Data Mining for Direct Marketing: Problems and Solutions

Charles X. Ling and **Chenghui Li**
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada N6A 5B7
Tel: 519-661-3341; Fax: 519-661-3515
E-mail: ling,cli@csd.uwo.ca

Abstract

Direct marketing is a process of identifying likely buyers of certain products and promoting the products accordingly. It is increasingly used by banks, insurance companies, and the retail industry. Data mining can provide an effective tool for direct marketing. During data mining, several specific problems arise. For example, the class distribution is extremely imbalanced (the response rate is about 1%), the predictive accuracy is no longer suitable for evaluating learning methods, and the number of examples can be too large. In this paper, we discuss methods of coping with these problems based on our experience on direct-marketing projects using data mining.

1 Introduction

Almost all industries that sell products and services need to advertise and promote their products and services. Banks, insurance companies, and retail stores are typical examples. There are generally two approaches to advertisement and promotion: mass marketing and direct marketing. Mass marketing, which uses mass media such as television, radio, and newspapers, broadcasts messages to the public without discrimination. It used to be an effective way of promotion when the products were in great demand by the public. For example, electronic goods (TVs, fridges) after World War II were in great demand, so TV commercials were effective in promoting those products (Hughes, 1996). However, in today's world where products are overwhelming and the market is highly competitive, mass marketing has become less effective. The response rate, the percent of people who actually buy the products after seeing the promotion, is often low. From the datasets that we have studied (see later), 1% is quite typical.

The second approach of promotion is direct marketing. Instead of promoting to customers indiscriminatively, direct marketing studies customers' characteristics and needs, and selects certain customers as the target for promotion. The hope is that the response rate for the selected customers can be much improved.

Nowadays, a huge amount of information on customers is kept in databases. Thus, data mining can be very effective for direct marketing. Regularities and patterns for buyers can be discovered from the database to predict and select worthy customers for promotion. Data mining, an integration of machine learning, computer visualization, and statistics, has been used widely in direct marketing to target customers (Agrawal, Ghosh, Imielinski, Iyer, & Swami, 1992; Terano & Ishino, 1996; Ciesielski & Palstra, 1996).

2 Process of Data Mining for Direct Marketing

According to Fayyad, Piatetsky-Shapiro, Smyth, and Uthurusamy (1996), data mining is a non-trivial process of discovering novel, implicit, useful, and comprehensive knowledge from a large amount of data. In direct marketing, this knowledge is a description of likely buyers or responders, and is useful in obtaining higher profit than mass marketing.

As an example, banks and insurance companies may have a large database of customers to whom they want to sell certain products (such as loan products, retirement packages, and life insurance policies). There can be two situations. In the first situation, some (say $X\%$) of the customers in the database have already bought the product, through previous mass marketing or passive promotion. X is usually rather small, typically around 1. Data mining can be used to discover patterns of buyers, in order to single out likely buyers from the current non-buyers, $(100-X)\%$ of all customers. More specifically, data mining for direct marketing in the first situation can be described in the following steps:

1. Get the database of all customers, among which $X\%$ are buyers.
2. Data mining on the dataset
   - Overlaying: append geo-demographic information to the database.
   - Data pre-processing: transform address and area codes, deal with missing values, etc.
   - Split the database into training set and testing set.
   - Apply learning algorithms to the training set.

3. Evaluate the patterns found on the testing set. It may be necessary to iterate to previous steps if the results are not satisfactory.
4. Use the patterns found to predict likely buyers among the current non-buyers.
5. Promote to the likely buyers (called *rollout*).

In the second situation, a brand new product is to be promoted to the customers in the database, so none of the customers are buyers. In this case, a pilot study is conducted, in which a small portion (say 5%) of the customers is chosen randomly as the target of promotion. Again, X% of the customers in the pilot group may respond to the promotion. Then data mining is performed in the pilot group to find likely buyers in the whole database.

Clearly, the two situations are quite similar from the data mining point of view, except that the dataset for mining in the second case is smaller. The datasets to be discussed in this paper belong to the first situation. They are particularly suitable for data mining since the sizes of the datasets are usually quite large.

Let us first take a quick look at the benefit of one form of direct marketing, direct mail campaign, compared to mass mail campaign. See Table 1 for details. Assume that the whole database contains 600,000 customers. In direct mailing, only 20% of the customers identified as likely buyers by data mining (which costs $40,000) are chosen to receive the promotion package in the mail. The mailing cost is thus reduced dramatically. At the same time, however, the response rate can be improved from 1% in mass mailing to 3% (a realistic improvement for a 20% rollout). The net profit from the promotion becomes positive in direct mailing, compared to a loss in mass mailing. See Table 1 for specific figures in this realistic example.

3 Datasets for Direct Marketing

We have three datasets used for direct marketing from three different sources. All three belong to the first situation discussed in the previous section: small percents of the customers in the datasets are already buyers or responders, and we need to discover likely buyers from the current non-buyers. Due to confidentiality of the datasets, we can only give very a brief description.

The first dataset is for a loan product promotion from a major bank in Canada. It has about 90,000 customers in total, and only 1.2% are responders. Each customer is described by 55 attributes, about half are discrete and the other half numerical. After transforming some attributes (such as the area code), a total of 62 attributes are used for the data mining purpose.

The second dataset is from a major life insurance company, for an RRSP (Registered Retirement Saving Plan) campaign. It has 80,000 customers in total, with an initial 7% buyers. Each customer is described by 10 numerical and discrete attributes.

The third dataset is from a company which runs a particular "bonus program". The company has over 100 sponsors (retail stores, gas stations, etc.), and when customers buy certain products from the partners, a certain amount of bonus is accumulated, which can be redeemed for other services and products (from free flights to free movies). Naturally, the company is gathering a huge amount of information on what customers are buying every day. The company then mails commercial flyers to customers along with quarterly summary statements, and it wants to send the right set of commercial flyers to customers to improve the response rate. The dataset we have contains about 104,000 customers, and only 1% are responders. Each customer is described by 299 attributes, most of which are numerical.

4 Specific Problems in Data Mining

During data mining on these three datasets for direct marketing, we encountered several specific problems.

The first and most obvious problem is the extremely imbalanced class distribution. Typically, only 1% of the examples are positive (responders or buyers), and the rest are negative. Most learning algorithms do not behave well on this kind of dataset: they simply discover one rule or pattern saying that all examples are negative. This rule can reach 99% accuracy in predicting training examples, as well as testing examples (assuming that they have a similar class distribution as training examples). Many data mining and machine learning researchers have recognized and studied this problem in recent years (Fawcett & Provost, 1996; Kubat, Holte, & Matwin, 1997; Kubat & Matwin, 1997; Lewis & Catlett, 1997; Pazzani, Merz, Murphy, Ali, Hume, & Brunk, 1997).

The second problem is that even if sensible patterns can be found, the predictive accuracy cannot be used as a suitable evaluation criterion for the data mining process. One reason is that classification errors must be dealt with differently: having false positive errors (recognizing buyers among non-buyers) is actually the goal, while false negative errors (recognizing non-buyers among existing buyers?) are not desired. Another reason is that the predictive accuracy is too weak for the purpose of customer targeting. Binary classification can only predict two classes, buyers and non-buyers, and it does not make a finer distinction among the predicted buyers or non-buyers. This does not provide enough flexibility in choosing a certain percent of likely buyers for promotion (e.g., we may want to promote to 30% of customers, but the programs only predict 6% as likely buyers). In addition, it does not provide an opportunity for using different *means* of promotion on the chosen customers. For example, we may want to make personal calls to the first 100 most likely buyers, and send personal mail to the next 1,000 likely buyers, etc.

The third problem is that when we split the whole dataset into training and testing sets with equal sizes, the training set with a large number of variables can

Table 1: A comparison between direct mail campaign and mass mail campaign.

	Mass mailing	Direct mailing
Number of customers mailed	600,000	(20%) 120,000
Cost of printing, mailing ($0.71 each)	$426,000	$85,200
Cost of data mining	$0	$40,000
Total promotion cost	$426,000	$125,200
Response rate	1.0%	3.0%
Number of sales	6,000	3,600
Profit from sale ($70 each)	$420,000	$252,000
Net profit from promotion	−$6,000	$126,800

be too large for certain learning algorithms. We must choose efficient learning algorithms for these datasets.

5 Solutions

To solve the second problem (of inadequacy of binary classification algorithms), we require learning algorithms not only to classify, but also to classify with a confidence measurement, such as a probability estimation or certainty factor. This allows us to *rank* training and testing examples. The ranking of non-buyers (from most likely to least likely buyers) makes it possible to choose any number of likely buyers for the promotion. It also provides a fine distinction among chosen customers to apply different means of promotion. Therefore, we need learning algorithms that can also produce probability estimation or certainty factors. We will discuss such algorithms that we used in Section 5.1.

Since we will use learning algorithms that can produce probability and thus can rank examples, we should use the *lift* instead of the predictive accuracy as the evaluation criterion. Lift analysis has been widely used in database marketing previously (Hughes, 1996). A lift reflects the redistribution of responders in the testing set after the testing examples are ranked. Clearly, if some regularities are found, the ranking of testing examples from most likely to least likely buyers would result in a distribution in which most existing buyers appear in the upper part of the ranked list. Section 5.2 discusses the lift analysis in more detail.

Presumably, since we apply learning algorithms that rank testing examples and we use lift as the evaluation criterion, we do not need to care much about the imbalanced class distribution in the training set — the first problem mentioned in Section 4 — as long as the learning algorithms produce suitable ranking of the testing examples even if all of them are predicted as negative. Section 5.3 shows that a certain class distribution of training examples produces the best lift, compared to other distributions. As a bonus, it also dramatically reduces the size of the training set, which is the third problem mentioned in Section 4.

5.1 Data Mining Algorithms

As discussed earlier, we need to use learning algorithms that also produce probability in order to rank the testing examples. Several types of learning algorithms satisfy this condition: Naive Bayes algorithm, nearest neighbour algorithm, and neural networks. Due to efficiency considerations, we choose Naive Bayes algorithm, which makes a conditional independent assumption: given the class label, the attribute values of any example are independent. This assumption is almost certainly violated, but Domingos and Pazzani (1996) showed that it may not affect the classification decision. Details of the Naive Bayes can be found in (Langley, Iba, & Thompson, 1992), and we will not repeat it here.

Decision tree learning algorithms such as C4.5 (Quinlan, 1993) normally only classify examples. We modify C4.5 slightly so that it also produces a certainty factor (CF) for its classification. Basically, the classification of each leaf is determined by the majority class in that leaf. The certainty factor of examples in a leaf is the ratio of the number of examples of the majority class over the total number of examples in that leaf. This method was used in the program `consult` that comes with the C4.5 package (Quinlan, 1993).

We do not apply Naive Bayes and C4.5 with CF directly on the datasets we have. Recent results show that using an ensemble of classifiers often improves the predictive accuracy by reducing the variation error of unstable classifiers (Domingos & Pazzani, 1996; Quinlan, 1996). Bagging (Breiman, 1996) and ada-boosting (Freund & Schapire, 1996) are two common methods of ensembling classifiers. We use ada-boost applied to the Naive Bayes (Elkan, 1997) and to C4.5 with CF as our learning algorithms.[1] Basically, ada-boost maintains a sampling probability distribution on the training set, and modifies the probability distribution after each classifier is built. The probability of examples with an incorrect prediction by the previous classifier is increased, so these examples will be sampled more heavily in the next round of boosting, to be learnt correctly.

However, the original ada-boost (Freund & Schapire, 1996) was designed to work with binary classifiers (without CF or probability). Since classifiers required for direct marketing also produce probability, we can

[1]The algorithms used in this paper are for non-commercial purpose.

make use of the probability in ada-boost. Therefore, ada-boost was modified at several places to work better with classifiers with CF. For example, the amount of update in probability of training examples is calculated from the difference in probability of the prediction, giving a more graded effect than just correct or incorrect predictions (Elkan, 1997).

The original Naive Bayes only produces a classification after weighing predictions from all classifiers. We modify this to produce probability of the prediction by combining probabilities of predictions from all classifiers.

5.2 Lift Index for Evaluation

As we discussed in the beginning of Section 5, we should use the lift instead of the predictive accuracy as the evaluation criterion. After the learning algorithm ranks all testing examples from most likely responders to least likely responders, we divide the ranked list into 10 equal deciles (could be more partitions), and see how the original responders distribute in the 10 deciles. If regularities are found, we will see more responders in the top deciles than bottom deciles (Hughes, 1996). Table 2 is an example of the lift table. In this example, the testing set has a total of 100,000 examples (thus each decile has 10,000), with a total of 1,000 positive examples. Clearly, this lift table shows a distribution in which more responders are gathered in the top deciles than bottom deciles.

Given such a lift table as in Table 2, how can we evaluate a learning algorithm? In KDD-97-Cup Competition, two measurements are used. One is the number of responders in the top decile (top 10%); in the above example, 410. This reflects the ability to locate a small number of highly likely buyers. The other is the total number of responders in the top 4 deciles (top 40%); in the above example, it is $410+190+130+76 = 806$. This reflects the ability to locate a large number of likely responders in a larger rollout. These two measurements are used separately in evaluating data mining tools in the KDD-97-Cup Competition.

We feel that this may not be a suitable way of measuring learning methods. First of all, the two measurements may not be correlated. More importantly, when using the number of responders in the top X deciles as the measurement, the distribution within the top X deciles and within the rest of the deciles can be skewed. For example, if the distribution of the top 4 deciles is 0, 0, 0, and 806, it will give the same score of 806, but that is clearly undesirable. It would be best to use just one number to measure the performance of learning algorithms based on the whole lift table.

We decide to use a weighted sum of the items in the lift table. Assuming the 10 deciles in the lift table (from top) are S_1, S_2, ..., S_{10}, then we define the *lift index* as

$$S_{lift} = (1 \times S_1 + 0.9 \times S_2 + ... + 0.1 \times S_1)/\sum_i S_i$$

In the above example, the lift index is $S_{lift} = (1 \times 410 + 0.9 \times 190 + ... + 0.1 \times 26)/1000 = 81.1\%$.

It turns out that such a lift index S_{lift} converges to

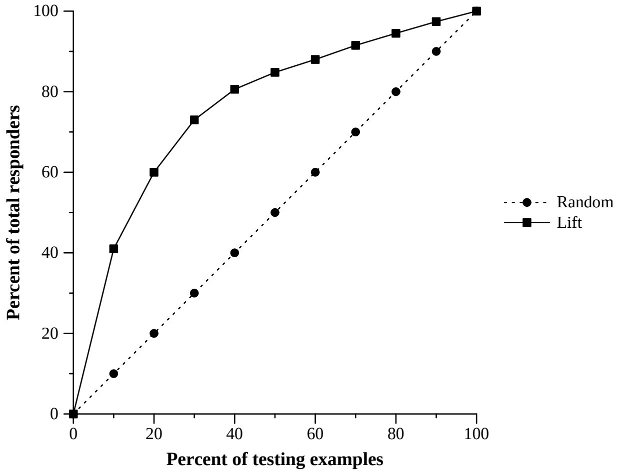

Figure 1: A cumulative lift curve.

the ratio of the area under the *cumulative* lift curve (AUC), which is illustrated in Figure 1. If the distribution of the buyers in the 10 deciles is random (no regularity is found), then the cumulative curve would converge to the diagonal line, and S_{lift} would be 55% (but converges to 50% with more partitions). In the best situation when $S_1 = \sum_i S_i < 10\%$, $S_{lift} = 100\%$. In the worse case when $S_{10} = \sum_i S_i$ (and rest of $S_i = 0$), $S_{lift} = 10\%$ (but converges to 0%). That is, the lift index has a nice property that it is independent to the number of the responders. It will be 50% for the random distribution, above 50% for better than random, and below 50% for worse than random distributions.

The cumulative lift curve as in Figure 1 has a close relation with the ROC (receiver operating characteristic) curves, which have been widely used previously (see, for example, Hanley & McNeil, 1982). To obtain an ROC curve, positive and negative cases are mixed randomly first, and then presented to a decision maker who rates each case, normally ranging from definitely negative (0) to definitely positive (1). The ROC curve is then obtained by considering broader and broader ranges for positive cases (i.e., greater than 0.95 as positive, greater than 0.9 as positive, greater than 0.8 as positive, etc.): the x axis is the rate of false positive (over all negative cases), while the y axis the rate of true positive (over all positive cases). The ROC curve looks similar to the cumulative lift curve, except the former lies slightly above the latter (except at (100, 100)). For any point (p, q) where $p < q$ on the ROC curve, the corresponding point on the lift curve would be $(q \times X\% + p \times (1 - X\%), q)$, where X is the percent of positive cases in the dataset. Thus, the lift curve is slightly under the ROC curve when $p < q$. Therefore,

Table 2: A typical lift table.

10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
410	190	130	76	42	32	35	30	29	26

the lift index we proposed is also closely related to the area under the curve (AUC) index of the ROC curves.

However, the cumulative lift curves and the lift index have better intuitive meanings for direct marketing. Thus, we use the lift index as the sole criterion in evaluating and comparing learning methods in the rest of the paper.

5.3 Mixture of Training Set

Now we have two learning algorithms (ada-boosted Naive Bayes and ada-boosted C4.5 with CF) that also produce probability. We split randomly all three datasets described in Section 3 into training and testing sets with equal sizes, and use the training set for learning. The learned results are used to rank the testing examples. We then calculate the lift index on the ranked testing examples to evaluate and compare learning algorithms. We repeat this 10 times for each dataset to obtain an average lift index.

Since we now rank testing examples, the imbalanced class distribution in the training set (the first problem described in Section 4) may no longer be a problem, as long as the ranking is preserved. To test this possibility, we keep all positive examples in the training set, but add different numbers of negative examples to form new training sets. Similar methods of reducing negative examples have been studied previously (Kubat & Matwin, 1997; Lewis & Catlett, 1997). The results of applying boosted Naive Bayes are shown in Table 3. Results with boosted decision trees are similar and thus omitted here.

It is interesting to see that for all three datasets, the best lift index is obtained when the positive and negative examples are about equal in number (the difference with more negative examples is not statistically significant). If only 1% of training examples are positive (responders), we only need to use another 1% of the negative examples (non-responders) to form a new training set. This naturally reduces the size of the training set dramatically.

A more careful look at the probability distribution of testing examples reveals why this is the case. Naive Bayes (and C4.5 with CF) does not produce true probabilities. However, with equal class distribution in training set, the probabilities of testing examples are evenly distributed between 0 and 1, thus reducing mistakes in ranking. If the class distribution is unequal, the probability distribution may be skewed within a narrow range in one end. Thus, errors in probability estimation affects the ranking more easily, causing a lower lift index.

While the best lift index is obtained when the ratio of positive and negative examples is about 1/1, the error rates (not shown here) drop dramatically from about 40% with 1/1 ratio to 3% with 1/8 ratio in one dataset. This verifies that the error rate is not suitable for obtaining the best lift in the extremely imbalanced datasets.

However, it seems undesirable to use only 1% of the negative examples, throwing away information in the rest of the negative examples. One possibility is to oversample with replacement the existing positive examples to a few times (2x, 5x, 10x, 20x) of the original size, while keeping the same numbers of negative examples.[2] This way, we naturally increase the information on negative examples in the training set while still keeping numbers of positive and negative examples the same. Table 4 lists the average lift index with boosted Naive Bayes on the three datasets.

It is surprising to see that oversampling does not provide any significant improvement using the Naive Bayes algorithm. This is because Naive Bayes is a global algorithm, unlike local algorithms such as C4.5, which can carve the instance space locally to finer partitions with more training examples. It has been observed previously that Naive Bayes does not scale up well (Kohavi, 1996). In addition, one must also remember that oversampling positive examples does not really increase information.

It might be expected that the local algorithm C4.5 (with CF) may perform better with larger training sets. It does, but only slightly (statistically significant). Table 5 illustrates the average lift index using C4.5 with CF on the three datasets. Clearly, the best results from C4.5 are similar to the ones with Naive Bayes without oversampling. This makes both Naive Bayes and C4.5 attractive algorithms that are also efficient to apply to large datasets.

6 Discussions and Summary

We must not lose the big picture of doing data mining, which is to improve the ROI (return of investment) or net profit. In the example of the direct mail campaign using data mining as illustrated in Table 1, if we obtain a lift curve as in Figure 1, we can plot a curve of net profit as in Figure 2. As we can clearly see, if we only mail to some top small percent of customers, even though the relative lift is higher, the total number of responders would be too small. If we mail to too many customers, the gain in response will not be justifiable for the printing and mailing cost. Therefore, there is

[2]Since there are enough negative examples, we sample them without replacement. This approach was used in (Solberg & Solberg, 1996).

Table 3: Average lift index on three datasets with different numbers of negative examples using boosted Naive Bayes.

Positive/negative	Bank	Life Insurance	Bonus Program
1/0.25	66.4%	72.3%	78.7%
1/0.5	68.8%	74.3%	80.3%
1/1	70.5%	75.2%	81.3%
1/2	70.4%	75.3%	81.2%
1/4	69.4%	75.4%	81.0%
1/8	69.1%	75.4%	80.4%

Table 4: Average lift index on three datasets with different numbers of oversampled positive examples and unique negative examples using boosted Naive Bayes.

Positive/negative	Bank	Life Insurance	Bonus Program
1x/1x	70.5%	75.2%	81.3%
2x/2x	70.5%	75.3%	81.4%
5x/5x	70.6%	75.2%	81.6%
10x/10x	70.6%	75.2%	81.8%
20x/20x	70.6%	NA	81.8%

an optimal percent that can bring us the maximum net profit. Of course, such optimal cut-off points depend critically on many factors in the whole process of direct marketing using data mining: cost of mailing, cost of data mining, profit per sale, lift curve, etc. Some of these factors depend on each other, making the strategic decision on data mining non-trivial. For example, more data mining effort (also more cost) may improve the lift index, but too much may not be worthy of doing.

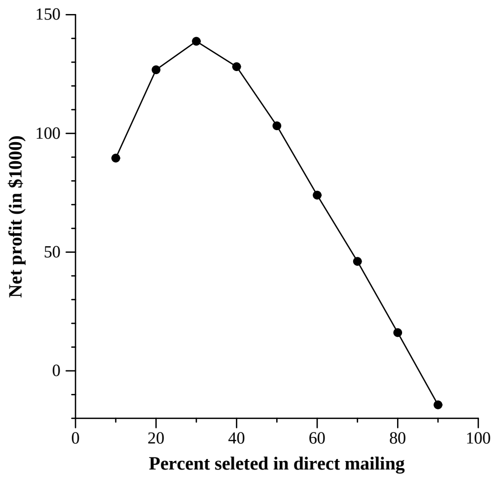

Figure 2: Net profit in direct mailing.

To summarize, we have used Naive Bayes and C4.5 with CF on three datasets for direct marketing from different sources. Several common problems emerged, and we have provided solutions to these problems. We demonstrated that data mining is an effective tool for direct marketing, which can bring more profit to banks, insurance companies, and the retail industry than the traditional means of mass marketing.

Acknowledgements

We thank gratefully Charles Elkan for providing us with his codes of ada-boosted Naive Bayes, on which our implementation is based, and Ross Quinlan for his C4.5 codes. These codes were used for non-commercial purpose in the paper. We also thank reviewers who provided useful comments.

Reference

Agrawal, R., Ghosh, A., Imielinski, T., Iyer, B., & Swami, A. (1992). An interval classifier for database mining applications. In *Proceedings of the 18th Conference on Very Large Databases, Morgan Kaufman pubs. (Los Altos CA), Vancouver.*

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Ciesielski, V., & Palstra, G. (1996). Using a hybrid neural/expert system for data base mining in market survey data. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 38–43.

Domingos, P., & Pazzani, M. (1996). Beyond independence: conditions for the optimality of the simple Bayesian classifier. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 105 – 112.

Elkan, C. (1997). Boosting and naive Bayesian learning. Tech. rep. CS97-557, Department of Com-

Table 5: Average lift index on three datasets with different numbers of oversampled positive examples and unique negative examples using boosted C4.5 with CF.

Positive/negative	Bank	Life Insurance	Bonus Program
1x/1x	65.6%	74.3%	74.3%
2x/2x	66.2%	75.7%	75.4%
5x/5x	67.9%	76.1%	77.5%
10x/10x	68.2%	76.2%	78.3%
20x/20x	68.5%	76.2%	78.9%

puter Science and Engineering, University of California.

Fawcett, T., & Provost, F. (1996). Combining data mining and machine learning for effective user profile. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 8–13.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Eds.). (1996). *Advances in Knowledge Discovery and Data Mining.* MII Press, Mento Park.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 148 – 156.

Hanley, J., & McNeil, B. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143*, 29–36.

Hughes, A. M. (1996). *The complete database marketer : second-generation strategies and techniques for tapping the power of your customer database.* Chicago, Ill. : Irwin Professional.

Kohavi, R. (1996). Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 202–207.

Kubat, M., Holte, R., & Matwin, S. (1997). Learning when negative examples abound. In *Proceedings of the 9th European Conference on Machine Learning.*

Kubat, M., & Matwin, S. (1997). Addressing the curse of imbalanced training sets. In *Proceedings of the 14th International Conference on Machine Learning.*

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 223 – 228.

Lewis, D., & Catlett, J. (1997). Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the 11th International Conference on Machine Learning*, pp. 148–156.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., & Brunk, C. (1997). Reducing misclassification costs. In *Proceedings of the 11th International Conference on Machine Learning*, pp. 217–225.

Quinlan, J. R. (1996). Bagging, boosting, and c4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 725 – 730.

Quinlan, J. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann: San Mateo, CA.

Solberg, A. H. S., & Solberg, R. (1996). A large-scale evaluation of features for automatic detection of oil spills in ERS SAR images. In *IEEE Symp. Geosc. Rem. Sens (IGARSS)*, pp. 1484–1486.

Terano, T., & Ishino, Y. (1996). Interactive knowledge discovery from marketing questionnaire using simulated breeding and inductive learning methods. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 279–282.