Machine Learning

Classification Metrics. Binary and Multi-Class cases.

Aleksandr Petiushko

ML Research

SOFIA UNIVERSITY

AP

Content

Content

1. Binary Classification Definitions
2. Confusion Matrix

Content

Content

Main math concepts: a reminder

- Kronecker delta function notation $f = [conditional\_expression]$:
  - $f = 0$ if the *condition* is not satisfied,
  - $f = 1$ if the *condition* is satisfied;
- Example: if $x = 10$, then:
  - $[x > 10] = 0$,
  - $[x = 10] = 1$.

Classification of binary classifier responses

- Training set $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$
- Classification problem into 2 classes: $X \to Y, Y = \{+1, -1\}$
- Classification algorithm $a(x) : X \to Y$
- The class labeled "+1" is called "**positive**"
- The class labeled "-1" is called "**negative**"

Classification of binary classifier responses

- Training set $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$
- Classification problem into 2 classes: $X \to Y, Y = \{+1, -1\}$
- Classification algorithm $a(x) : X \to Y$
- The class labeled "+1" is called "**positive**"
- The class labeled "-1" is called "**negative**"

Table: Classification of responses

	Algorithm output	Correct answer
TP (True Positive)	$a(x_i) = +1$	$y_i = +1$
TN (True Negative)	$a(x_i) = -1$	$y_i = -1$
FP (False Positive)	$a(x_i) = +1$	$y_i = -1$
FN (False Negative)	$a(x_i) = -1$	$y_i = +1$

Confusion Matrix

Let's depicted these relationships via a **confusion matrix** (a matrix of errors)

		Correct answer	
		$y = +1$	$y = -1$
Algorithm Output	$a(x) = +1$	True Positive	False Positive (Type 1 Error)
	$a(x) = -1$	False Negative (Type 2 Error)	True Negative

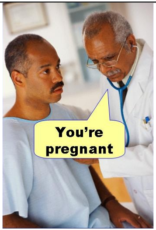
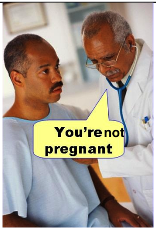
Confusion Matrix

Let's depicted these relationships via a **confusion matrix** (a matrix of errors)

		Correct answer	
		$y = +1$	$y = -1$
Algorithm Output	$a(x) = +1$	True Positive	False Positive (Type 1 Error)
	$a(x) = -1$	False Negative (Type 2 Error)	True Negative

Note. Words "*positive*"/"*negative*" signalize about the output of a classifier $a(x)$, while the words "*true*"/"*false*" compare the output of a classifier $a(x)$ with the ground truth label y.

Confusion Matrix

	$y = +1$	$y = -1$
$a(x) = +1$		
$a(x) = -1$		

The simplest quality metric

- The simplest quality metric is the proportion of correct answers on a test (control sample)
- Common name: **Accuracy**

Accuracy formula

$$Accuracy = \frac{1}{m}\sum_{i=1}^{m}[a(x_i) = y_i] = \frac{TP+TN}{TP+FP+TN+FN}$$

The simplest quality metric

- The simplest quality metric is the proportion of correct answers on a test (control sample)
- Common name: **Accuracy**

Accuracy formula

$$Accuracy = \frac{1}{m}\sum_{i=1}^{m}[a(x_i) = y_i] = \frac{TP+TN}{TP+FP+TN+FN}$$

Disadvantages

- Ignores class imbalance
- The cost of an error on objects of different classes is not taken into account

Metrics based on the positive response of the algorithm

Consider the metrics that are based on the calculation of the proportion of positive responses of the algorithm.

False Positive Rate, or **FPR**

It is a proportion of *incorrect* positive classifications among objects with ground truth label $y = -1$.

$$FPR(a, X^m) = \frac{\sum_{i=1}^{m} [y_i = -1][a(x_i) = +1]}{\sum_{i=1}^{m} [y_i = -1]} = \frac{FP}{FP + TN}$$

Metrics based on the positive response of the algorithm

Consider the metrics that are based on the calculation of the proportion of positive responses of the algorithm.

False Positive Rate, or **FPR**

It is a proportion of *incorrect* positive classifications among objects with ground truth label $y = -1$.

$$FPR(a, X^m) = \frac{\sum_{i=1}^m [y_i = -1][a(x_i) = +1]}{\sum_{i=1}^m [y_i = -1]} = \frac{FP}{FP + TN}$$

True Positive Rate, or **TPR**

It is a proportion of *correct* positive classifications among among objects with ground truth label $y = +1$.

$$TPR(a, X^m) = \frac{\sum_{i=1}^m [y_i = +1][a(x_i) = +1]}{\sum_{i=1}^m [y_i = +1]} = \frac{TP}{TP + FN}$$

Metrics based on the positive response of the algorithm

Consider the metrics that are based on the calculation of the proportion of positive responses of the algorithm.

False Positive Rate, or **FPR**

It is a proportion of *incorrect* positive classifications among objects with ground truth label $y = -1$.

$$FPR(a, X^m) = \frac{\sum_{i=1}^m [y_i = -1][a(x_i) = +1]}{\sum_{i=1}^m [y_i = -1]} = \frac{FP}{FP + TN}$$

True Positive Rate, or **TPR**

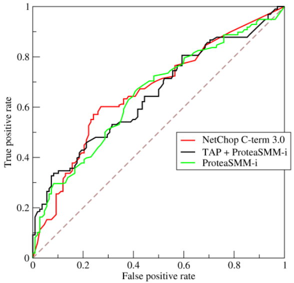
It is a proportion of *correct* positive classifications among among objects with ground truth label $y = +1$.

$$TPR(a, X^m) = \frac{\sum_{i=1}^m [y_i = +1][a(x_i) = +1]}{\sum_{i=1}^m [y_i = +1]} = \frac{TP}{TP + FN}$$

Note. Notice the different denominators!

Error Curve

Best known as **Receiver Operating Characteristic** (**ROC-curve**), in which we look at the trade-off between false alarm rate and correct response rate.

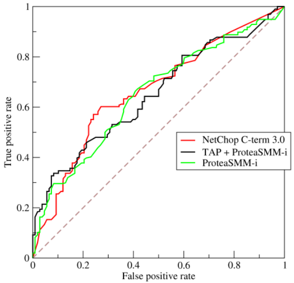


FPR is plotted along the X-axis, TPR is plotted along the Y-axis[1].

[1] https://wikipedia.org

Error Curve

Best known as **Receiver Operating Characteristic** (**ROC-curve**), in which we look at the trade-off between false alarm rate and correct response rate.



FPR is plotted along the X-axis, TPR is plotted along the Y-axis[1].
Note. On this curve, **miss rate** (FN) is not taken into account in any way.

[1] `https://wikipedia.org`

Area under the ROC curve and types of ROC curves

AUROC

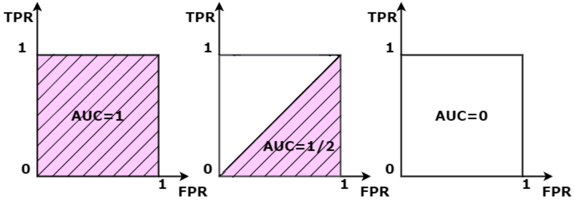
The greater the value of the correct TPR prediction for each FPR error value, the better the classifier performs.
Thus, the area under the curve (**Area Under Curve, AUC / AUROC**) must be maximized.

Area under the ROC curve and types of ROC curves

AUROC

The greater the value of the correct TPR prediction for each FPR error value, the better the classifier performs.
Thus, the area under the curve (**Area Under Curve, AUC / AUROC**) must be maximized.

ROC-curves for the best (AUC=1), random (AUC=0.5) and worst (AUC=0) algorithm:



The Task: build ROC, find AUROC

Suppose that the binary classification algorithm $a(x_i)$ on the sample X^m decides to assign a class based on some scalar value $g_\theta(x_i) \in \mathbb{R}$, where θ is the set of model parameters and $g_\theta(x_i)$ is the discriminant function:

- Let's treat Positive response by a (varying) threshold t: $g_\theta(x_i) \geq t$

The Task: build ROC, find AUROC

Suppose that the binary classification algorithm $a(x_i)$ on the sample X^m decides to assign a class based on some scalar value $g_\theta(x_i) \in \mathbb{R}$, where θ is the set of model parameters and $g_\theta(x_i)$ is the discriminant function:

- Let's treat Positive response by a (varying) threshold t: $g_\theta(x_i) \geq t$

Task

- We want to build an ROC curve, i.e. find points $\{(FPR_i, TPR_i)\}_{i=1}^{m}$
- Calculate area under curve - AUROC

The Task: build ROC, find AUROC

Suppose that the binary classification algorithm $a(x_i)$ on the sample X^m decides to assign a class based on some scalar value $g_\theta(x_i) \in \mathbb{R}$, where θ is the set of model parameters and $g_\theta(x_i)$ is the discriminant function:

- Let's treat Positive response by a (varying) threshold t: $g_\theta(x_i) \geq t$

Task

- We want to build an ROC curve, i.e. find points $\{(FPR_i, TPR_i)\}_{i=1}^m$
- Calculate area under curve - AUROC

Let's count the number of correct answers of different types:

- $m_+ = \sum_{i=1}^m [y(x_i) = +1]$ (TPR denominator)
- $m_- = \sum_{i=1}^m [y(x_i) = -1]$ (FPR denominator); $m = m_+ + m_-$

Let us order the training set X^m in descending order of the values $g_\theta(x_i)$.
Then the formula for $AUROC = \frac{1}{m_-} \sum_{i=1}^m [y_i = -1] TPR_i$ (see below).

Task solution

Algorithm

We put the first point at the origin: $(FPR_0, TPR_0) = (0, 0), AUROC = 0$.

Task solution

Algorithm

We put the first point at the origin: $(FPR_0, TPR_0) = (0,0)$, $AUROC = 0$.

Loop over ordered selection $i = 1 \dots m$

Threshold — the next value of the discriminant function $t = g_\theta(x_i)$

If $y_i = -1$:

- $(FPR_i, TPR_i) = (FPR_{i-1} + \frac{1}{m_-}, TPR_{i-1})$ (move along the X-axis)
- $AUROC = AUROC + \frac{1}{m_-} TPR_i$

Task solution

Algorithm

We put the first point at the origin: $(FPR_0, TPR_0) = (0, 0)$, $AUROC = 0$.

Loop over ordered selection $i = 1 \dots m$

Threshold — the next value of the discriminant function $t = g_\theta(x_i)$

If $y_i = -1$:

- $(FPR_i, TPR_i) = (FPR_{i-1} + \frac{1}{m_-}, TPR_{i-1})$ (move along the X-axis)
- $AUROC = AUROC + \frac{1}{m_-} TPR_i$

If $y_i = +1$:

- $(FPR_i, TPR_i) = (FPR_{i-1}, TPR_{i-1} + \frac{1}{m_+})$ (move along the Y-axis)

Metrics based on the negative (or missing) response of the algorithm

Consider the metrics that are based on the calculation of the proportion of negative responses of the algorithm.

False Negative Rate, or **FNR**

It is a proportion of *incorrect* negative classifications among objects with ground truth label $y = +1$.

$$FNR(a, X^m) = \frac{\sum_{i=1}^{m} [y_i = +1][a(x_i) = -1]}{\sum_{i=1}^{m} [y_i = +1]} = \frac{FN}{FN + TP} = 1 - TPR$$

Metrics based on the negative (or missing) response of the algorithm

Consider the metrics that are based on the calculation of the proportion of negative responses of the algorithm.

False Negative Rate, or **FNR**

It is a proportion of *incorrect* negative classifications among objects with ground truth label $y = +1$.

$FNR(a, X^m) = \frac{\sum_{i=1}^{m}[y_i=+1][a(x_i)=-1]}{\sum_{i=1}^{m}[y_i=+1]} = \frac{FN}{FN+TP} = 1 - TPR$

True Negative Rate, or **TNR**

It is a proportion of *correct* negative classifications among among objects with ground truth label $y = -1$.

$TNR(a, X^m) = \frac{\sum_{i=1}^{m}[y_i=-1][a(x_i)=-1]}{\sum_{i=1}^{m}[y_i=-1]} = \frac{TN}{TN+FP} = 1 - FPR$

Metrics based on the negative (or missing) response of the algorithm

Consider the metrics that are based on the calculation of the proportion of negative responses of the algorithm.

False Negative Rate, or **FNR**

It is a proportion of *incorrect* negative classifications among objects with ground truth label $y = +1$.

$FNR(a, X^m) = \frac{\sum_{i=1}^m [y_i = +1][a(x_i) = -1]}{\sum_{i=1}^m [y_i = +1]} = \frac{FN}{FN+TP} = 1 - TPR$

True Negative Rate, or **TNR**

It is a proportion of *correct* negative classifications among among objects with ground truth label $y = -1$.

$TNR(a, X^m) = \frac{\sum_{i=1}^m [y_i = -1][a(x_i) = -1]}{\sum_{i=1}^m [y_i = -1]} = \frac{TN}{TN+FP} = 1 - FPR$

Note. Notice the different denominators!

Other Important Metrics 1

In information retrieval problems

- **Precision**: $Precision = \frac{TP}{TP+FP}$ (percentage of relevant objects among those found)
- **Recall**: $Recall = \frac{TP}{TP+FN} = TPR$ (percentage of found objects among relevant ones)

Other Important Metrics 1

In information retrieval problems

- **Precision**: $Precision = \frac{TP}{TP+FP}$ (percentage of relevant objects among those found)
- **Recall**: $Recall = \frac{TP}{TP+FN} = TPR$ (percentage of found objects among relevant ones)

How to apply

- **Precision**: allows you to ensure that there are few false alarms; but it does not say anything about misses (the cost of a false alarm is high, and the price of a miss is low).
- **Recall**: allows you to ensure that there are few misses; but it does not say anything about false alarms (the price of a miss is high, and the price of a false alarm is low).

Other Important Metrics 1

In information retrieval problems

- **Precision**: $Precision = \frac{TP}{TP+FP}$ (percentage of relevant objects among those found)
- **Recall**: $Recall = \frac{TP}{TP+FN} = TPR$ (percentage of found objects among relevant ones)

How to apply

- **Precision**: allows you to ensure that there are few false alarms; but it does not say anything about misses (the cost of a false alarm is high, and the price of a miss is low).
- **Recall**: allows you to ensure that there are few misses; but it does not say anything about false alarms (the price of a miss is high, and the price of a false alarm is low).

Remark. Often the task is to optimize one metric while fixing another.

Other Important Metrics 2

In problems of medical diagnostics

- **Sensitivity**: $Sensitivity = \frac{TP}{TP+FN} = Recall$ (percentage of correct positive diagnoses)
- **Specificity**: $Specificity = \frac{TN}{TN+FP} = TNR$ (percentage of correct negative diagnoses)

Other Important Metrics 2

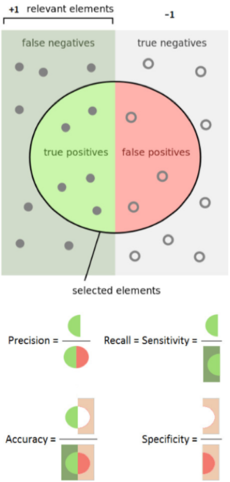
In problems of medical diagnostics

- **Sensitivity**: $Sensitivity = \frac{TP}{TP+FN} = Recall$ (percentage of correct positive diagnoses)
- **Specificity**: $Specificity = \frac{TN}{TN+FP} = TNR$ (percentage of correct negative diagnoses)

How to apply

- **Sensitivity**: Maximize the number of true positive diagnoses, but ignore false diagnoses (treatment cost is low and skip cost is high).
- **Specificity**: Maximize the number of correct negative diagnoses, but don't take into account missed diagnoses (treatment cost is high and skip cost is low).

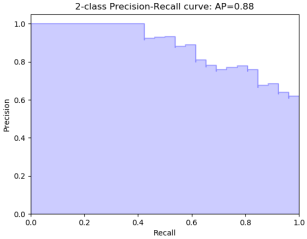
Metrics illustration[2]



[2] https://wikipedia.org

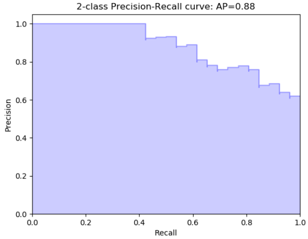
Aggregated Metrics over Precision-Recall

You can build a Precision-Recall (**PR-curve**) similar to the ROC-curve:



Aggregated Metrics over Precision-Recall

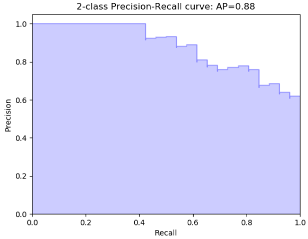
You can build a Precision-Recall (**PR-curve**) similar to the ROC-curve:



Remark. Note that in this case the curve is **not** necessarily **monotonic**!

Aggregated Metrics over Precision-Recall

You can build a Precision-Recall (**PR-curve**) similar to the ROC-curve:



Remark. Note that in this case the curve is **not** necessarily **monotonic**!

AUPRC

- Similar to AUROC, you can calculate the area under the PR curve - **AUPRC**
- Another name is **Average Precision** (with some assumptions on the integration method): the more, the better

Multi-class classification

For each class $c \in Y$, denote by TP_c, FP_c, and FN_c true positives, false positives, and false negatives. Then:

Precision and recall with micro-averaging

- $Precision = \frac{\sum_c TP_c}{\sum_c (TP_c + FP_c)}$
- $Recall = \frac{\sum_c TP_c}{\sum_c (TP_c + FN_c)}$
- Insensitive to errors on small classes

Multi-class classification

For each class $c \in Y$, denote by TP_c, FP_c, and FN_c true positives, false positives, and false negatives. Then:

Precision and recall with **micro-averaging**

- $Precision = \frac{\sum_c TP_c}{\sum_c (TP_c + FP_c)}$
- $Recall = \frac{\sum_c TP_c}{\sum_c (TP_c + FN_c)}$
- Insensitive to errors on small classes

Precision and recall with **macro-averaging**

- $Precision = \frac{1}{|Y|} \sum_c \frac{TP_c}{TP_c + FP_c}$
- $Recall = \frac{1}{|Y|} \sum_c \frac{TP_c}{TP_c + FN_c}$
- Sensitive to errors on small classes

Mandatory external links to read

1. Classification metrics:
   - The page about hands-on example of the Confusion Matrix calculation (read until "*Confusion Matrix Using Scikit-learn in Python*").
   - The page about hands-on example of the Binary Classifier metrics calculation (starting from "Confusion Matrix" until "*Exploring F1-score*").
   - Two examples on how to calculate Classification metrics in the multi-class settings (micro- vs macro- averaging): example1, example2.

Summary of classification quality metrics

- Precision and Recall are suitable for information retrieval tasks when the proportion of objects of the relevant class is small

Summary of classification quality metrics

- Precision and Recall are suitable for information retrieval tasks when the proportion of objects of the relevant class is small
- Sensitivity and specificity are usually used in medicine

Summary of classification quality metrics

- Precision and Recall are suitable for information retrieval tasks when the proportion of objects of the relevant class is small
- Sensitivity and specificity are usually used in medicine
- Macro-averaging is the solution for the imbalanced multi-class problem

Summary of classification quality metrics

- Precision and Recall are suitable for information retrieval tasks when the proportion of objects of the relevant class is small
- Sensitivity and specificity are usually used in medicine
- Macro-averaging is the solution for the imbalanced multi-class problem
- AUROC is suitable for quality assessment with a non-fixed error (miss rate) cost ratio

Summary of classification quality metrics

- Precision and Recall are suitable for information retrieval tasks when the proportion of objects of the relevant class is small
- Sensitivity and specificity are usually used in medicine
- Macro-averaging is the solution for the imbalanced multi-class problem
- AUROC is suitable for quality assessment with a non-fixed error (miss rate) cost ratio
- Another aggregated quality score - F-measure:
  $F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$
  - This is the *harmonic mean* that goes to zero when at least one of the values goes to zero

Summary of classification quality metrics

- Precision and Recall are suitable for information retrieval tasks when the proportion of objects of the relevant class is small
- Sensitivity and specificity are usually used in medicine
- Macro-averaging is the solution for the imbalanced multi-class problem
- AUROC is suitable for quality assessment with a non-fixed error (miss rate) cost ratio
- Another aggregated quality score - F-measure:
  $F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$
  - This is the *harmonic mean* that goes to zero when at least one of the values goes to zero
- TP/FP/TN/FN are just **counts**, while TPR/FPR/TNR/FNR are **ratios** (from 0 to 1)

Thank you!