12
Measuring the Performance of a Classifier

Up to now we have generally assumed that the best (or only) way of measuring the performance of a classifier is by its *predictive accuracy*, i.e. the proportion of unseen instances it correctly classifies. However this is not necessarily the case.

There are many other types of classification algorithm as well as those discussed in this book. Some require considerably more computation or memory than others. Some require a substantial number of training instances to give reliable results. Depending on the situation the user may be willing to accept a lower level of predictive accuracy in order to reduce the run time/memory requirements and/or the number of training instances needed.

A more difficult trade-off occurs when the classes are severely unbalanced. Suppose we are considering investing in one of the leading companies quoted on a certain stock market. Can we predict which companies will become bankrupt in the next two years, so we can avoid investing in them? The proportion of such companies is obviously small. Let us say it is 0.02 (a fictitious value), so on average out of every 100 companies 2 will become bankrupt and 98 will not. Call these 'bad' and 'good' companies respectively.

If we have a very 'trusting' classifier that always predicts 'good' under all circumstances its predictive accuracy will be 0.98, a very high value. Looked at only in terms of predictive accuracy this is a very successful classifier. Unfortunately it will give us no help at all in avoiding investing in bad companies.

On the other hand, if we want to be very safe we could use a very 'cautious' classifier that always predicted 'bad'. In this way we would never lose our money in a bankrupt company but would never invest in a good one either. This is

M. Bramer, *Principles of Data Mining*, Undergraduate Topics in Computer Science, DOI 10.1007/978-1-4471-7307-6_12

similar to the ultra-safe strategy for air traffic control: ground all aeroplanes, so you can be sure that none of them will crash. In real life, we are usually willing to accept the risk of making some mistakes in order to achieve our objectives.

It is clear from this example that neither the very trusting nor the very cautious classifier is any use in practice. Moreover, where the classes are severely unbalanced (98% to 2% in the company example), predictive accuracy on its own is not a reliable indicator of a classifier's effectiveness.

12.1 True and False Positives and Negatives

The idea of a confusion matrix was introduced in Chapter 7. When there are two classes, which we will call positive and negative (or simply + and −), the confusion matrix consists of four cells, which can be labelled TP, FP, FN and TN as in Figure 12.1.

		Predicted class		Total instances
		+	−	
Actual class	+	TP	FN	P
	−	FP	TN	N

Figure 12.1 True and False Positives and Negatives

TP: true positives
The number of positive instances that are classified as positive

FP: false positives
The number of negative instances that are classified as positive

FN: false negatives
The number of positive instances that are classified as negative

TN: true negatives
The number of negative instances that are classified as negative

P = TP + FN
The total number of positive instances

N = FP + TN
The total number of negative instances

It is often useful to distinguish between the two types of classification error: false positives and false negatives.

False positives (also known as *Type 1 Errors*) occur when instances that should be classified as negative are classified as positive.

False negatives (also known as *Type 2 Errors*) occur when instances that should be classified as positive are classified as negative.

Depending on the application, errors of these two types are of more or less importance.

In the following examples we will make the assumption that there are only two classifications, which will be called positive and negative, or + and −. The training instances can then be considered as positive and negative examples of a concept such as ‘good company’, ‘patient with brain tumour’ or ‘relevant web page’.

Bad Company Application. Here we would like the number of false positives (bad companies that are classified as good) to be as small as possible, ideally zero. We would probably be willing to accept a high proportion of false negatives (good companies classified as bad) as there are a large number of possible companies to invest in.

Medical Screening Application. It would not be possible in any realistic system of healthcare to screen the entire population for a condition that occurs only rarely, say a brain tumour. Instead the doctor uses his or her experience to judge (based on symptoms and other factors) which patients are most likely to be suffering from a brain tumour and sends them to a hospital for screening.

For this application we might be willing to accept quite a high proportion of false positives (patients screened unnecessarily) perhaps as high as 0.90, i.e. only 1 in 10 of patients screened has a brain tumour, or even higher. However we would like the proportion of false negatives (patients with a brain tumour who are not screened) to be as small as possible, ideally zero.

Information Retrieval Application. A web search engine can be looked at as a kind of classifier. Given a specification such as ‘pages about American poetry’ it effectively classifies all pages on the web that are known to it as either ‘relevant’ or ‘not relevant’ and displays the URLs of the ‘relevant’ ones to the user. Here we may be willing to accept a high proportion of false negatives (relevant pages left out), perhaps 30% or even higher, but probably do not want too many false positives (irrelevant pages included), say no more than 10%. In such information retrieval applications the user is seldom aware of the false negatives (relevant pages not found by the search engine) but false positives are visible, waste time and irritate the user.

These examples illustrate that, leaving aside the ideal of perfect classification accuracy, there is no single combination of false positives and false negatives that is ideal for every application and that even a very high level of predictive accuracy may be unhelpful when the classes are very unbalanced. To go further we need to define some improved measures of performance.

12.2 Performance Measures

We can now define a number of performance measures for a classifier applied to a given test set. The most important ones are given in Figure 12.2. Several measures have more than one name, depending on the technical area (signal processing, medicine, information retrieval etc.) in which they are used.

For information retrieval applications the most commonly used measures are Recall and Precision. For the search engine application, Recall measures the proportion of relevant pages that are retrieved and Precision measures the proportion of retrieved pages that are relevant. The F1 Score combines Precision and Recall into a single measure, which is their product divided by their average. This is known as the *harmonic mean* of the two values.

The values of P and N, the number of positive and negative instances, are fixed for a given test set, whichever classifier is used. The values of the measures given in Figure 12.2 will generally vary from one classifier to another. Given the values of True Positive Rate and False Positive Rate (as well as P and N) we can derive all the other measures.

We can therefore characterise a classifier by its True Positive Rate (TP Rate) and False Positive Rate (FP Rate) values, which are both proportions from 0 to 1 inclusive. We start by looking at some special cases.

<u>A: The Perfect Classifier</u>

Here every instance is correctly classified. $TP = P$, $TN = N$ and the confusion matrix is:

		Predicted class		Total instances
		+	−	
Actual class	+	P	0	P
	−	0	N	N

True Positive Rate or Hit Rate or Recall or Sensitivity or TP Rate	TP/P	The proportion of positive instances that are correctly classified as positive
False Positive Rate or False Alarm Rate or FP Rate	FP/N	The proportion of negative instances that are erroneously classified as positive
False Negative Rate or FN Rate	FN/P	The proportion of positive instances that are erroneously classified as negative = 1 − True Positive Rate
True Negative Rate or Specificity or TN Rate	TN/N	The proportion of negative instances that are correctly classified as negative
Precision or Positive Predictive Value	TP/(TP+FP)	Proportion of instances classified as positive that are really positive
F1 Score	(2 × Precision × Recall) /(Precision + Recall)	A measure that combines Precision and Recall
Accuracy or Predictive Accuracy	(TP + TN)/(P + N)	The proportion of instances that are correctly classified
Error Rate	(FP + FN)/(P + N)	The proportion of instances that are incorrectly classified

Figure 12.2 Some Performance Measures for a Classifier

TP Rate (Recall) $= P/P = 1$
FP Rate $= 0/N = 0$
Precision $= P/P = 1$
F1 Score $= 2 \times 1/(1 + 1) = 1$
Accuracy $= (P + N)/(P + N) = 1$

B: The Worst Possible Classifier

Every instance is wrongly classified. $TP = 0$ and $TN = 0$. The confusion matrix is:

		Predicted class		Total instances
		+	−	
Actual class	+	0	P	P
	−	N	0	N

TP Rate (Recall) $= 0/P = 0$
FP Rate $= N/N = 1$
Precision $= 0/N = 0$
F1 Score is not applicable (as Precision + Recall = 0)
Accuracy $= 0/(P + N) = 0$

C: The Ultra-liberal Classifier

This classifier always predicts the positive class. The True Positive rate is 1 but the False Positive rate is also 1. The False Negative and True Negative rates are both zero. The confusion matrix is:

		Predicted class		Total instances
		+	−	
Actual class	+	P	0	P
	−	N	0	N

TP Rate (Recall) $= P/P = 1$
FP Rate $= N/N = 1$
Precision $= P/(P + N)$
F1 Score $= 2 \times P/(2 \times P + N)$
Accuracy $= P/(P+N)$, which is the proportion of positive instances in the test set.

D: The Ultra-conservative Classifier

This classifier always predicts the negative class. The False Positive rate is zero, but so is the True Positive rate. The confusion matrix is:

		Predicted class		Total instances
		+	−	
Actual class	+	0	P	P
	−	0	N	N

TP Rate (Recall) $= 0/P = 0$
FP Rate $= 0/N = 0$
Precision is not applicable (as $TP + FP = 0$)
F1 Score is also not applicable
Accuracy $= N/(P + N)$, which is the proportion of negative instances in the test set.

12.3 True and False Positive Rates versus Predictive Accuracy

One of the strengths of characterising a classifier by its TP Rate and FP Rate values is that they do not depend on the relative sizes of P and N. The same applies to using the FN Rate and TN Rate values or any other combination of two 'rate' values calculated from *different* rows of the confusion matrix. In contrast, Predictive Accuracy and all the other measures listed in Figure 12.2 are derived from values in *both* rows of the table and so are affected by the relative sizes of P and N, which can be a serious weakness.

To illustrate this, suppose that the positive class corresponds to those who pass a driving test at the first attempt and that the negative class corresponds to those who fail. Assume that the relative proportions in the real world are 9 to 10 (a fictitious value) and the test set correctly reflects this.

Then the confusion matrix for a particular classifier on a given test set might be

		Predicted class		Total instances
		+	−	
Actual class	+	8,000	1,000	9,000
	−	2,000	8,000	10,000

This gives a true positive rate of 0.89 and a false positive rate of 0.2, which we will assume is a satisfactory result.

Now suppose that the number of successes grows considerably over a period of time because of improved training, so that there is a higher proportion of passes. With this assumption a possible confusion matrix for a future series of trials would be as follows.

		Predicted class		Total instances
		+	−	
Actual class	+	80,000	10,000	90,000
	−	2,000	8,000	10,000

The classifier will of course still work exactly as well as before to predict the correct classification of either a pass or a fail with which it is presented. For both confusion matrices the values of TP Rate and FP Rate are the same (0.89 and 0.2 respectively). However the values of the Predictive Accuracy measure are different.

For the original confusion matrix, Predictive Accuracy is 16,000/19,000 = 0.842. For the second one, Predictive Accuracy is 88,000/100,000 = 0.88.

An alternative possibility is that over a period of time there is a large increase in the relative proportion of failures, perhaps because of an increase in the number of younger people being tested. A possible confusion matrix for a future series of trials would be as follows.

		Predicted class		Total instances
		+	−	
Actual class	+	8,000	1,000	9,000
	−	20,000	80,000	100,000

Here the Predictive Accuracy is 88,000/109,000 = 0.807.

Whichever of these test sets was used with the classifier the TP Rate and FP Rate values would be the same. However the three Predictive Accuracy values would vary from 81% to 88%, reflecting changes in the relative numbers of positive and negative values in the test set, rather than any change in the quality of the classifier.

12.4 ROC Graphs

The TP Rate and FP Rate values of different classifiers on the same test set are often represented diagrammatically by a *ROC Graph*. The abbreviation ROC Graph stands for 'Receiver Operating Characteristics Graph', which reflects its original uses in signal processing applications.

On a ROC Graph, such as Figure 12.3, the value of FP Rate is plotted on the horizontal axis, with TP Rate plotted on the vertical axis.

Each point on the graph can be written as a pair of values (x, y) indicating that the FP Rate has value x and the TP Rate has value y.

The points $(0, 1)$, $(1, 0)$, $(1, 1)$ and $(0, 0)$ correspond to the four special cases A, B, C and D in Section 12.2, respectively. The first is located at the best possible position on the graph, the top left-hand corner. The second is at the worst possible position, the bottom right-hand corner. If all the classifiers are good ones, all the points on the ROC Graph are likely to be around the top left-hand corner.

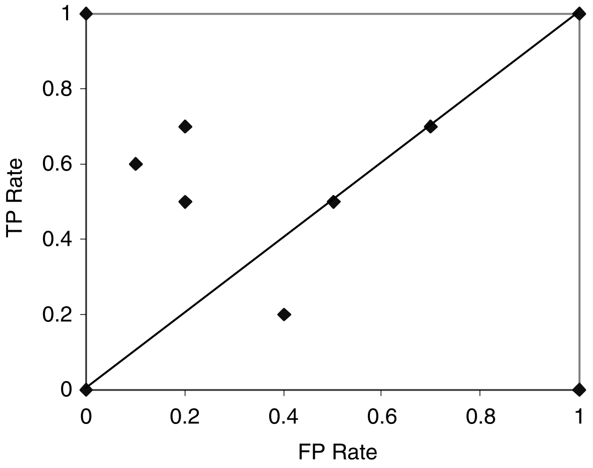

Figure 12.3 Example of ROC Graph

The other six points shown are $(0.1, 0.6)$, $(0.2, 0.5)$, $(0.4, 0.2)$, $(0.5, 0.5)$, $(0.7, 0.7)$ and $(0.2, 0.7)$.

One classifier is better than another if its corresponding point on the ROC Graph is to the 'north-west' of the other's. Thus the classifier represented by $(0.1, 0.6)$ is better than the one represented by $(0.2, 0.5)$. It has a lower FP Rate and a higher TP Rate. If we compare points $(0.1, 0.6)$ and $(0.2, 0.7)$, the latter has a higher TP Rate but also a higher FP Rate. Neither classifier is superior to the other on both measures and the one chosen will depend on the relative importance given by the user to the two measures.

The diagonal line joining the bottom left and top right-hand corners corresponds to random guessing, whatever the probability of the positive class may be. If a classifier guesses positive and negative at random with equal frequency, it will classify positive instances correctly 50% of the time and negative instances as positive, i.e. incorrectly, 50% of the time. Thus both the TP Rate and the FP Rate will be 0.5 and the classifier will lie on the diagonal at point $(0.5, 0.5)$.

Similarly, if a classifier guesses positive and negative at random with positive selected 70% of the time, it will classify positive instances correctly 70% of the time and negative instances as positive, i.e. incorrectly, 70% of the time. Thus both the TP Rate and the FP Rate will be 0.7 and the classifier will lie on the diagonal at point $(0.7, 0.7)$.

We can think of the points on the diagonal as corresponding to a large number of random classifiers, with higher points on the diagonal corresponding to higher proportions of positive classifications generated on a random basis.

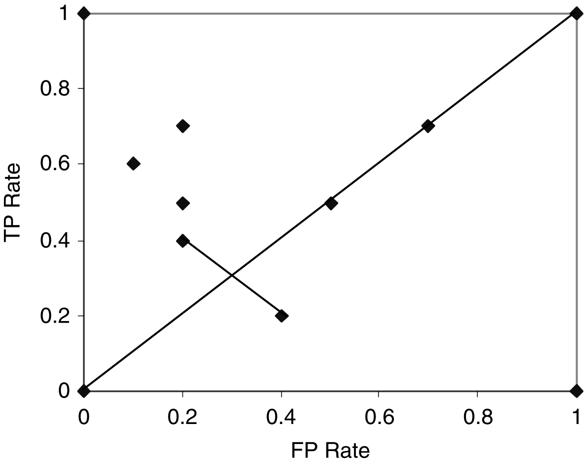

Figure 12.4 Example of ROC Graph (Amended)

The upper left-hand triangle corresponds to classifiers that are better than random guessing. The lower right-hand triangle corresponds to classifiers that are worse than random guessing, such as the one at $(0.4, 0.2)$.

A classifier that is worse than random guessing can be converted to one that is better than random guessing simply by reversing its predictions, so that every positive prediction becomes negative and vice versa. By this method the classifier at $(0.4, 0.2)$ can be converted to the new one at $(0.2, 0.4)$ in Figure 12.4. The latter point is the former reflected about the diagonal line.

12.5 ROC Curves

In general, each classifier corresponds to a single point on a ROC Graph. However there are some classification algorithms that lend themselves to 'tuning', so that it is reasonable to think of a *series* of classifiers, and thus points on a ROC Graph, one for each value of some variable, generally known as a *parameter*. For a decision tree classifier such a parameter might be the 'depth cutoff' (see Chapter 9) which can vary from 1, 2, 3 etc.

In such a case the points can be joined to form a *ROC Curve* such as Figure 12.5.

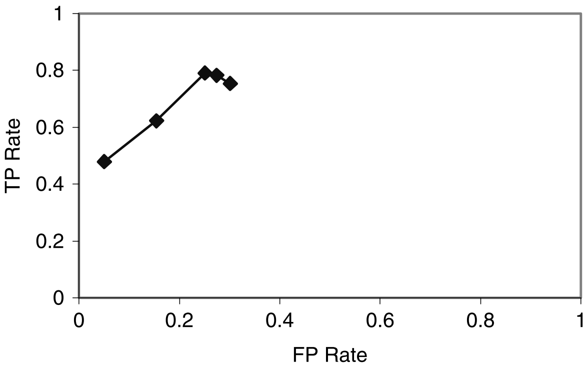

Figure 12.5 Example of ROC Curve

Examining ROC curves can give insights into the best way of tuning a classification algorithm. In Figure 12.5 performance clearly degrades after the third point in the series.

The performance of different types of classifier with different parameters can be compared by inspecting their ROC curves.

12.6 Finding the Best Classifier

There is no infallible way of finding the best classifier for a given application, unless we happen to find one that gives perfect performance, corresponding to the $(0, 1)$ point on the ROC Graph. One approach that is sometimes used is to measure the distance of a classifier on the ROC Graph from the perfect classifier.

Figure 12.6 shows the points $(\mathit{fprate}, \mathit{tprate})$ and $(0, 1)$. The Euclidean distance between them is $\sqrt{\mathit{fprate}^2 + (1 - \mathit{tprate})^2}$.

We can write $\mathit{Euc} = \sqrt{\mathit{fprate}^2 + (1 - \mathit{tprate})^2}$.

The smallest possible value of Euc is zero, when $\mathit{fprate} = 0$ and $\mathit{tprate} = 1$ (the perfect classifier). The largest value is $\sqrt{2}$, when fprate is 1 and tprate is zero (the worst possible classifier). We could hypothesise that the smaller the value of Euc the better the classifier.

Euc is a useful measure but does not take into account the relative importance of true and false positives. There is no best answer to this. It depends on the use to which the classifier will be put.

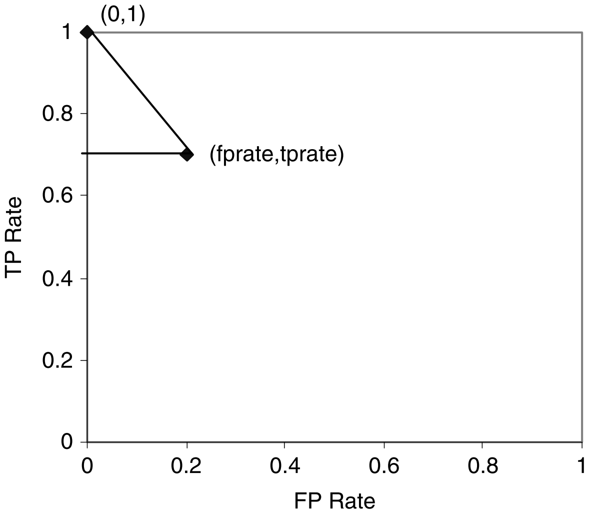

Figure 12.6 Measuring the Distance from the Perfect Classifier

We can specify the relative importance of making *tprate* as close to 1 as possible and making *fprate* as close to zero as possible by a weight w from 0 to 1 and define the *Weighted Euclidean Distance* as

$$WEuc = \sqrt{(1-w)fprate^2 + w(1-tprate)^2}$$

If $w = 0$ this reduces to $WEuc = fprate$, i.e. we are only interested in minimising the value of *fprate*.

If $w = 1$ it reduces to $WEuc = 1 - tprate$, i.e. we are only interested in minimising the difference between *tprate* and 1 (thus maximising *tprate*).

If $w = 0.5$ the formula reduces to

$$WEuc = \sqrt{0.5 * fprate^2 + 0.5 * (1-tprate)^2}$$

which is a constant multiple of

$\sqrt{fprate^2 + (1-tprate)^2}$, so the effect when comparing one classifier with another is the same as if there were no weighting at all.

12.7 Chapter Summary

This chapter looks at the use of true and false positive and negative classifications as a better way of measuring the performance of a classifier than predictive accuracy alone. Other performance measures can be derived from these four basic ones, including *true positive rate* (or hit rate), *false positive rate* (or false alarm rate), *precision*, *accuracy* and *F1 score*.

The values of true positive rate and false positive rate are often represented diagrammatically by a *ROC graph*. Joining the points on a ROC graph to form a ROC curve can often give insight into the best way of tuning a classifier. A Euclidean distance measure of the difference between a given classifier and the performance of a hypothetical perfect classifier is described.

12.8 Self-assessment Exercise for Chapter 12

Four classifiers are generated for the same training set, which has 100 instances. They have the following confusion matrices.

		Predicted class	
		+	−
Actual class	+	50	10
	−	10	30

		Predicted class	
		+	−
Actual class	+	55	5
	−	5	35

		Predicted class	
		+	−
Actual class	+	40	20
	−	1	39

		Predicted class	
		+	−
Actual class	+	60	0
	−	20	20

Calculate the values of true positive rate and false positive rate for each classifier and plot them on a ROC graph. Calculate the value of the Euclidean distance measure *Euc* for each one. Which classifier would you consider the best if you were equally concerned with avoiding false positive and false negative classifications?