis
Overview

• This method first uses the multivariate Gaussian distribution in order to model, for each output class $k$, the joint probability distribution of the inputs

$$f_k(X) \equiv \Pr(X = x | Y = k)$$

• Then it uses Bayes’ theorem to obtain the final probabilities

$$\Pr(Y = k | X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

• Finally it uses The Bayes classifier optimal rule to decide the estimated class from the previous posterior probabilities
Discriminant Analysis
Multivariate Gaussian distribution

• (Univariate) Gaussian distribution
  \[ X \approx N(\mu, \sigma^2) \]
  \[ f(x) = \frac{1}{\sqrt{2\pi \sigma^2}} e^{-(x-\mu)^2 / 2\sigma^2} \]

  Generalization to higher dimensions

• Multivariate Gaussian distribution
  \[ X \sim N(\mu, \Sigma) \]
  \[ f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp \left( -\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu) \right) \]
Discriminant Analysis
Multivariate Gaussian distribution

- **Parameters**

\[ X \sim N(\mu, \Sigma) \]

- \( \mu \): Vector of *means* \((p\text{-dimensional})\)
  - Center of the Gaussian in the \(p\)-dimensional input space
  - \( \mu = E(X) \)

- \( \Sigma \): Covariance matrix \((p \times p \text{ matrix})\)
  - Spread and Shape of the Gaussian
  - \( \Sigma = \text{Cov}(X) \)

Number of parameters: \( K + p(p + 1)/2 \)
Discriminant Analysis
Multivariate Gaussian distribution

- Bivariate Gaussian distribution (p=2)

\[ X \sim N(\mu, \Sigma) \]

\[ f(x,y) = \frac{1}{2\pi \sigma_x \sigma_y \sqrt{1-\rho^2}} \exp \left( -\frac{1}{2(1-\rho^2)} \left[ \frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x \sigma_y} \right] \right) \]

- Five parameters:

\[ \mu = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \sigma_x^2 & \rho \sigma_x \sigma_y \\ \rho \sigma_x \sigma_y & \sigma_y^2 \end{pmatrix} \]

> Covariance between vars \( x \) and \( y \)

Linear correlation coefficient

\[ \rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \]
Discriminant Analysis
Multivariate Gaussian distribution

- Example (p=2)

\[ \mu = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix} \]

Same variance and no correlation
Discriminant Analysis
Multivariate Gaussian distribution

• Example (p=2)

\[ \mu = \begin{pmatrix} 1.0 \\ -1.5 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 0.25 & 0.0 \\ 0.0 & 1.0 \end{pmatrix} \]

\[ \rho = \frac{0.0}{\sqrt{0.25 \cdot 1}} = 0.0 \]

var(x2) is larger than var(x1), there is no correlation

\[ \mu = \begin{pmatrix} 1.0 \\ -1.5 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 0.25 & 0.3 \\ 0.3 & 1.0 \end{pmatrix} \]

\[ \rho = \frac{0.3}{\sqrt{0.25 \cdot 1}} = 0.6 \]

var(x2) is larger than var(x1), there is positive correlation
Discriminant Analysis
Using Bayes’ Theorem for Classification

- **Bayes’ theorem** states that

\[
P(B_i|A) = \frac{P(A|B_i)P(B_i)}{\sum_{j=1}^{n}[P(A|B_j)P(B_j)]}
\]

\[
P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}
\]

- Thus, the **K posterior probabilities** can be computed by plugging in estimates of
  - **The priors** (easy if we have a random sample of the population)
    \[
    \pi_1, \ldots, \pi_K
    \]
  - **The likelihoods** (the difficult part, using Multivariate Gaussians)
    \[
    f_k(X) \equiv \Pr(X = x | Y = k)
    \]
Discriminant Analysis

Bayes Classifier: the optimal rule

• Assigns each observation to the most likely class, given its input values
  • This rule minimizes the TEST error rate

\[
\Pr(Y = j \mid X = x_0)
\]

Assign the class \(j\) for which the conditional probability is the largest

• Classification rule for two classes with equal priors (e.g. \(Y\) equals 1 or 2)

\[
\Pr(Y = 1 \mid X = x_0) > 0.5 \quad \text{YES} \\
\Pr(Y = 2 \mid X = x_0) \leq 0.5 \quad \text{NO}
\]

\(Y = 1\)

\(Y = 2\)
Discriminant Analysis
Bayes Classifier: the optimal rule

- Two-class problem with one input variable (K=2, p=1)
  - Same variance for each class

\[ \pi_1 = 0.5 \quad \pi_2 = 0.5 \]

\[ \pi_1 = 0.1 \]

\[ \pi_2 = 0.9 \]

If \( x=3 \) then class B is more probable.
Discriminant Analysis
Bayes Classifier: the optimal rule

- Three-class problem with one input variable (K=3, p=1)
  - Same variance for each class

Same priors

Different priors

\[
\begin{align*}
\pi_1 &= 0.1 \\
\pi_2 &= 0.1 \\
\pi_3 &= 0.8
\end{align*}
\]
Discriminant Analysis
Bayes Classifier: the optimal rule

- Three-class problem with one input variable (K=3, p=1)
  - Different variances and priors can imply very different posterior probabilities and partition of the input space

If x > 7.35 then Y = C
Discriminant Analysis

Linear Discriminant Analysis for $p=1$

- In order to estimate the densities, the linear version make some assumptions about its form
  - They are Gaussian
  - They have equal variance

$$f_k(x) = \frac{1}{\sqrt{2\pi\sigma_k}} \exp \left( -\frac{1}{2\sigma_k^2} (x - \mu_k)^2 \right).$$

$$\sigma_1^2 - \ldots - \sigma_K^2$$

- Then, the posterior probabilities are given by

$$Pr(Y = k|X = x) = \frac{\pi_k \frac{1}{\sqrt{2\pi\sigma}} \exp \left( -\frac{1}{2\sigma^2} (x - \mu_k)^2 \right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi\sigma}} \exp \left( -\frac{1}{2\sigma^2} (x - \mu_l)^2 \right)}$$
Discriminant Analysis

Linear Discriminant Analysis for \( p=1 \)

- The Bayes classifier rule involves assigning an observation to the largest posterior probability. It is equivalent to assigning the observation to the class for which the discriminant function is largest.

\[
\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)
\]

- Example

These are LINEAR functions of \( x \)
Discriminant Analysis

Linear Discriminant Analysis for $p=1$

• The LDA classifier first estimates all the required parameters of the classification model

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\pi}_k = \frac{n_k}{n}$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

• Then it plugs the estimates in the discriminant functions

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

• Finally, assign an observation $X=x$ to the class for which the discriminant function is largest
Discriminant Analysis

Linear Discriminant Analysis for \( p > 1 \)

- In order to estimate the densities, the linear version make some assumptions about its form:
  - They are multivariate Gaussian
  - They have equal covariance matrix

\[
f(x) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)
\]

- In this case the discriminant function is

\[
\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k
\]
Discriminant Analysis
Linear Discriminant Analysis for $p > 1$

• Three-class example (20 observations for each class)
Discriminant Analysis

Quadratic Discriminant Analysis

• Alternative approach to LDA

• QDA assumes that the observations of each class are drawn form a Multivariate Gaussian distribution, but each class has its own covariance matrix

\[ N(\mu_k, \Sigma) \quad \rightarrow \quad X \sim N(\mu_k, \Sigma_k) \]

LDA \hspace{3cm} K + p(p + 1)/2

QDA \hspace{3cm} K + Kp(p + 1)/2

• In this case the discriminant function is quadratic in x

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

\[
= -\frac{1}{2} x^T \Sigma_k^{-1} x + x^T \Sigma_k^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma_k^{-1} \mu_k - \frac{1}{2} \log |\Sigma_k| + \log \pi_k
\]
Discriminant Analysis

Quadratic Discriminant Analysis

• Two-class example (50 observations for each class)