A^3 - Aprendizagem Automática Avançada

Classification Systems
and
Evaluation Measures

𝔾. Marques

Classification

- Classification is the process of identifying or categorizing objects, beings, observations, ideas, etc, into a set of pre-defined classes.
- In machine learning, classification is a **supervised process**. This means that the decisions are based on a training set of pre-classified examples (*training set*).
- In order to use classification algorithms, each instance (observation, object, etc) has to be represented in the same fashion. It is common in machine learning to use a vector representation of each observation.
- The classifiers have to be tested on new data (the *test set*) in order to obtain a fair evaluation.

Classification

Types of Classification:

- **Multi-class** Classification:

  This is the most common scenario in classification. Each observation belongs to one of a pre-defined number of classes. The classes are **mutually exclusive**: one observation must belong to only one class.

- **Binary** Classification:

  Binary classification is a particular case of multi-class classification with only two classes.

  Binary classification is important because:

  - There are specific performance measures for the binary classification case.
  - Detection and retrieval problems can be viewed as binary classification cases.
  - Multi-class and multi-label classifications can be decomposed into several binary classification problems.

- **Multi-Label** Classification:

  In multi-label classification there are several classes but these **are not** mutually exclusive. In this case, the classes can be considered as labels or tags, and each observation can be identified with one or more tags (a.k.a. *auto-tagging*).

Classification

Example:

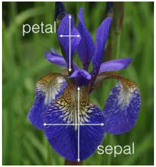
- Imagine that a botanist is interested in classifying three different types of iris flowers:

Iris Setosa

Iris Versicolor

Iris Virginica

- The botanist has taken measurements of petal and sepal widths and lengths. Based on these four measurements, the objective is to classify automatically the type of new iris flowers.



Classification

Example:

- Know the data:

  Import data scikit-learn

  ```
  >>> from sklearn import datasets
  ```

  Load dataset “Iris”

  ```
  >>> Iris=datasets.load_iris()
  ```

  Iris: variável do tipo dictionary, com vários campos:

  ```
  >>> Iris.keys() # ver os campos do dicionário
  ['target_names', 'data', 'target', 'DESCR', 'feature_names']
  ```

- Data – X is a np.array of (150,4):

  ```
  >>> X=Iris.data
  ```

- Data classes – trueClass é um np.array de (150,):

  ```
  >>> trueClass=Iris.target
  ```

Classification

Example:

- Dataset Description:

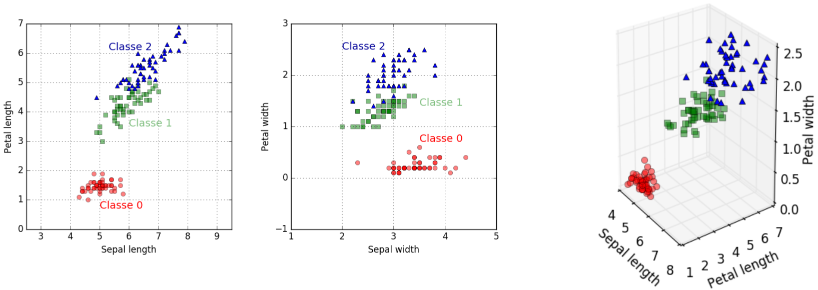
```
>>> print iris.DESC
```

```
Iris Plants Database Characteristics:
    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica
    :Summary Statistics:
    ============== ==== ==== ======= ===== ====================
                    Min  Max   Mean    SD   Class Correlation
    ============== ==== ==== ======= ===== ====================
    sepal length:   4.3  7.9   5.84   0.83    0.7826
    sepal width:    2.0  4.4   3.05   0.43   -0.4194
    petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
    petal width:    0.1  2.5   1.20   0.76    0.9565  (high!)
    ============== ==== ==== ======= ===== ====================
    :Missing Attribute Values: None
    :Class Distribution: 33.3% for each of 3 classes.
    :Creator: R.A. Fisher, July, 1988
```

Classification

Example:

- Data Visualization:

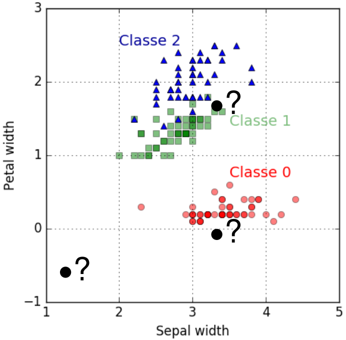


- Points from class 0 (*iris setosa*) are more compactly grouped than points from the other two classes.
- Classes 1 and 2 have some feature overlap.

Classification

Example:

- How to classify new data?



Classification

Theory and Notation:

(Binary and multi-class classification)

- Observations are represented by *d*-dimensional feature vectors.

  Data is also referred as:

  • points • vectors • observations • instance • patterns

  $$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

- Each feature vector belongs to one of a set of *c* classes:

  $\Omega = \{\varpi_1, \varpi_2, \cdots, \varpi_c\}$.

  Notation:

  - $\mathbf{x} \in \varpi_{\mathbf{k}} \quad \Longrightarrow$ the vector belongs to class *k*
  - $\mathbf{x} \in \hat{\varpi}_{\mathbf{k}} \quad \Longrightarrow$ o vector was classified in class *k*

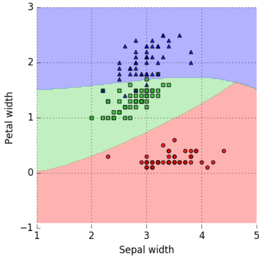
- Classification is equivalent to dividing the feature space into a set of *c* *decision regions*.
- Classification is also equivalent to defining a set of *c* **discriminant functions**.

Classification

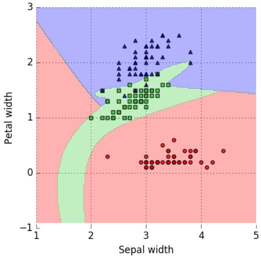
Decision Regions:

Classification can be accomplished by dividing the feature space into *c* decision regions (or surfaces) – as many as the number of classes.

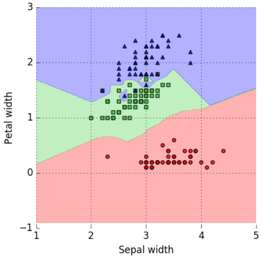
- The regions do not have to be contiguous and can span of distinct regions of the feature space.
- Each region is associated with a class.
- A new observation (vector) is classified by determining in which region it has fallen and assigning it the corresponding class.



Classifier 1



Classifier 2



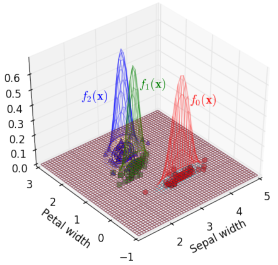
Classifier 3

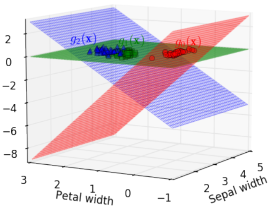
Classification

Discriminant Functions:

The process of enumeration the decision region can be very complex, particularly in high dimensional spaces. Typically it is preferable to use **discriminant functions**.

- Define *c* discrimant function – as many as classes.
  $\mathcal{F} = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_c(\mathbf{x})\}$
- Each function represents a class.
- Classifying a new vector, **x**, corresponds to determining which function has the highest output for **x**.
  $\mathbf{x} \in \hat{\varpi}_k$ se e só se $k = \underset{i=1,2,\ldots,c}{\operatorname{argmax}}(f_i(\mathbf{x}))$





Classification

Discriminant Functions:

With the set of c discriminant functions, one does not need to determine the decision regions of the feature space.

- A set of discriminant functions $\mathcal{F} = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_c(\mathbf{x})\}$ can be transformed into another equivalent set of discriminant functions (for example with a real, and monotonically increasing transformation).
    - $h(\cdot) \Longrightarrow$ real, monotonically increasing function.
    - $\mathcal{G} = \{g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_c(\mathbf{x})\}$ com $g_i(\mathbf{x}) = h(f_i(\mathbf{x}))$
    - The two sets $\mathcal{F}$ e $\mathcal{G}$ are equivalent (obtain the same results).
- Discriminant functions are gain functions – seek to determine which one produces the highest value (gain function are also called reward, utility, profit, etc).
- Classification can also be accomplished by using a set of **cost** or **loss** function. In this case, we seek to determine the function with the smallest value.

Classification

Performance Evaluation

In order to evaluate the performance of a classifier it is necessary to know two things:

1. The total probability of error of the classifier.
2. How the errors are distributed by class - the **confusion matrix**.

Classification

Confusion Matrix:

- Squared Matrix, **P** de $c \times c$, where c is the total number of classes.

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1c} \\ p_{21} & p_{22} & \cdots & p_{2c} \\ \vdots & & \ddots & \\ p_{c1} & p_{c2} & \cdots & p_{cc} \end{bmatrix}$$

- The matrix coefficients, p_{ij}, are probabilities.
  $p_{ij} = p(\mathbf{x} \in \hat{\varpi}_j | \mathbf{x} \in \varpi_i)$ is the probability of **x** belonging to class ϖ_i and being classified in class ϖ_j.
- Each line of the matrix pertains to a single class. In the first line are the probabilities for class ϖ_1, in the second line for the class ϖ_2, and so on.
- The sum of the probabilities in each line must add up to one:
  $\sum_{i=1}^{c} p_{ki} = 1$
- Ideally, **P** is the identity matrix (no errors).

Classification

Confusion Matrix:

- Squared Matrix, **P** de $c \times c$, where c is the total number of classes.

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1c} \\ p_{21} & p_{22} & \cdots & p_{2c} \\ \vdots & & \ddots & \\ p_{c1} & p_{c2} & \cdots & p_{cc} \end{bmatrix}$$

- The matrix coefficients, p_{ij}, are probabilities.
  $p_{ij} = p(\mathbf{x} \in \hat{\varpi}_j | \mathbf{x} \in \varpi_i)$ is the probability of **x** belonging to class ϖ_i and being classified in class ϖ_j.

- To analytically determine the value of p_{ij}, it is necessary:
  - Knowledge of the conditional distribution function of class ϖ_i.
    $p(\mathbf{x}|\varpi_i)$: **conditional probability** density function, given class ϖ_i.
  - Knowledge of the decision region, $\mathcal{S}_j$, of the class ϖ_j.
  - Calculate integral $p_{ij} = \int_{\mathcal{S}_j} p(\mathbf{x}|\varpi_i) d\mathbf{x}$

Classification

Confusion Matrix and Total Error Probability:

- The confusion matrix is a representation of the per-class error distributions. To determine the total error probability it is necessary to take account for the *a priori* class distributions, $p(\varpi_i)$ com $i = 1, \ldots, c$.
    - Error probability of class $\varpi_i = \sum_{j \neq i}^{c} p_{ij} = 1 - p_{ii}$
    - **Total error probability** $= \sum_{i=1}^{c} p(\varpi_i) \left(\sum_{j \neq i}^{c} p_{ij} \right) = \sum_{i=1}^{c} p(\varpi_i)(1 - p_{ii})$
- The total error probability is the sum of the error probabilities in the individual classes weighted by the classes a priori values.

Classification

Example: Consider a classifier defined by the following discriminant functions:

$f_1(x) = \exp(-x)$, $f_2(x) = \exp(-x^2 + 2)$, $f_3(x) = \exp(x/2 + 1/2)$.

Based on this table, determine the confusion matrix and the total error probability.

x	-1.5	0.5	-0.2	2.3	-2.1	2.5	1.5	-1.1	1.6	1.1	0.9	-0.1
ϖ	ϖ_2	ϖ_2	ϖ_2	ϖ_3	ϖ_1	ϖ_2	ϖ_2	ϖ_1	ϖ_3	ϖ_3	ϖ_2	ϖ_1

Classification

Example: Consider a classifier defined by the following discriminant functions:

$f_1(x) = \exp(-x)$, $f_2(x) = \exp(-x^2 + 2)$, $f_3(x) = \exp(x/2 + 1/2)$.

Based on this table, determine the confusion matrix and the total error probability.

x	-1.5	0.5	-0.2	2.3	-2.1	2.5	1.5	-1.1	1.6	1.1	0.9	-0.1
ϖ	ϖ_2	ϖ_2	ϖ_2	ϖ_3	ϖ_1	ϖ_2	ϖ_2	ϖ_1	ϖ_3	ϖ_3	ϖ_2	ϖ_1

R:

i. For simplification purposes, apply the logarithmic function to all the discriminant functions.

ii. Determine the decision regions.

iii. Classify the table data.

iv. Determine the confusion matrix and the total error probability.

Classification

Example: Consider a classifier defined by the following discriminant functions: $f_1(x) = \exp(-x)$, $f_2(x) = \exp(-x^2 + 2)$, $f_3(x) = \exp(x/2 + 1/2)$.

Based on this table, determine the confusion matrix and the total error probability.

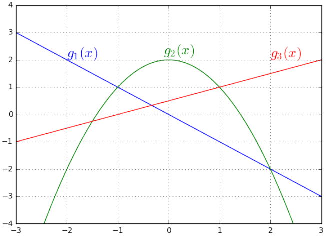
x	-1.5	0.5	-0.2	2.3	-2.1	2.5	1.5	-1.1	1.6	1.1	0.9	-0.1
ϖ	ϖ_2	ϖ_2	ϖ_2	ϖ_3	ϖ_1	ϖ_2	ϖ_2	ϖ_1	ϖ_3	ϖ_3	ϖ_2	ϖ_1

R:

i. $g_i(x) = \ln(f_i(x))$

$g_1(x) = -x$, $g_2(x) = -x^2 + 2$, $g_3(x) = x/2 + 1/2$.

ii. $\mathcal{S}_1 =]-\infty, -1]$, $\mathcal{S}_2 = [-1, +1]$, $\mathcal{S}_3 = [+1, +\infty[$



Classification

Example: Consider a classifier defined by the following discriminant functions:

$f_1(x) = \exp(-x)$, $f_2(x) = \exp(-x^2 + 2)$, $f_3(x) = \exp(x/2 + 1/2)$.

Based on this table, determine the confusion matrix and the total error probability.

x	-1.5	0.5	-0.2	2.3	-2.1	2.5	1.5	-1.1	1.6	1.1	0.9	-0.1
ϖ	ϖ_2	ϖ_2	ϖ_2	ϖ_3	ϖ_1	ϖ_2	ϖ_2	ϖ_1	ϖ_3	ϖ_3	ϖ_2	ϖ_1
$\hat{\varpi}$	ϖ_1	ϖ_2	ϖ_2	ϖ_3	ϖ_1	ϖ_3	ϖ_3	ϖ_1	ϖ_3	ϖ_3	ϖ_2	ϖ_2
	X					**X**	**X**					**X**

R:

iii. Classification errors marked with "**X**"

- $N = 12$ total number of points
- Class ϖ_1: $n_1 = 3 \Longrightarrow \quad p(\varpi_1) = \frac{3}{12} = \frac{1}{4}$
  $n_{11} = 2 \quad n_{12} = 1 \quad n_{13} = 0$
- Class ϖ_2: $n_2 = 6 \Longrightarrow \quad p(\varpi_2) = \frac{6}{12} = \frac{1}{2}$
  $n_{21} = 1 \quad n_{22} = 3 \quad n_{23} = 2$
- Class ϖ_3: $n_3 = 3 \Longrightarrow \quad p(\varpi_3) = \frac{3}{12} = \frac{1}{4}$
  $n_{31} = 0 \quad n_{32} = 0 \quad n_{33} = 3$

Classification

Example: Consider a classifier defined by the following discriminant functions:
$f_1(x) = \exp(-x)$, $f_2(x) = \exp(-x^2 + 2)$, $f_3(x) = \exp(x/2 + 1/2)$.

Based on this table, determine the confusion matrix and the total error probability.

x	-1.5	0.5	-0.2	2.3	-2.1	2.5	1.5	-1.1	1.6	1.1	0.9	-0.1
ϖ	ϖ_2	ϖ_2	ϖ_2	ϖ_3	ϖ_1	ϖ_2	ϖ_2	ϖ_1	ϖ_3	ϖ_3	ϖ_2	ϖ_1
$\hat{\varpi}$	ϖ_1	ϖ_2	ϖ_2	ϖ_3	ϖ_1	ϖ_3	ϖ_3	ϖ_1	ϖ_3	ϖ_3	ϖ_2	ϖ_2
	X					**X**	**X**					**X**

R:

iv. Confusion matrix and total error probability

- Class ϖ_1: $n_1 = 3$ e $n_{11} = 2 \quad n_{12} = 1 \quad n_{13} = 0$
  $p_{11} = \frac{n_{11}}{n_1} = \frac{2}{3} \quad p_{12} = \frac{1}{3} \quad p_{13} = 0$
- Class ϖ_2: $n_2 = 6$ e $n_{21} = 1 \quad n_{22} = 3 \quad n_{23} = 2$
  $p_{21} = \frac{1}{6} \quad p_{22} = \frac{1}{2} \quad p_{23} = \frac{1}{3}$
- Class ϖ_3: $n_3 = 3$ e $n_{31} = 0 \quad n_{32} = 0 \quad n_{33} = 3$
  $p_{21} == \quad p_{22} = 0 \quad p_{13} = 1$

$$\mathbf{P} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & 0 \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix}$$

Total error probability $= \sum_{i=1}^{3} (1 - p_{ii})\, p(\varpi_i) = \left(1 - \frac{2}{3}\right)\frac{3}{12} + \left(1 - \frac{1}{2}\right)\frac{6}{12} + (1 - 1)\,\frac{3}{12} = \frac{4}{12} = \frac{1}{3}$

Direct method:
There are 4 errors in 12 examples $\Longrightarrow$ total error probability $= \frac{4}{12}$

Classification

Practical Questions:

- Usually one does not know the class conditional probability density functions, $p(\mathbf{x}|\varpi_i)$.
- Usually one does not know the class decision regions.
- Even if one knows the two previous items, the probability $p_{ij} = \int_{\mathcal{S}_j} p(\mathbf{x}|\varpi_i)d\mathbf{x}$ is usually too complex to be determined analytically.

SOLUTION:

- Estimate the class conditioned error probabilities p_{ij} and the total error probability by analyzing the results on a set of observations for which we know the true class. This set is called the *test set* and consist of examples not present in the *training set* that was used to train the classifier. It is necessary to use new data in the evalution phase in order to have a reliable measure of the error, and assess its *generalization* capability.
- Based on the test set classification results, the confusion matrix coefficients can be estimated the following way: $p_{ij} = \dfrac{n_{ij}}{n_i}$

  n_{ij} number of points from class ϖ_i classified in class ϖ_j
  n_i number of points in class ϖ_i

Classification

Non-Normalized Confusion Matrix:

- Typically, the values of the confusion matrix coefficients, $p_{ij} = p(\mathbf{x} \in \varpi_j | \mathbf{x} \in \varpi_i)$, are based on the classification results of the test set.
- It is sometimes more intuitive to present the results in terms of absolute values rather then probabilities. The matrix coefficients are now integer numbers, n_{ij}, and the sum of each line is equal to the number of examples in that class.
- n_{ij} Number of examples in class ϖ_i classified in class $\hat{\varpi}_j$

$$\text{Normalized Confusion Matriz=} \begin{bmatrix} n_{11} & n_{12} & \cdots & n_{1c} \\ n_{21} & n_{22} & \cdots & n_{2c} \\ \vdots & & \ddots & \\ n_{c1} & n_{c2} & \cdots & n_{cc} \end{bmatrix}$$

Note: one can also obtain the total error with this matrix.

Example: Dataset `Iris`

- Data: iris flowers represented with 4-dimensional feature vectors ($\mathbf{x} = [x_1, x_2, x_3, x_4]^\top$).
- Classes: 3 iris species - setosa, versicolor, virginica (classes $\varpi_1, \varpi_2,$ e ϖ_3 respectively).
- 150 observations, 50 for each class.

Classification

Non-Normalized Confusion Matrix:

- Typically, the values of the confusion matrix coefficients, $p_{ij} = p(\mathbf{x} \in \varpi_j | \mathbf{x} \in \varpi_i)$, are based on the classification results of the test set.
- It is sometimes more intuitive to present the results in terms of absolute values rather then probabilities. The matrix coefficients are now integer numbers, n_{ij}, and the sum of each line is equal to the number of examples in that class.
- n_{ij} Number of examples in class ϖ_i classified in class $\hat{\varpi}_j$

$$\text{Normalized Confusion Matriz}= \begin{bmatrix} n_{11} & n_{12} & \cdots & n_{1c} \\ n_{21} & n_{22} & \cdots & n_{2c} \\ \vdots & & \ddots & \\ n_{c1} & n_{c2} & \cdots & n_{cc} \end{bmatrix}$$

Note: one can also obtain the total error with this matrix.

Example: Dataset `Iris`

Classification resultados wih the following discriminant functions:

$$\begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ f_3(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} -8 & 1 & 5 & -4 & -1 \\ 20 & 0 & -9 & 4 & -9 \\ -25 & 0 & 3 & 0 & 11 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \qquad \mathbf{P} = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 32 & 18 \\ 0 & 5 & 45 \end{bmatrix}$$

Total error probability $= (18 + 5)/150 \approx 15.3\%$

Binary Classification

The binary (two classes) case appears in many application areas

- Detection systems - is a patient ill?
- Alarm systems - is there an intrusion?
- Identification systems - is this the correct person?
- Tagging systems - is this music danceable?
- Information retrieval systems - has the search returned a relevant document?

- It is common to refer the two classes as the positive and the negative classes $(\varpi_{\mathrm{p}}, \varpi_{\mathrm{n}})$.
- In several real situations, the number of positive examples is significantly less than the negative ones. In this condition, the total accuracy (or the total error probability) is not a good performance measure. Systems that classify all the observations as negative ones obtain high accuracies.
- In binary classification there are several metrics that are more suitable than the accuracy to evaluate the performance.

Binary Classification

Performance Metrics

The performance metrics for the binary classification case are based in the coefficients of the non-normalized confusion matrix.

	$\hat{\varpi}_\mathrm{p}$	$\hat{\varpi}_\mathrm{n}$
ϖ_p	**T**rue **P**ositives	**F**alse **N**egatives
ϖ_n	**F**alse **P**ositives	**T**rue **N**egatives

- Positive Class ϖ_p:
  - Number of examples: TP+FN
  - $p\left(\varpi_\mathrm{p}\right) = \dfrac{\mathrm{TP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}}$
- Negative Class ϖ_n:
  - Number of examples: FP+TN
  - $p\left(\varpi_\mathrm{n}\right) = \dfrac{\mathrm{FP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}}$

- In binary classification, different classification metrics reflect different aspects of the performance of the classifiers. The choice of which metric to use is dependent on the problem at hand, and which type of errors are more important. For instance, in medical diagnostics it is common to use the *recall* with the *sensitivity*, while in machine learning and information retrieval the recall and *precision* are usually preferred.

Binary Classification

Performance Metrics

The performance metrics for the binary classification case are based in the coefficients of the non-normalized confusion matrix.

	$\hat{\varpi}_{\mathrm{p}}$	$\hat{\varpi}_{\mathrm{n}}$
ϖ_{p}	**T**rue **P**ositives	**F**alse **N**egatives
ϖ_{n}	**F**alse **P**ositives	**T**rue **N**egatives

- Positive Class ϖ_{p}:
  - Number of examples: TP+FN
  - $p\left(\varpi_{\mathrm{p}}\right) = \dfrac{\mathrm{TP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}}$
- Negative Class ϖ_{n}:
  - Number of examples: FP+TN
  - $p\left(\varpi_{\mathrm{n}}\right) = \dfrac{\mathrm{FP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}}$

- There are eight basic metrics that can be directly obtained from the non-normalized confusion matrix. These are calculated by dividing each of the eight coefficients by the sum of the lines or the columns of the matrix.
- The normalization by the sum of the line values, the metrics are calculated in terms of class percentages. These metrics are not affected by class imbalance.
- The normalization column wise refers to the number of points classified in each class. These values are affected by the number of examples in each class.

Binary Classification

Performance Metrics

	$\hat{\varpi}_{\mathrm{p}}$	$\hat{\varpi}_{\mathrm{n}}$
ϖ_{p}	**T**rue **P**ositives	**F**alse **N**egatives
ϖ_{n}	**F**alse **P**ositives	**T**rue **N**egatives

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} \frac{\mathrm{TP}}{\mathrm{TP+FN}} & \frac{\mathrm{FN}}{\mathrm{TP+FN}} \\ \frac{\mathrm{FP}}{\mathrm{FP+TN}} & \frac{\mathrm{TN}}{\mathrm{FP+TN}} \end{bmatrix}$$

(normalized confusion matrix)

$$\text{Total Error Probability} = \frac{\mathrm{FP+FN}}{\mathrm{TP+FP+TN+FN}}$$

- $\text{TP-rate} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} = p_{11}$

  correctly classified positives

  Synonyms: • **recall** • sensitivity

- $\text{FN-rate} \frac{\mathrm{FN}}{\mathrm{TP+FN}} = p_{12}$

  incorrectly classified positives

- $\text{TN rate} = \frac{\mathrm{TN}}{\mathrm{FP+TN}} = p_{22}$

  correctly classified negatives

  Synonyms: • specificity

- $\text{FP rate} = \frac{\mathrm{FP}}{\mathrm{FP+TN}} = p_{21}$

  incorrectly classified negatives

  Synonyms: • false alarm • fall-out

Binary Classification

Performance Metrics

	$\hat{\varpi}_\mathrm{p}$	$\hat{\varpi}_\mathrm{n}$
ϖ_p	**T**rue **P**ositives	**F**alse **N**egatives
ϖ_n	**F**alse **P**ositives	**T**rue **N**egatives

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} \frac{\mathrm{TP}}{\mathrm{TP+FN}} & \frac{\mathrm{FN}}{\mathrm{TP+FN}} \\ \frac{\mathrm{FP}}{\mathrm{FP+TN}} & \frac{\mathrm{TN}}{\mathrm{FP+TN}} \end{bmatrix}$$

(normalized confusion matrix)

$$\text{Total Error Probability} = \frac{\mathrm{FP+FN}}{\mathrm{TP+FP+TN+FN}}$$

- PPV Positive Predicted Value $= \dfrac{\mathrm{TP}}{\mathrm{TP+FP}}$

  classified as positive correctly

  Sinónimos: • **precision**

- FDR False Discovery Rate $= \dfrac{\mathrm{FP}}{\mathrm{TP+FP}}$

  classified as positive incorrectly

  Note that $\mathrm{FDR} = 1 - \mathrm{PPV}$

- NPV Negative Predicted Value $= \dfrac{\mathrm{TN}}{\mathrm{TN+FN}}$

  classified as negative correctly

- FOR False Omission Rate $= \dfrac{\mathrm{FN}}{\mathrm{TN+FN}}$

  classified as negative incorrectly

Binary Classification

Performance Metrics

	$\hat{\varpi}_\mathrm{p}$	$\hat{\varpi}_\mathrm{n}$
ϖ_p	**T**rue **P**ositives	**F**alse **N**egatives
ϖ_n	**F**alse **P**ositives	**T**rue **N**egatives

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} \frac{\mathrm{TP}}{\mathrm{TP+FN}} & \frac{\mathrm{FN}}{\mathrm{TP+FN}} \\ \frac{\mathrm{FP}}{\mathrm{FP+TN}} & \frac{\mathrm{TN}}{\mathrm{FP+TN}} \end{bmatrix}$$

(normalized confusion matrix)

$$\text{Total Error Probability} = \frac{\mathrm{FP+FN}}{\mathrm{TP+FP+TN+FN}}$$

- Precision + recall are the commonly used metrics in machine learning and information retrieval. Be aware that trivial classifiers can obtain good results in recall or in precision. Only truly valid classifiers obtain good results in both metrics.
- Precision and recall related metrics
  - F-Score$= \dfrac{1}{1/\text{recall} + 1/\text{precision}} = 2 \times \dfrac{\text{precision} \times \text{recall}}{\text{precision+recall}}$
    (harmonic mean for precision and recall)
  - G-Score$= \sqrt{\text{precision} \times \text{recall}}$
    (geometric mean for precision and recall)

Binary Classification

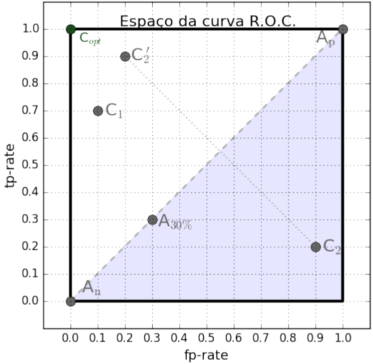
ROC curves

- ROC curves (Receiving Operating Characteristics) plots to illustrate and compare performance of one or more classifiers.
- In a ROC curve the true positive rate (recall) is plotted against the false positive rate (fp-rate). The values are comprised in the interval $[0, 1]$.
- ROC curves are an important tool for classifiers diagnostics and evaluations.
- One classifiers corresponds to a single point in the curve.
- In most all classification models the decision threshold can be adjusted to produce more conservative or liberal outcomes. Different threshold values result in a curve in the ROC space that can be used to calibrate the model.

Binary Classification

ROC curves

ROC curves for 7 classifiers: $\mathrm{A_n}$, $\mathrm{A_p}$, $A_{30\%}$, C_1, C_2, C_2', e C_{opt}.

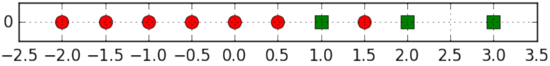


- C_{opt}: optimal classifier (no errors).
- Random classifiers: $\mathrm{A_n}$, $\mathrm{A_p}$ e $A_{30\%}$
  All classifiers that are located in the **dotted diagonal line** (from $(0, 0)$ to $(1, 1)$) are random classifiers.
- Inferior triangle: classifiers that are **worst** than random. These can be repositioned in the upper triangular part of the curve by **inverting** the decision (replace positives with negatives and vice versa).
  Example: C_2 e C_2'

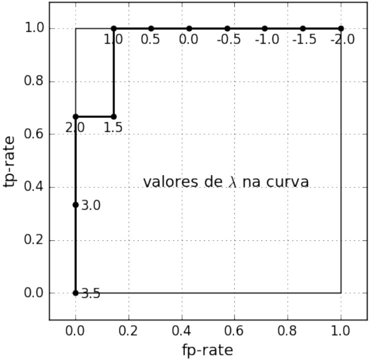
Binary Classification

ROC curves - Example 1:

Consider a $N=10$, 1D point set divide into two classes ($\square \in \varpi_\mathrm{p}$, $\circ \in \varpi_\mathrm{n}$).



Also consider the following classification: $x \in \hat{\varpi}_\mathrm{p}$ se $x \geq \lambda$, $x \in \hat{\varpi}_\mathrm{n}$ se $x < \lambda$. Shifting the threshold λ from $-\infty$ to $+\infty$ produces a ROC curve (in this case it was enough to vary λ from 3.5 to -2 by 0.5 units).



λ	tp-rate	fp-rate
+3.5	0	0
+3.0	1/3	0
+2.5	1/3	0
+2.0	2/3	0
+1.5	2/3	1/7
+1.0	1	1/7
+0.5	1	2/7
0.0	1	3/7
−0.5	1	4/7
−1.0	1	5/7
−1.5	1	6/7
−2.0	1	1

Binary Classification

ROC curves - Example 2:

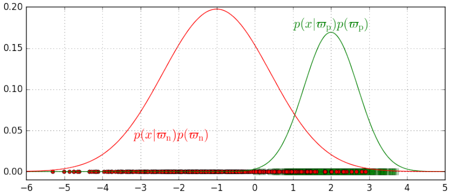
Consider a 1D point set divided into two classes with the following distributions:

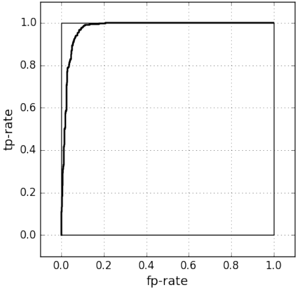
Positives: $p\left(x|\varpi_{\mathrm{p}}\right) = \dfrac{1}{\sqrt{\pi}} \exp\left\{-(x-2)^2\right\}$ e $p(\varpi_{\mathrm{p}}) = 0.3$

Negatives: $p\left(x|\varpi_{\mathrm{n}}\right) = \dfrac{1}{\sqrt{4\pi}} \exp\left\{-\dfrac{1}{4}(x+1)^2\right\}$ e $p(\varpi_{\mathrm{n}}) = 0.7$

Consider the following classifier: $x \in \hat{\varpi}_{\mathrm{p}}$ se $x \geq \lambda$, $x \in \hat{\varpi}_{\mathrm{n}}$ se $x < \lambda$.

In the left figure are represented $N = 1000$ points and the corresponding density functions. In the right hand side figure is the classifiers ROC curve obtained by varying the threshold λ.





Binary Classification

Other Performances Curves and Measures:

- AUC -Area Under the ROC Curve:
  AUC is a measure that combines the tp-rate and fp-rate for the different thresholds. It is the classifiers capacity to correctly discriminat between positive and negative observations.

- DET - Detection Error Tradeoff:
  DET curves show the false negative rate (fn-rate) versus false positive rate (fp-rate).

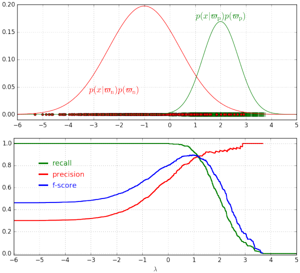
- Precision vs Recall curves:
  Precision-recall curves are widely used in machine learning. Note that while DET or ROC curves are not affected by class imbalance, the precision-recall curves are.

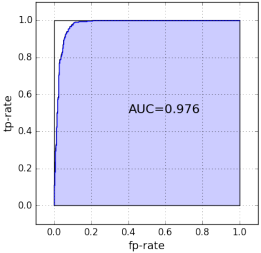
Binary Classification

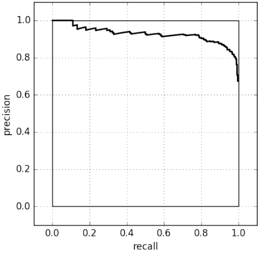
Example 2:

$$p\left(x|\varpi_{\mathrm{p}}\right) = \frac{1}{\sqrt{\pi}} \exp\left\{-(x-2)^2\right\} \text{ e } p(\varpi_{\mathrm{p}}) = 0.3$$

$$p\left(x|\varpi_{\mathrm{n}}\right) = \frac{1}{\sqrt{4\pi}} \exp\left\{-\frac{1}{4}(x+1)^2\right\} \text{ e } p(\varpi_{\mathrm{n}}) = 0.7$$





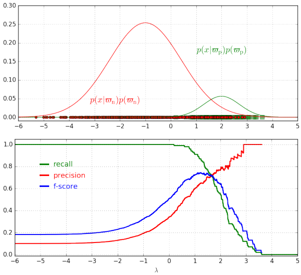


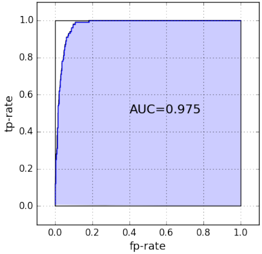
Binary Classification

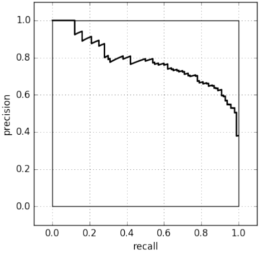
Example 2:

$$p\left(x|\varpi_{\mathrm{p}}\right) = \frac{1}{\sqrt{\pi}} \exp\left\{-(x-2)^2\right\} \text{ e } p(\varpi_{\mathrm{p}}) = 0.1$$

$$p\left(x|\varpi_{\mathrm{n}}\right) = \frac{1}{\sqrt{4\pi}} \exp\left\{-\frac{1}{4}(x+1)^2\right\} \text{ e } p(\varpi_{\mathrm{n}}) = 0.9$$







- The ROC curve remains approximately the same in both cases.
- The precision-recall curve varies significantly.