# Evaluation metrics for classification

Tianpei Xie

Sep. 1st., 2022

## Contents

Table 3.2. *Alternative representation of the confusion matrix*

| | Pred_Negative | Pred_Positive | |
|---|---|---|---|
| Act_Negative | True negative (TN) | False positive (FP) | $N = \text{TN} + \text{FP}$ |
| Act_Positive | False negative (FN) | True positive (TP) | $P = \text{FN} + \text{TP}$ |

**Figure 1: The confusion table for binary classification**

# 1 Confusion table and related metrics

- For a test set $T$ of examples and a classifier $f$, the **confusion matrix** [Japkowicz and Shah, 2011] $\boldsymbol{C}(f)$ can be defined as

$$\boldsymbol{C}(f) = \left\{ c_{i.j}(f) = \sum_{\boldsymbol{x} \in T} \mathbb{1}\left\{(y = i) \wedge (f(\boldsymbol{x}) = j)\right\} \right\} \in \mathbb{R}^{l \times l}, \tag{1}$$

  where $\boldsymbol{x}$ is a test example and $y$ is its corresponding label such that $y \in \{1, 2, \ldots, l\}$. Each element $c_{i,j}(f)$ of the confusion matrix denotes the number of examples that *actually* have a class $i$ label and that the classifier $f$ *assigns* to class $j$. Columns corespond to predicted values and rows correspond to true values.

- row sum $\sum_j c_{i,j}(f) = c_{i,\cdot}(f)$ denotes the total number of examples of class $i$ in the test set. column sum $\sum_i c_{i,j}(f) = c_{\cdot,j}(f)$ denotes the total of examples assigned to class $j$ by classifier $f$. diagonal sum is the total number of *corrected classified* examples; all the nondiagonal entries denote *misclassifications*.

- The ***empirical error rate*** for classifier $f$ is

$$Error(f) := \frac{\sum_{i \neq j} c_{i.j}(f)}{\sum_i \sum_j c_{i.j}(f)} \tag{2}$$

- For binary classification, the confusion table looks as in Figure 1. Specifically, we define
  - **True Positive (TP)** $\#\{y = \text{positive} \wedge f(\boldsymbol{x}) = \text{positive}\} = c_{1,1}(f)$;
  - **False Positive (FP)** $\#\{y = \text{negative} \wedge f(\boldsymbol{x}) = \text{positive}\} = c_{0,1}(f)$;
  - **True Negative (TN)** $\#\{y = \text{negative} \wedge f(\boldsymbol{x}) = \text{negative}\} = c_{0,0}(f)$;
  - **False Negative (FN)** $\#\{y = \text{positive} \wedge f(\boldsymbol{x}) = \text{negative}\} = c_{1,0}(f)$;

  Total positives $= TP + FN$ and total negatives $= FP + TN$.

- Given binary classficiation confusion table, define the followings
  - **True Positive Rate (TPR)**: True positive among ***all positives***

$$\text{TPR}_i = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{c_{i,i}(f)}{\sum_j c_{i,j}(f)}. \tag{3}$$

In its general form, this measure refers to the proportion of the examples of some class $i$ of interest actually assigned to class $i$ by the learning algorithm.

- **False Positive Rate (FPR)**: False positive among ***all negatives***

$$\mathrm{FPR}_i = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = \frac{\sum_{j \neq i} c_{j,i}(f)}{\sum_{j \neq i} \sum_k c_{j,k}(f)}. \tag{4}$$

Note the denominator is **all negatives** not all positives. FPRi(f ) measures the proportion of examples not belonging to class $i$ that are nonetheless erroneously classified as belonging to class $i$.

True- and false-positive rates generally form a **complement pair** of reported performance measures when the performance is *measured over the* ***positive*** class in the binary classification scenario. Moreover, we can obtain the same measures on the *negative* class as below.

- **True Negative Rate (TNR)**L True negatives among ***all negatives***

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}}. \tag{5}$$

- **False Negative Rate (FNR)**: False negatives among ***all positives***.

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}}. \tag{6}$$

Note the denominator is **all positives** not all negatives.

- In signal detection theory, the true-positive rate is also known as the ***hit rate***, whereas the false-positive rate is referred to as the ***false-alarm rate*** or the *fallout*.

- In hypothesis testing, with null hypothesis $\Theta_0$. The ***type-I error*** $\alpha$ of test $T$ is

$$\alpha := \textbf{type-I error}(T) = \mathbb{P}\left\{T(\boldsymbol{x}) \textbf{ rejects } \Theta_0 \mid \Theta_0 \text{ is true}\right\} \tag{7}$$

**FPR = Type-I error** when the null hypothesis $\Theta_0$ is ***negative***.

Similarly, the ***type-II error*** $\beta$ of test $T$ is

$$\beta := \textbf{type-II error}(T) = \mathbb{P}\left\{T(\boldsymbol{x}) \textbf{ failes to reject } \Theta_0 \mid \Theta_0 \text{ is false}\right\} \tag{8}$$

**FNR = Type-II error** when the null hypothesis $\Theta_0$ is ***negative***.

- The true-positive rate of a classifier is also referred to as the ***sensitivity*** of the classifier. For instance, it is used when investigating how sensitive the test is to the presence of the disease. The complement metric to this, in the case of the two-class scenario, would focus on the *proportion of* ***negative*** *instances* (e.g., control cases or healthy subjects) that are detected. This metric is called the ***specificity*** of the learning algorithm.

- ***sensitivity*** is defined as

$$\text{sensitivity} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = \mathrm{TPR} \tag{9}$$

– ***specificity*** is defined as

$$\text{specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} = \text{TNR} = (1 - \text{FPR}) \tag{10}$$

- An important measure related to the sensitivity and specificity of the classifier, known as the ***likelihood ratio***, aims to combine these two notions to assess the extent to which the classifier is effective in predicting the two classes. Note the ratio is the likelihood of a given test result ***under null hypothesis*** vs. the same test result ***under alternative hypothesis***.

$$\text{LR}_+ = \frac{\text{sensitivity}}{1 - \text{specificity}} \tag{11}$$

$$\text{LR}_- = \frac{1 - \text{sensitivity}}{\text{specificity}} \tag{12}$$

  In terms of probabilities, LR+ is the ratio of the probability of a positive result in people who do encounter a recurrence to the probability of a positive result in people who do not. Similarly, LR− is the ratio of the probability of a negative result in people who do encounter a recurrence to the probability of a negative result in people who do not.

  A higher positive likelihood and a lower negative likelihood mean better performance on positive and negative classes, respectively, so we want to ***maximize*** LR+ and ***minimize*** LR−.

- Another aspect of assessment is the question of *what the **proportion** of examples that **truly belong** to class $i$ is from among all the examples **assigned** to (or classified as) class $i$.* We referred to it as ***precision*** of a test:

$$\text{PPV}(f) = \text{Precision}(f) = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{c_{i,i}(f)}{\sum_j c_{j,i}(f)} = \frac{c_{i,i}(f)}{c_{\cdot,i}(f)} \tag{13}$$

  That is, this metric measures how "precise" the algorithm is when identifying the examples of a given class. It is also called ***positive predictive value (PPV)***.

  Note that **precision is not equal to true positive rate, nor false positive rate, false negative rate, true negative rate**. The TPR, FNR both assumed positive ground truth, while FPR, TNR assumes negative ground truth. However, the precision/PPV do not make assumption on fixed ground truth, instead focusing on the *postive prediction* itself. One is conditioned on $y$, another one is conditioned on $f(\boldsymbol{x})$.

  The counterpart of PPV in the binary classification scenario is the ***negative predictive value (NPV)***. PPV in this case typically refers to the class of interest (positive) whereas NPV measures the same quantity with respect to the negative (e.g., control experiments in medical applications) class.

  A concrete judgment call on the superiority of the classifier in one case or the other is almost impossible based on the two metrics of PPV and NPV alone. On the other hand, with a reliability perspective, these metrics give an insight into how ***reliable*** the class-wise predictions of a classifier is.

- **False discovery rate (FDR)** $\text{FDR} = \frac{\text{FP}}{\text{TP}+\text{FP}} = 1 - \text{PPV}$.

- In information retrieval, we introduce the term "***recall***", which is the TPR or sensitivity itself

$$\text{Recall}(f) = \text{TPR} = \text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{14}$$

So **recall is equal to true positive rate and is equal to sensitivity**.

The pair $(\mathbf{PPV}, \mathbf{TPR}) = (\mathbf{Precision}, \mathbf{Recall})$ are typical statistics of interest in domains such as information retrieval in which we are interested not only in the ***proportion of relevant information*** identified, but also in investigating the **actually relevant information** from the information ***tagged as relevant***.

- The *F measure* attempts to address the issue of convenience brought on by a single metric versus a pair of metrics. It combines precision and recall in a single metric. More specifically, the ***F measure*** is a *weighted harmonic mean* of precision and recall. For any $\alpha \in \mathbb{R}, \alpha > 0$, a general formulation of the F measure can be given as

$$\begin{aligned} F_\alpha &= \frac{(1+\alpha)\left[\text{Precision}(f) \times \text{Recall}(f)\right]}{\left[\alpha \times \text{Precision}(f)\right] + \text{Recall}(f)} \\ F_1 &= \frac{2\left[\text{Precision}(f) \times \text{Recall}(f)\right]}{\text{Precision}(f) + \text{Recall}(f)} \end{aligned} \tag{15}$$

- ***Precision at*** $k$ (referred as $P@k$) is the precision value among *top k retrieval results*. Its shortcoming is thatit fails to take into account the positions of the relevant documents among the top $k$.

- ***Average precision*** computes the average value of $p(r)$ over the interval from recall $r = 0$ to $r = 1$:

$$\text{AveP} = \int_0^1 p(r)dr = \frac{\sum_{k=1}^{n} \text{Precision}(k)\, \mathbb{1}\left\{d_k \text{ at rank } k \text{ is relevent}\right\}}{\#\text{ relevent docs}} \tag{16}$$

It is the ***area under PR-curve*** (AUC of PR). Precision($k$) is the precision at cut-off rank $k$ in the list.

- The ***interpolated precision*** $p_{\text{interp}}$ at *a certain* ***recall*** *level* $r$ is defined as the highest precision found for any recall level $r' \geq r$:

$$p_{\text{interp}}(r) := \max_{r' \geq r} p(r') \tag{17}$$

Some choose 11-***point interpolated average precision*** $\text{AveP} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1.0\}} p_{\text{interp}}(r)$.

- ***Mean average precision (MAP)*** for a set of queries is the *mean* of the *average precision* scores for ***each query*** [Manning, 2008].

$$\text{MAP} = \frac{1}{Q} \sum_{q=1}^{Q} \text{AveP}(q) \tag{18}$$

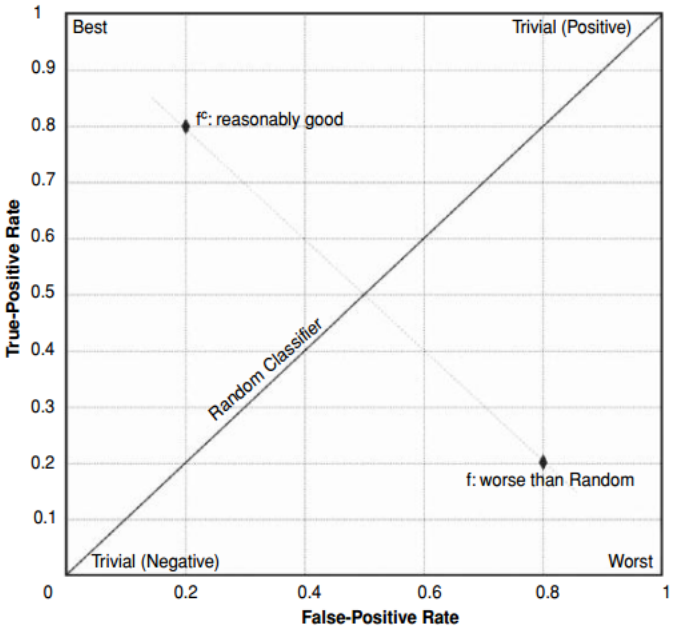


Figure 4.1. The ROC space.

**Figure 2: The ROC space**

# 2 Receiver Operating Characteristic (ROC)

## 2.1 Graphical Performance Measures

It is desirable for a performance measure not only to take into account the information contained in the confusion matrix, but also to incorporate considerations such as *skew* and *prior class distributions*. Moreover, when dealing with a scoring classifier, it is often desirable that the measure enables an assessment of classifier performance over its full operating range (of possible scores). Typically, information related to skew, cost, or prior probabilities (other than the class distribution in the training set) of the data is generally not known. Even when the asymmetric nature of the misclassification cost is known, it is not easily quantifiable. Graphical performance measures are very useful in such cases because they enable visualization of the classifier performance over the full operating range and hence under different skew ratios and class distribution priors.

Graphical performance measures has advantages when we want to discover zones of optimality, given the information over the full operating range. The question of choosing one single optimal classifier based on some quantification of the resulting graphs gives rise to measures that incorporate all the details into a scalar metric. Inevitably, compressing the information in a scalar metric results in significant information loss.

## 2.2 ROC Space

***Receiver operating characteristic (ROC)*** analysis has its origin in signal detection theory as a means to set a threshold or an operating point for the receiver to detect the presence or absence of signal. An ***ROC curve*** is a plot in which the **horizontal axis** (the x axis) denotes the *false-*

*positive rate* ***FPR*** and the **vertical axis** (the y axis) denotes the *true-positive rate* ***TPR*** of a classifier.

Note that the **TPR** = **sensitivity** of the classifier whereas the **FPR** = 1−**TNR** (TNR is the true negative rate) or equivalently 1 specificity of the classifier. Hence, in this sense, ***ROC analysis*** *studies the relationship between the* ***sensitivity*** *and the* ***specificity*** *of the classifier.* (It is like likelihood rato $LR_+$)

Because, for both the TPR and FPR, it holds that $0 \leq \text{TPR} \leq 1$ and $0 \leq \text{FPR} \leq 1$, the ROC space is a unit square, as shown in Figure 2. The output of a deterministic classifier results in a single point in this ROC space. The point $(0, 0)$ denotes a trivial classifier that *classifies* ***all*** *the instances as* ***negative*** and hence results in both the TPR and the FPR being zero. On the other end of the square, the point $(1, 1)$ corresponds to the trivial classifier that *labels* ***all*** *the instances as* ***positive*** and hence has both the TPR and the FPR values of unity.

The diagonal connecting these two points [(0, 0) and (1, 1)] has **TPR = FPR** at all the points. The classifiers falling along this diagonal can hence be considered to be ***random classifiers*** (that is, they assign positive and negative labels to the instances randomly). This resembles a biased coin toss at every point along the diagonal with bias $p = \text{TPR} = \text{FPR}$ of assigning a positive label and $1 - p$ of assigning a negative label.

The points (1, 0) and (0, 1) give the other two extremes of the ROC space. The point (1, 0) has FPR = 1 and TPR = 0, meaning that the classifier denoted by this point gets ***all its predictions wrong***. On the other hand, the point (0, 1) denotes the ideal classifier, one that gets ***all the positives right*** *and makes* ***no errors on the negatives***. The diagonal connecting these two points has **TPR = 1 - FPR = TNR**. This goes to show that the *classifiers along this diagonal perform* ***equally well on both the positive and the negative classes***.

An ***operating point*** in the ROC space corresponds to a particular **decision threshold** of the classifier that is used to assign discrete labels to the examples. As just mentioned, the instances achieving a score ***above*** the threshold are labeled *positive* whereas the ones *below* are labeled *negative*. **Each point** on the ROC space denotes a particular **TPR and FPR** for a classifier. Now, *each such point will have an associated confusion matrix* summarizing the classifier performance. Consequently an **ROC curve is a collection of various *confusion matrices* over different *varying decision thresholds*** for a classifier.

Theoretically, we can obtaint the ROC curve by tuning the decision threshold over the continuous interval between the minimum and maximum scores received by the instances in the dataset. However, this is not necessarily the case in most practical scenarios. There are two reasons:

- The **limited size** of the dataset limits the number of values on the ROC curves that can be realized. That is, when the instances are ***sorted*** in terms of the ***classification scores***, then all the decision thresholds in the ***interval of scores*** of any ***two consecutive instances*** will essentially give the same TPR and FPR on the dataset, resulting **in a single point**. The maximum number of points that can be obtained are **upper bounded by** the **number of examples** in the dataset
- This argument assumes that a ***continuous tuning*** of the decision threshold is indeed possible. This is not necessarily the case for all the scoring classifiers. Classifiers such as **decision trees**, for instance, allow for only a finite number of thresholds (**upper bounded** by the number of *possible labels* over the *leafs of the decision tree*).

Based on ROC curves, we can compare different classifiers.

- The classifiers appearing on the **left-hand side** on an ROC graph can be thought of as more ***conservative*** in their classification of positive examples. They **demand *low FPR***, **preferring misclassifying positive** examples to risking the **misclassification of negative** examples.
- The classifiers on the **right-hand side**, on the other hand, are more ***liberal*** in their classification of positive examples, meaning that they **prefer misclassifying negative** examples **to failing to recognize a positive example** as such.

This can be seen as quite a useful feature of ROC graphs because different operating points might be desired in the context of different application settings.

Each point on the ROC curve represents a different **trade-off** between the *false positives* and *false negatives* (also known as ***the cost ratio***). The cost ratio is defined by the **slope** of the line **tangent to the ROC curve** at a given point.

## 2.3 Skew Considerations

The *ROC graphs* are ***insensitive to class skews (or class imbalances)***. This is because ROC plots are measures of **TPR** and **FPR** of a classifier and do **not** take into account the actual class distributions of the positive and negative examples, unlike measures such as accuracy, empirical risk, or the F measure.

However, this observation has both a significant underlying ***assumption*** and subsequent implications. An ROC curve is based on a $2 \times 2$ confusion matrix, which has 3 degrees of freedom. The points in the **2D ROC space** hence are essentially **projections of points from a three-dimensional (3D) space**. The first two dimensions of the space correspond to the TPR and the FPR. However, the third dimension generally depends on the *specific performance measure* used to evaluate the algorithm. We can add the **class-ratio as the third dimension**. This third dimension enables us to characterize the various TPRs and FPRs that can be realized by the classifier over its entire operating range and, further, **over different class distributions**.

When considering performances on a 2D slice of the 3D ROC space, we have the ***implicit assumption*** that **TPR and the FPR are independent of the empirical (or expected) class distributions**.

More generally, the factors such as *class imbalances*, *misclassification costs*, and *credits for correct classification* can be incorporated by a single metric, ***the skew ratio***. A skew ratio $r_s$ can be utilized such that $r_s < 1$ if positive examples are deemed more important, for instance, because of a class imbalance with fewer positives in the test set compared with the negatives or because of a high misclassification cost associated with the positives.

## 2.4 Isometrics

To select an optimal operating point on the ROC curve, any performance measure can be used as long as it can be formulated in terms of the algorithms TPR and FPR. For instance, given a skew

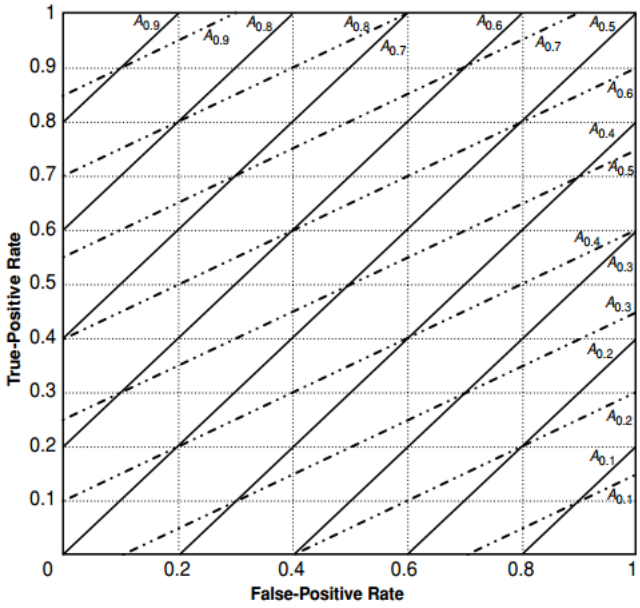


Figure 4.4. An ROC graph showing isoaccuracy lines calculated according to Equation (4.1). The subscripts following A denotes respective accuracies for $r_s = 1$ (black lines) and $r_s = 0.5$ (dash-dotted lines).

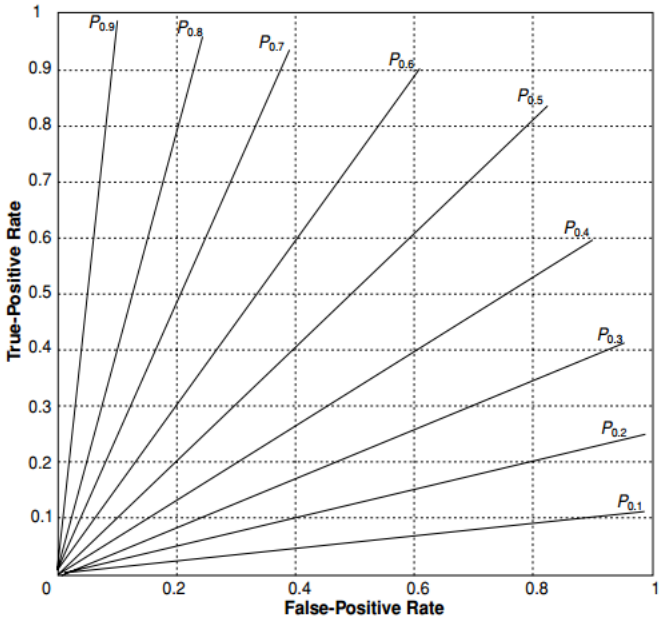


Figure 4.5. An ROC graph showing isoprecision lines calculated according to Equation (4.2) for $r_s = 1$. The subscripts following the P denote respective precisions.

**Figure 3: The isometrics for given skew ratio.**

ratio $r_s$ we can define the *(skew-sensitive) formulation of the* ***accuracy*** of classifier $f$ as

$$\text{Acc}(f) = \frac{\text{TPR}(f) + (1 - r_s)\text{FPR}(f)}{1 + r_s} \tag{19}$$

where $r_s$ is the skew ratio (class ratio)

$$r_s = \frac{\text{FP} + \text{TN}}{\text{TP} + \text{FN}} = \frac{\#\text{N}}{\#\text{P}}$$

Given the preceding definition of accuracy for a fixed $r_s$, the lines in the 2D ROC curve with the same value of accuracy are referred to as the *isoaccuracy lines*. More generally, for any performance measure or metric, such lines or curves denoting the **same metric value** for a given $r_s$ are referred to as ***isometrics*** or ***isoperformances*** lines (or curves).

We can also represent the **precision** in terms of the algorithms TPR and FPR

$$\text{Precision}(f) = \frac{\text{TPR}(f)}{\text{TPR}(f) + r_s\,\text{FPR}(f)} \tag{20}$$

Under the preceding definition of rs, we can obtain isoprecision lines on the 2D ROC graph (and surfaces in the 3D ROC space).

For a given performance measure, we can consider the **highest point** on the ROC curve that touches a given **isoperformance line** of interest (that is, with desired $r_s$) to select the desired operating point. This can be easily done by starting with the desired isoperformance line at the best classifier position in the ROC graph **(FPR = 0, TPR = 1)** and gradually sliding it down until it touches one or more points on the curve. The points thus obtained are ***the optimal performances*** of the algorithm for a desired skew ratio. We can obtain the value of the performance measure at this optimal point by looking at the ***intersection*** of the **isoperformance** line and the **diagonal** connecting the points (FPR = 0, TPR = 1) and (FPR = 1, TPR = 0).

The line for isopermission and isoaccuracy is as below

$$\text{TPR}(f) = \frac{r_s\,\text{Precision}(f)}{(1 - \text{Precision}(f))}\,\text{FPR}(f)$$
$$\text{TPR}(f) = -(1 - r_s)\text{FPR}(f) + (1 + r_s)\text{Acc}(f)$$

See Figure 3 for illustrations on isometrics.

The set of points on the ROC curve that are ***not suboptimal*** forms the ***ROC convex hull (ROCCH)***. The classifiers in the convex hull represent the ***optimal classifiers*** (under a given performance measure) for a given skew ratio. Moreover, in the case of multiple classifiers, the convex hull identifies the best classifier(s) for different operating points. The points on the ROC curve of a learning algorithm give a snapshot of the classifier performance for a given skew ratio.

## 2.5 ROC Curve Generation

The efficient implementations of the ROC curve-generation process can also be found as a standard package in many machine learning toolkits. We attached an efficient algorithm by [Fawcett, 2006] in Figure 4.

**Algorithm 1.** Efficient method for generating ROC points
**Inputs:** $L$, the set of test examples; $f(i)$, the probabilistic classifier's estimate that example $i$ is positive; $P$ and $N$, the number of positive and negative examples.
**Outputs:** $R$, a list of ROC points increasing by *fp rate*.
**Require:** $P > 0$ and $N > 0$

1: $L_{\text{sorted}} \leftarrow L$ sorted decreasing by $f$ scores
2: $FP \leftarrow TP \leftarrow 0$
3: $R \leftarrow \langle\rangle$
4: $f_{\text{prev}} \leftarrow -\infty$
5: $i \leftarrow 1$
6: **while** $i \leqslant |L_{\text{sorted}}|$ **do**
7: **if** $f(i) \neq f_{\text{prev}}$ **then**
8: push $\left(\frac{FP}{N}, \frac{TP}{P}\right)$ onto $R$
9: $f_{\text{prev}} \leftarrow f(i)$
10: **end if**
11: **if** $L_{\text{sorted}}[i]$ is a positive example **then**
12: $TP \leftarrow TP + 1$
13: **else** /* $i$ is a negative example */
14: $FP \leftarrow FP + 1$
15: **end if**
16: $i \leftarrow i + 1$
17: **end while**
18: push $\left(\frac{FP}{N}, \frac{TP}{P}\right)$ onto $R$ /* This is (1,1) */
19: **end**

**Figure 4: The algorithm to generate ROC curve [Fawcett, 2006]**

In the algorithm, we exploit the ***monotonicity*** *of thresholded classifications*: any instance that is classified *positive* with respect to a given threshold will be classified *positive* for ***all lower thresholds*** as well. $y_+ = f(x) \geq t_0 \Rightarrow y_+ = f(x) \geq t, \forall t \leq t_0$. We can simply **sort** the test instances **decreasing** by $f$ scores and move down the list, processing one instance at a time and updating TP and FP as we go. In this way an ROC graph can be created from a linear scan. Statements 7-10 need some explanation. These are necessary in order to correctly handle sequences of *equally scored instances*.

Let $n$ be the number of points in the test set. This algorithm requires an $O(n \log n)$ sort followed by an $O(n)$ scan down the list, **resulting in** $O(n \log n)$ **total complexity**.

### 2.6 Summary Statistics and the AUC

ROC curve does not allow us to quantify this comparative analysis that can facilitate decision making with regard to the suitability or preference of one classifier over others in the form of an objective ***scalar metric***. Some such representative statistics include these:

- The area ***between*** the ROC curve and the diagonal of the ROC graph connecting the points FPR = TPR = 0 and FPR = TPR = 1; It measures the performance that a learning algorithm can achieve above the random classifier along TPR = FPR.
- The ***intercept*** of the ROC curve with the diagonal connecting FPR = 1, TPR = 0 and FPR = 0, TPR = 1; This statistics signifies the *operating range* of the algorithm that yields classifiers with *lower expected cost*.
- The ***total area*** *under the ROC curve*, abbreviated as ***AUC***.

The **AUC** represents the performance of the classifier ***averaged*** over *all the possible cost ratios*. AUC($f$) $\in [0, 1]$, with the upper bound attained for a perfect classifier (one with TPR = 1 and FPR = 0). The ***random*** *classifier* represented by the diagonal cuts the ROC space in half and hence AUC($f_{\text{random}}$) = 0.5. On the other hand, if the classifier assigns the ***same score to all examples***, whether *negative* or *positive*, we would obtain classifiers along the diagonal **TPR = FPR**. We can also obtain a similar curve if the classifier *assigns similar distributions* of the score.

**Another interpretation** of an **AUC** can be obtained for **ranking classifiers** in that AUC represents the **ability** of a classifier to rank a *randomly* chosen **positive** test example **higher than** a negative one. In this respect, this is shown to **be equivalent to** ***Wilcoxon's Rank Sum test*** (also known as the ***Mann-Whitney U test***). With regard to the ***Gini coefficient (Gini)***, a measure of statistical dispersion popular in economics, it has been shown that

$$\text{AUC} = \frac{(Gini + 1)}{2}$$

.

A simpler algorithm to estimate AUC is to use Wilcoxons Rank Sum statistic. We first **rank** the scores for each test instances **in decreasing order**. Then we can calculate the AUC as

$$\text{AUC}(f) = \frac{\sum_i^{|T_p|}(R_i - i)}{|T_p|\,|T_n|}, \tag{21}$$

where $T_p \subset T$ and $T_n \subset T$ are, respectively, the subsets of *positive* and *negative* examples in test

**Algorithm 2.** Calculating the area under an ROC curve
**Inputs:** $L$, the set of test examples; $f(i)$, the probabilistic classifier's estimate that example $i$ is positive; $P$ and $N$, the number of positive and negative examples.
**Outputs:** $A$, the area under the ROC curve.
**Require:** $P > 0$ and $N > 0$

```
1: L_sorted ← L sorted decreasing by f scores
2: FP ← TP ← 0
3: FP_prev ← TP_prev ← 0
4: A ← 0
5: f_prev ← −∞
6: i ← 1
7: while i ⩽ |L_sorted| do
8:    if f(i) ≠ f_prev then
9:       A ← A + TRAPEZOID_AREA(FP, FP_prev,
               TP, TP_prev)
10:      f_prev ← f(i)
11:      FP_prev ← FP
12:      TP_prev ← TP
13:   end if
14:   if i is a positive example then
15:      TP ← TP + 1
16:   else      /* i is a negative example */
17:      FP ← FP + 1
18:   end if
19:   i ← i + 1
20: end while
21: A ← A + TRAPEZOID_AREA(N, FP_prev, N, TP_prev)
22: A ← A/(P × N)   /* scale from P × N onto the unit
    square */
23: end
 1: function TRAPEZOID_AREA(X1, X2, Y1, Y2)
 2: Base ← |X1 − X2|
 3: Height_avg ← (Y1 + Y2)/2
 4: return Base × Height_avg
 5: end function
```

**Figure 5: The algorithm to compute AUC of ROC curve [Fawcett, 2006]**

set $T$ , and $R_i$ is the rank of the $i$-th example in $T_p$ given by classifier $f$. Computing AUC can be done simiar to computing ROC, which is $O(n \log n)$.

As other single-metric performance measures, AUC misses information on ***concavities*** *in the performance*, or ***trade-off behaviors*** between the true-positive and the false-positive performances.

Some criticisms have also appeared warning against the use of AUC across classifiers **for comparative purposes**.

- One of the most obvious is that, because the classifiers are typically optimized to obtain the best performance (in context of the given performance measure), the ROC curves thus obtained in the two cases would be similar. This then would yield *uninformative AUC differences.*
- Further, if the **ROC curves** of the two classifiers ***intersect***, the AUC-based comparison between the classifiers can be relativly uninformative and even **misleading**.
- However, a more serious limitation of the AUC for comparative purposes lies in the fact that

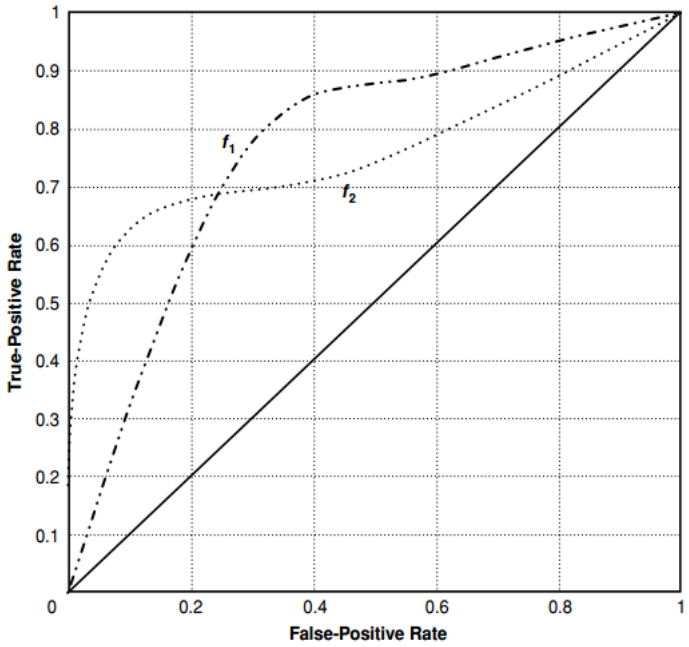


Figure 4.9. The ROC curves for two hypothetical scoring classifiers $f_1$ and $f_2$, in which a single classifier is not strictly dominant throughout the operating range.

**Figure 6: The comparison of two algorithms in ROC curve [Japkowicz and Shah, 2011]**

the *misclassification* ***cost*** *distributions* (and hence the **skew-ratio** distributions) used by the AUC are different for different classifiers.

In the event the ***ranking property*** **of the classifier** is important (for instance, in information-retrieval systems), AUC can be a more reliable measure of classifier performance than measures such as accuracy because it **assesses the ranking capability** of the classifier in a direct manner.

## 2.7 Calibration

The classifiers' thresholds based on the training set may or may not reflect the empirical realizations of labelings in the test set. That is, if no example obtains a score in the interval $[t_1, t_2] \subset I$, then no threshold in the interval $[t_1, t_2]$ will yield a different point in the ROC space. One solution to deal with this problem is ***calibration***. All the scores in the interval $[t_1, t_2]$ can be mapped to the fraction of the positive instances obtained as a result of assigning any score in this interval, i.e. *linear interpolation*.

This is a workable solution as long as there are *no* ***concavities*** *in the ROC curve*. **Concavity** in the curve means that there are ***skew ratios*** for which the classifier is ***suboptimal***. This essentially means that **better classifier** performance can be obtained for the skew ratios lying in the *concave region* of the curve although the empirical estimates do not suggest this. See Figure 6. In the case of concavities, the behavior of the calibrated scores does not mimic the desired behavior of the slope of the threshold interval. The classifier obtained over ***calibrated*** scores can **overfit** the data, resulting in poor generalization. One can use **isotonic regression** to map the scores corresponding to the *concave interval* $[t_1, t_2]$ to an *unbiased* estimate of the slope of the line segment connecting the two points corresponding to the thresholds $t_1$ and $t_2$.

# 3 Other Visual Analysis Methods

## 3.1 Precision-Recall (PR) Curves

***Precision-recall Curves***, sometimes abbreviated as ***PR curves***, are similar to ROC curves and lift charts in that they explore the trade-off between the wellclassified positive examples and the number of misclassified negative examples. As the name suggests, PR curves plot the **precision** of the classifier as **a function of its recall**.

The curves look **different** from ROC curves and lift curves because they have a ***negative slope***. This is because precision **decreases** as recall **increases**. PR curves can sometimes be more appropriate than the ROC curves in the events of ***highly imbalanced data*** [Davis and Goadrich, 2006].

## 3.2 Lift Charts

***Lift charts*** are a performance visualization technique closely related to the ROC curves. Lift charts plot the **true positives** against the **dataset size** required to **achieve** this number of true positives. That is, the ***vertical axis*** of the lift charts plots ***the true positives*** (and **not the TPR**) whereas the ***horizontal axis*** denotes the **number of examples** in the dataset considered for the specific true positives on the vertical axis.

In other words, the ROC curve counts the number of negative examples that have slipped into the data sample for which the classifier issued a particular true-positive rate, whereas the lift chart **counts both the positive and the negative examples** in that set. In **highly imbalanced datasets**, in which, typically the number of positive examples is much smaller than that of negative examples, the horizontal axes of lift charts and ROC curves look **very similar** as do the curves.

## References


Jesse Davis and Mark Goadrich. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

Tom Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

Nathalie Japkowicz and Mohak Shah. *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.

Christopher D Manning. *Introduction to information retrieval*. Syngress Publishing,, 2008.