Copyrighted Material



with the original proportion of *no* instances is used for testing, fewer errors will be made on these than on *yes* instances—that is, there will be fewer false positives than false negatives—because false positives have been weighted 10 times more heavily than false negatives. Varying the proportion of instances in the training set is a general technique for building cost-sensitive classifiers.

One way to vary the proportion of training instances is to duplicate instances in the dataset. However, many learning schemes allow instances to be weighted. (As we mentioned in Section 3.2, this is a common technique for handling missing values.) Instance weights are normally initialized to one. To build cost-sensitive trees the weights can be initialized to the relative cost of the two kinds of error, false positives and false negatives.

## Lift charts

In practice, costs are rarely known with any degree of accuracy, and people will want to ponder various scenarios. Imagine you're in the direct mailing business and are contemplating a mass mailout of a promotional offer to 1,000,000 households—most of whom won't respond, of course. Let us say that, based on previous experience, the proportion who normally respond is known to be 0.1% (1000 respondents). Suppose a data mining tool is available that, based on known information about the households, identifies a subset of 100,000 for which the response rate is 0.4% (400 respondents). It may well pay off to restrict the mailout to these 100,000 households—that depends on the mailing cost compared with the return gained for each response to the offer. In marketing terminology, the increase in response rate, a factor of four in this case, is known as the *lift* factor yielded by the learning tool. If you knew the costs, you could determine the payoff implied by a particular lift factor.

But you probably want to evaluate other possibilities, too. The same data mining scheme, with different parameter settings, may be able to identify 400,000 households for which the response rate will be 0.2% (800 respondents), corresponding to a lift factor of two. Again, whether this would be a more profitable target for the mailout can be calculated from the costs involved. It may be necessary to factor in the cost of creating and using the model—including collecting the information that is required to come up with the attribute values. After all, if developing the model is very expensive, a mass mailing may be more cost effective than a targeted one.

Given a learning method that outputs probabilities for the predicted class of each member of the set of test instances (as Naïve Bayes does), your job is to find subsets of test instances that have a high proportion of positive instances, higher than in the test set as a whole. To do this, the instances should be sorted in descending order of predicted probability of *yes*. Then, to find a sample of a given size with the greatest possible proportion of positive instances, just read

Copyrighted Material

Ian H. Witten, Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). (Morgan Kaufmann, 2005). Page 166.

Copyrighted Material



**Table 5.6 Data for a lift chart.**

| Rank | Predicted probability | Actual class | Rank | Predicted probability | Actual class |
|---|---|---|---|---|---|
| 1 | 0.95 | *yes* | 11 | 0.77 | *no* |
| 2 | 0.93 | *yes* | 12 | 0.76 | *yes* |
| 3 | 0.93 | *no* | 13 | 0.73 | *yes* |
| 4 | 0.88 | *yes* | 14 | 0.65 | *no* |
| 5 | 0.86 | *yes* | 15 | 0.63 | *yes* |
| 6 | 0.85 | *yes* | 16 | 0.58 | *no* |
| 7 | 0.82 | *yes* | 17 | 0.56 | *yes* |
| 8 | 0.80 | *yes* | 18 | 0.49 | *no* |
| 9 | 0.80 | *no* | 19 | 0.48 | *yes* |
| 10 | 0.79 | *yes* | ... | ... | ... |

the requisite number of instances off the list, starting at the top. If each test instance's class is known, you can calculate the lift factor by simply counting the number of positive instances that the sample includes, dividing by the sample size to obtain a success proportion and dividing by the success proportion for the complete test set to determine the lift factor.

Table 5.6 shows an example for a small dataset with 150 instances, of which 50 are *yes* responses—an overall success proportion of 33%. The instances have been sorted in descending probability order according to the predicted probability of a *yes* response. The first instance is the one that the learning scheme thinks is most likely to be positive, the second is the next most likely, and so on. The numeric values of the probabilities are unimportant: rank is the only thing that matters. With each rank is given the actual class of the instance. Thus the learning method was right about items 1 and 2—they are indeed positives—but wrong about item 3, which turned out to be a negative. Now, if you were seeking the most promising sample of size 10 but only knew the predicted probabilities and not the actual classes, your best bet would be the top ten ranking instances. Eight of these are positive, so the success proportion for this sample is 80%, corresponding to a lift factor of four.

If you knew the different costs involved, you could work them out for each sample size and choose the most profitable. But a graphical depiction of the various possibilities will often be far more revealing than presenting a single "optimal" decision. Repeating the preceding operation for different-sized samples allows you to plot a lift chart like that of Figure 5.1. The horizontal axis shows the sample size as a proportion of the total possible mailout. The vertical axis shows the number of responses obtained. The lower left and upper right points correspond to no mailout at all, with a response of 0, and a full mailout, with a response of 1000. The diagonal line gives the expected result for different-

Copyrighted Material

Ian H. Witten, Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). (Morgan Kaufmann, 2005). Page 167.

Copyrighted Material



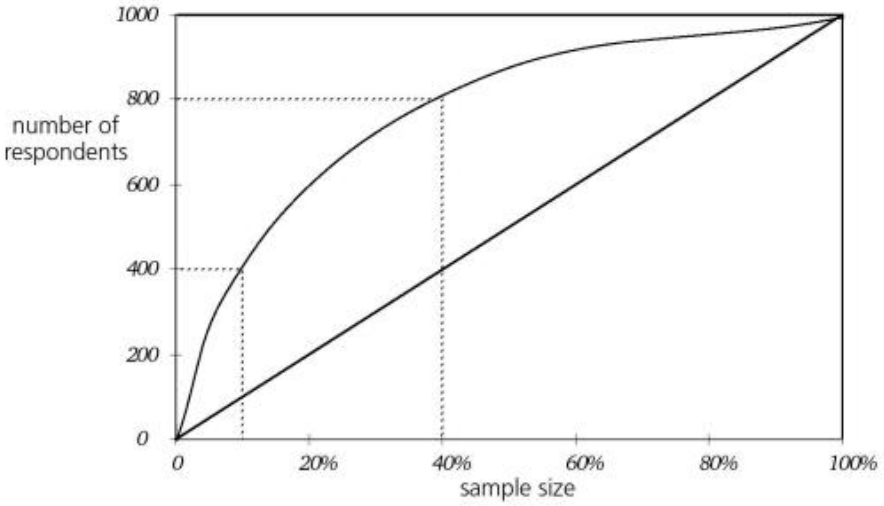


**Figure 5.1** A hypothetical lift chart.

sized random samples. But we do not choose random samples; we choose those instances which, according to the data mining tool, are most likely to generate a positive response. These correspond to the upper line, which is derived by summing the actual responses over the corresponding percentage of the instance list sorted in probability order. The two particular scenarios described previously are marked: a 10% mailout that yields 400 respondents and a 40% one that yields 800.

Where you'd like to be in a lift chart is near the upper left-hand corner: at the very best, 1000 responses from a mailout of just 1000, where you send only to those households that will respond and are rewarded with a 100% success rate. Any selection procedure worthy of the name will keep you above the diagonal—otherwise, you'd be seeing a response that was worse than for random sampling. So the operating part of the diagram is the upper triangle, and the farther to the northwest the better.

## ROC curves

Lift charts are a valuable tool, widely used in marketing. They are closely related to a graphical technique for evaluating data mining schemes known as *ROC curves,* which are used in just the same situation as the preceding one, in which the learner is trying to select samples of test instances that have a high proportion of positives. The acronym stands for *receiver operating characteristic,* a term used in signal detection to characterize the tradeoff between hit rate and false alarm rate over a noisy channel. ROC curves depict the performance of a classifier without regard to class distribution or error costs. They plot the number

Copyrighted Material

Ian H. Witten, Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). (Morgan Kaufmann, 2005). Page 168.