

COMP9417: Machine Learning and Data Mining

A note on the two-class confusion matrix, lift charts and ROC curves

Session 1, 2003

Acknowledgement

The material in this supplementary note is based in part on Chapter 5, Section 7 of Witten and Frank [2] and the paper by Fawcett [1].

1 Two-class confusion matrix

This may be used to summarise the predictive performance of a classifier on test data. It is commonly encountered in a two-class format, but can be generated for any number of classes. Suppose we have a two-class problem with classes referred to as *positive* and *negative*. A single prediction by a classifier can have four outcomes which are displayed in the confusion matrix of Table 1.

		Predicted Class	
		<i>positive</i>	<i>negative</i>
Actual Class	<i>positive</i>	true positive	false negative
	<i>negative</i>	false positive	true negative

Table 1: Two-class confusion matrix (also known as a 2×2 contingency table).

true positive the actual class of the test instance is *positive* and the classifier correctly predicts the class as *positive*

false negative the actual class of the test instance is *positive* but the classifier incorrectly predicts the class as *negative*

false positive the actual class of the test instance is *negative* but the classifier incorrectly predicts the class as *positive*

true negative the actual class of the test instance is *negative* and the classifier correctly predicts the class as *negative*

For a particular data set the confusion matrix will contain in its cells the number of instances for each of the four possible classification outcomes. Note that the “correct” predictions lie on the main diagonal of the matrix while the “incorrect” predictions are

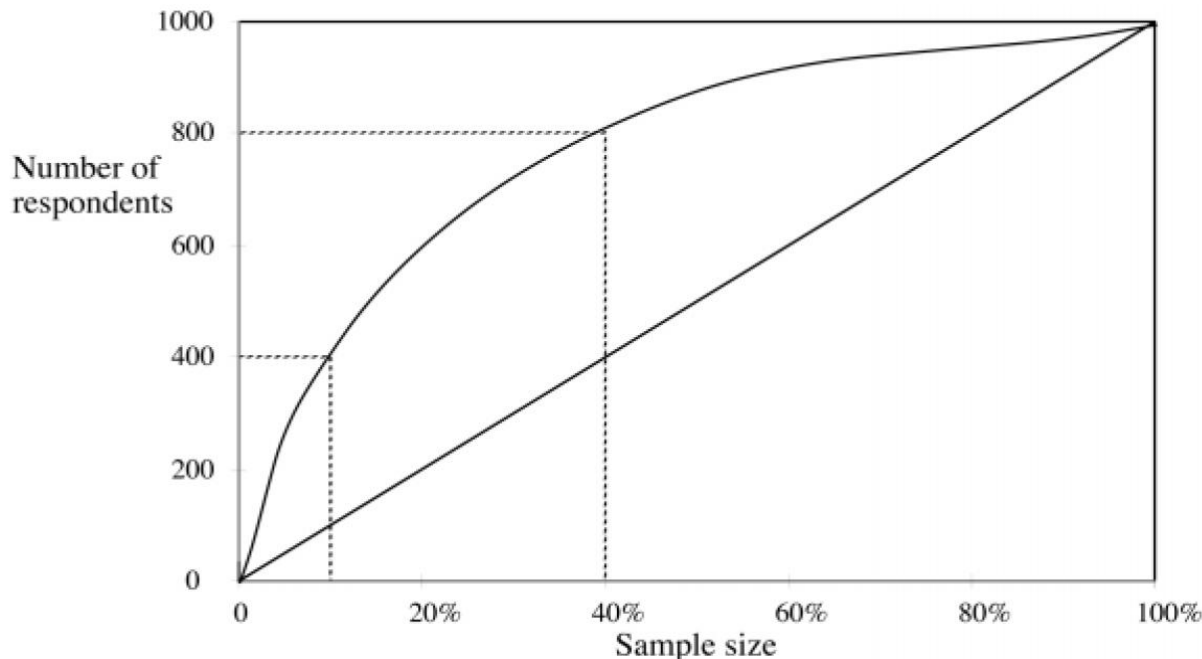


Figure 1: A lift chart. The diagonal indicates the expected success rate (number of true positives predicted for a sample) which would be obtained for random samples. The two cases discussed in the text are marked on the curve from our fictional data set: where 10% of the recipients contain 400 respondents, and 40% of the recipients contain 800 respondents.

in the off-diagonal cells. We can obtain the familiar measure of predictive accuracy of a classifier by

$$\text{accuracy} = \frac{\mathbf{tp} + \mathbf{tn}}{\mathbf{tp} + \mathbf{fn} + \mathbf{fp} + \mathbf{tn}} \times 100\%$$

where **tp** stands for **true positive**, **fn** stands for **false negative**, **fp** stands for **false positive** and **tn** stands for **true negative**. The predictive accuracy of a two-class classifier is maximised by maximising the sum of the true positives and true negatives as a proportion of the total number of instances.

1.1 Costs

Real-world applications usually have several kinds of attached cost. For example, simply collecting example data may be expensive. Alternatively, sample instances may be easily obtained whereas applying *classification labels* for some target function to this data may be much more costly. These costs are “up-front” since they apply before any machine learning method can be applied. However, there are also issues to do with the cost of using a hypothesis constructed by machine learning in order to do something. For example, the runtime efficiency of a classifier in making predictions may be critical in a real-time system. Another common situation which occurs in many applications is that misclassification *costs* can be different for false positives and false negatives. It may also happen that the classification *benefits* are different for true positives and true negatives.

Unfortunately, bear in mind that it may not always be possible to quantify accurately the actual costs for each situation in the confusion matrix.

2 Lift charts

This is a method of evaluating performance based on identifying subsets of test data on which a classifier can predict with better accuracy relative to its performance on the entire set of test data. It is usually associated with problems from marketing.

The *lift factor* is defined as the success rate in predicting positives on a subset of the data as a proportion of the success rate in predicting positives on all of the data.

$$\text{Lift factor} = \frac{\frac{tp_S}{N_S}}{\frac{tp_T}{N_T}}$$

where tp_S and N_S are the number of true positives and the size of the sample S , and tp_T and N_T are the number of true positives and the size of the test set T .

For example, suppose we have a data set from a mail-out marketing campaign. In this data set there are 1000 respondents to mail which was sent to 1000000 recipients. Now we learn a classifier which can identify a subset of 100000 recipients in which there are 400 respondents.

$$\begin{aligned} \text{Lift factor} &= \frac{\frac{400}{100000}}{\frac{1000}{1000000}} \\ &= \frac{0.004}{0.001} \\ &= 4.0 \end{aligned}$$

In marketing terminology this lift factor indicates the potential increase in response rate. If costs are known, it can be calculated whether it is better to mail to the complete list of recipients or to the subset identified.

Now suppose that another classifier is learned, with slightly different parameters, identifying a subset of 400000 recipients of whom 800 were respondents. The lift factor here is two. We can see that in fact there may be many subsets of the entire test set, and it would be of interest to know the lift factor for them.

An efficient way of calculating the lift factor for different subsets of the data is possible when a classifier is able to produce not only a predicted class label for a test instance but also a measure of *confidence*, such as a probability, in its prediction. In such cases we simply sort the data set in decreasing order of confidence of prediction for each instance of the class of interest, say the positive class. Then we can find a subset of a given size N with the greatest possible proportion of positive instances by simply taking the first N instances in the sorted list. The lift factor can then be calculated as above by taking the number of true positives from the sample of size N .

By repeating this process for each of the first k instances, from zero to the total number of instances we can obtain the success rate (number of true positives), for each such sample. Graphing the number of true positives on the y -axis against the proportion of the sample to the total number of instances on the x -axis gives a *lift chart*. A hypothetical lift chart is shown in Figure 1. By inspection it is clear that results in or close to the top

Rank	Confidence	Actual class	Rank	Confidence	Actual class
1	0.95	<i>pos</i>	11	0.77	<i>neg</i>
2	0.93	<i>pos</i>	12	0.76	<i>pos</i>
3	0.93	<i>neg</i>	13	0.73	<i>pos</i>
4	0.88	<i>pos</i>	14	0.65	<i>neg</i>
5	0.86	<i>pos</i>	15	0.63	<i>pos</i>
6	0.85	<i>pos</i>	16	0.58	<i>neg</i>
7	0.82	<i>pos</i>	17	0.56	<i>pos</i>
8	0.80	<i>pos</i>	18	0.49	<i>neg</i>
9	0.80	<i>neg</i>	19	0.48	<i>pos</i>
10	0.79	<i>pos</i>

Table 2: A set of predictions for a hypothetical two-class test set, ranked in decreasing order of the confidence of prediction for the *positive* class, also showing the actual class for each instance.

left corner of a lift chart are the most desirable, since these indicate close to a maximum success rate from those instances ranked most likely to be true positives by our learned classifier.

3 ROC curves

The method of ROC (“Receiver Operating Characteristic”) analysis is closely related to lift charts. It provides techniques for visualizing classifier performance, and evaluating and comparing algorithms. However, in this note we consider only the first of these applications. More details can be found in [1].

ROC analysis originated in signal detection theory to depict the trade-off between so-called “hit rate” and “false alarm rate”. For the purposes of machine learning these correspond to the **true positive** rate and the **false positive** rate, respectively, of classifiers. We define these as follows.

$$\text{true positive rate} = \frac{\text{true positives}}{\text{total no. of positives}}$$

$$\text{false positive rate} = \frac{\text{false positives}}{\text{total no. of negatives}}$$

As for lift charts, we approach the analysis of classifier performance by first sorting its test set predictions in decreasing order of confidence (or probability). Note that, if we have a classifier which outputs only class predictions, such as a decision tree, it is usually possible to extract a “confidence” measure as well (in the case of decision trees, we could use the proportion of true positives out of all of the instances at a leaf node). Also, for a two-class case we can find the confidence of a prediction of the positive class given a confidence for the negative class by subtracting it from the maximum confidence. Table 3 contains a hypothetical set of ranked predictions. Here the “confidence” is a probability.

Note that in Table 3 we can see the distribution of true positives throughout the list

Algorithm Graph of ROC curve

Input: list of test examples T sorted in descending order of confidence

begin

Let $K = |T|$ be the number of test examples

Let P be the number of positive examples in T

Let N be the number of negative examples in T

$TP := 0$

$FP := 0$

for $i = 1$ to K **do**

if example $T[i]$ is positive **then**

$TP := TP + 1$

else

$FP := FP + 1$

endif

 output point on ROC curve $\langle \frac{TP}{P}, \frac{FP}{N} \rangle$

$i := i + 1$

endfor

end

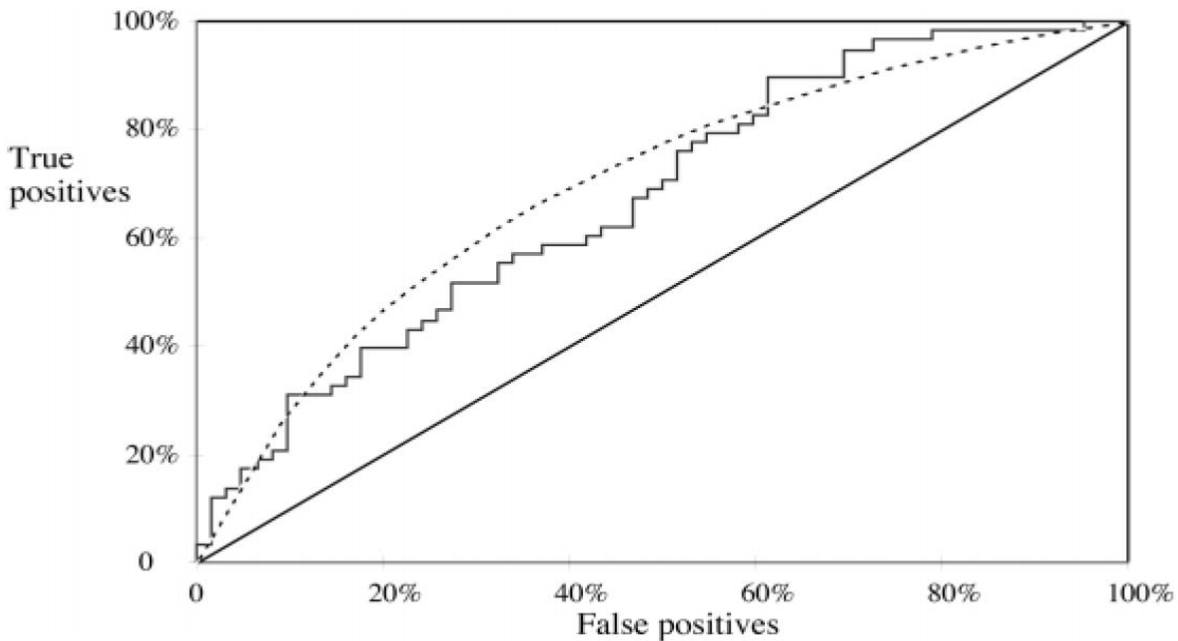


Figure 2: A ROC curve for the sample in Table 3. See text for details.

of predictions. The first two predictions are true positives (the actual class is *positive*) whereas the third and ninth predictions are false positives (the actual class is *negative*).

Taken together, we can develop an algorithm to generate a ROC curve for a two-class learning problem. The algorithm computes a set of $\langle x, y \rangle$ points on a two-dimensional graph, where the y -axis denotes the true-positive rate and the x -axis denotes the false-positive rate.

A ROC curve corresponding to the data in Table 3 is shown in Figure 2. The jagged line corresponds to the single data set in the table. The smooth curve is obtained by cross-validation, as follows. For each fold of the cross-validation, evaluate the classifier's predictions on the test set. For each different position on the y -axis, i.e. each possible number of false positives up to the total number of negatives in the test set, take just enough of the ranked set of test set examples to include that number of false positives and find the number of true positives. Taking the mean of these true positive counts over all folds gives the $\langle x, y \rangle$ points to draw the smooth curve.

4 Summary

ROC analysis is a very useful tool for assessing classifier performance where there are variable misclassification costs or highly skewed class distributions (e.g. test set examples are nearly all of one class, so it is easy to get an apparently high predictive accuracy). Both lift charts and ROC curves give a graphical view of classification performance and can be easily calculated where a ranking on predictions is available. The two-class confusion matrix or contingency table is a fundamental method which is the starting point for many statistical analysis techniques.

References

- [1] T. Fawcett. ROC Graphs: Notes and Practical Considerations for Data Mining Researchers. Tech Report HPL-2003-4, HP Labs, 2003. Available at <http://www.hpl.hp.com/personal/Tom.Fawcett/papers/>.
- [2] Ian H. Witten and Eibe Frank. *Data Mining*. Morgan-Kaufmann, San Francisco, CA, 2000.