Mining Customer Value: From Association Rules to Direct Marketing [*]

Ke Wang
Simon Fraser University
wangk@cs.sfu.ca

Senqiang Zhou
Simon Fraser University
szhoua@cs.sfu.ca

Qiang Yang
Hong Kong University of Science and Technology
qyang@cs.ust.hk

Jack Man Shun Yeung
Simon Fraser University
yeung@cs.sfu.ca

Abstract

Direct marketing is a modern business activity with an aim to maximize the profit generated from marketing to a selected group of customers. A key to direct marketing is to select a right subset of customers so as to maximize the profit return while minimizing the cost. Achieving this goal is difficult due to the extremely imbalanced data and the inverse correlation between the probability that a customer responds and the dollar amount generated by a response. We present a solution to this problem based on a creative use of association rules. Association rule mining searches for all *rules above an interestingness threshold, as opposed to* some *rules in a heuristic-based search. Promising association rules are then selected based on the observed value of the customers they summarize. Selected association rules are used to build a model for predicting the value of a future customer. On the challenging KDD-CUP-98 dataset, this approach generates 41% more profit than the KDD-CUP winner and 35% more profit than the best result published thereafter, with 57.7% recall on responders and 78.0% recall on non-responders. The average profit per mail is 3.3 times that of the KDD-CUP winner.*

[*] Research was supported in part by research grants from the Natural Science and Engineering Research Council of Canada

Keywords: association rule, classification, direct marketing, regression

1 Introduction

Direct marketing makes it possible to offer goods or services or transmit messages to a specific, targeted segment of the population by mail, telephone, email or other direct means [1]. Direct marketing is one of the most effective and measurable methods of marketing available. For example, retail industries need to identify buyers of certain products; banks and insurance companies need to promote loan insurance products to customers; and fundraising organizations need to identify potential donors. Direct marketing campaigns are only as successful as the mailing list used. A good mailing list will target only the consumers that are potential or valuable customers. Typically, direct marketing models select addresses by predicting future response behavior. In management and marketing science, Stochastic models are used to describe the response behavior of customers, including binary choice models (Bult and Wansbeek, 1995), latent class models (Desarbo and Ramaswamy, 1994), neural networks (Levin and Zahavi, 1996, Potharst et al., 2002) and Markov chains (Bitran and Mondschein, 1996).

In this paper, we propose a data mining method for determining the mailing list. Available is a *historical* database containing information about previous mailing campaigns, including whether a customer responded and the dollar amount collected if responded. The task is to build a model to predict *current* customers who are likely to respond. The goal is to maximize the sum of net profit, $\Sigma(dollar\ amount - mailing\ cost)$, over the contacted customers. We choose the KDD-CUP-98 dataset (KDD98, 1998a) as the case study. This dataset was collected from the result of the 1997 Paralyzed Veterans of America fundraising mailing campaign (more details in Section 2) and only 5% of records are responders. Thus, simply classifying all customers into non-responders would give 95% accuracy, but this does not generate profit.

A principled method is ranking customers by the estimated probability to respond and selecting some top portion of the ranked list (Ling and Li, 1998, Masand and Shapiro, 1996). For example, if the top 5% of the ranked list contains 30% of all responders, the lift model gives the lift of $30/5 = 6$ (Ling and Li, 1998, Masand and Shapiro, 1996). A significant drawback of this approach is that the actual customer value, e.g., the donation amount in the example of fundraising, is ignored in the ranking, or it requires a uniform customer value for all customers. As pointed out in (KDD98, 1998b) for the KDD-CUP-98 task, there is an inverse correlation between the likelihood to buy (or donate) and the dollar amount to spend (or donate).

[1] http://www.commerce-database.com/directmarketing.htm

This inverse correlation reflects the general trend that the more dollar amount is involved, the more cautious the buyer (or donor) is in making a purchase (or donation) decision. As a result, a probability based ranking tends to rank down, rather than rank up, valuable customers.

The realization that a cost-sensitive treatment is required in applications like direct marketing has led to a substantial amount of research. (Domingos, 1999) proposed the MetaCost framework for adapting accuracy-based classification to cost-sensitive learning by incorporating a *cost matrix* $C(i, j)$ for misclassifying true class j into class i. (Zadrozny and Elkan, 2001) examined the more general case where the *benefit* $B(i, j, x)$ depends not only on the classes involved but also on the individual customers x. For a given customer x, the "optimal prediction" is the class i that leads to the minimum expected cost

$$\Sigma_j P(j|x)C(i, j)$$

or the maximum expected benefit

$$\Sigma_j P(j|x)B(i, j, x).$$

Both methods require to estimate the conditional class probability $P(j|x)$. In this phase, since only the frequency information about x, not the customer value of x, is examined, valuable customers, who tend to be infrequent because of the "inverse correlation", are likely to be ignored. The customer value is factored only at the end via the factor $B(i, j, x)$.

In this paper, we propose a novel approach to address the above issues. First, we exploit association rules (Agrawal et al., 1993, Agrawal and Srikant, 1994) of the form $X \rightarrow respond$ to extract features for responders, where X is a set of items that is correlated with the "respond" class. Unlike traditional rule induction (Michalski, 1969, Quinlan, 1983, Clark and Niblett, 1989) that examines *one variable at a time*, association rules evaluate *a combination of variables (i.e., X) at a time*, therefore, better represent correlated features. We select a small subset of association rules to identify potential customers in the current campaign. We address two key issues, namely, push the customer value in selecting association rules, and maximize profitability over the customers (instead of historical ones). On the challenging KDD-CUP-98 task, which has 5% responders and 95% non-responders, this method generated 41% more profit than the winner of the competition and 35% more profit than the best known result after the competition, and the average profit per mail is 3.3 times that of the winner. This method identifies correctly 57.7% of responders and 78% of non-responders, thus, also provides a competitive solution to the cost-sensitive classification.

The motivation of association rules in the market basket analysis has led to several attempts to extend and apply such rules in business environments. (Savasere et al., 1998) considers negative association rules that tell what items a customer will not likely buy given that he/she buys a certain set of other items. (Tan et al., 2000) considers indirect association rules where the association of two items is conditioned on the presence of some set of other items. Such associations are purely count or occurrence based and have no direct relationships with the “inverse correlation” considered here that addresses profit. We focus on *using* association rules based on customer value, whereas these works focus on *finding* association rules based on count information. This distinction is substantial because association rules themselves do not tell how to maximize an objective function, especially in the presence of the “inverse correlation”. Our work differs from the product recommendation in (Wang et al., 2002) and item selection in (Brijs et al., 1999, Wang and Su, 2002) in that we identify valuable customers instead of items or products.

The rest of this paper is organized as follows. In Section 2, we examine the challenges in the KDD-CUP-98 dataset and outline our approach. In Section 3, we present the detailed algorithm. We evaluate our method using the KDD-CUP-98 task in Section 4. Finally, we conclude the paper.

2 Challenges and Our Proposals

The KDD-CUP-98 dataset (KDD98, 1998a) contains 191,779 records about individuals contacted in the 1997 mailing campaign. Each record is described by 479 non-target variables and two target variables indicating the “respond”/“not_respond” classes and the actual donation in dollars. About 5% of records are “respond” records and the rest are “not_respond” records. The dataset has been pre-split into 50% for learning and 50% for validation. The validation set is reserved for evaluation and is held out from the learning phase. The competition task is to build a prediction model of the donation amount using the learning set. The participants are contested on $\Sigma(actual\ donation - \$0.68)$ over the validation records with predicted donation greater than the mailing cost $0.68.

We chose this fundraising task because it shares several key requirements with direct marketing. Both activities are only as successful as the mailing list used, and require identifying a subset of ”valuable” individuals to maximize some objective function (e.g., sales, customer services, donation amount). The fundraising dataset contains offerings and responses in previous campaigns, similar to those kept in a typical direct marketing campaign. The target variable “actual donation” corresponds to the sales value on a contacted customer, and $0.68 corresponds to the cost associated with contacting a customer. In fact, this problem is more general than the direct marketing considered in (Ling and Li, 1998) in that it allows to model a different profitability for different customers, just as a different sale could yield a different sales profit due to the difference in products, quantity and promotion. However, this generalization raises some new issues as explained below.

2.1 The challenges

This real life dataset presents two challenges.

Challenge 1. Quoted from (KDD98, 1998b), “there is often an inverse correlation between the likelihood to respond and the dollar amount of the gift”. This inverse correlation could exist in the offerings to the same customer or different customers. For the same customer, a standard handling is avoiding multiple offerings with a certain time period. For different customers, it means that there are many “small customers” making small purchases and few “big customers” making big purchases. We focus on this type of inverse correlation. A pure probability based ranking tends to favor “small customers” because of higher likelihood to respond, and ignore “big customers”. Previous researches addressed this issue in two steps: obtain the probability estimation from a standard classification model such as decision tree (Ling and Li, 1998, Masand and Shapiro, 1996), bagging (Domingos, 1999) and smoothing (Zadrozny and Elkan, 2001), and re-rank the probability based ranking by taking into account the customer value (Masand and Shapiro, 1996, Zadrozny and Elkan, 2001). The disadvantage of this approach is that the customer value is ignored in the first step.

Challenge 2. The high dimensionality and the scare target population present a significant challenge for extracting the features of the “respond” class. The dataset is very high in dimensionality, i.e., 481 variables, and very scare in the “respond” class population, only 5% of the dataset. Since any subset of variables can be a feature for distinguishing the “respond” class from “not_respond” class, searching for such features is similar to searching for a needle from a haystack. The “one attribute at a time” gain criterion (Quinlan, 1993) does not search for correlated variables as features. Though, the independence assumption of the Naive Bayesian classifier is quite robust to classification, which only depends on the maximum class probability (Domingos and Pazzani, 1996), it suffers from distortion if used for probability estimation where non-maximum class probabilities are also used for ranking customers. Our study on the KDD-CUP-98 dataset shows that taking into account this correlation yields a significantly higher profit.

2.2 The proposed approach

We address these challenges in two steps.

In the first step, we propose the notion of *focused association rules* to focus on the features that are typical of the “respond” class and not typical of the “not_respond” class. A focused association rule has the form $X \rightarrow respond$, where X is a set of items for non-target variables, such that X occurs *frequently* in the “respond” class and each item in X occurs *infrequently* in the “not_respond” class. A formal definition will be given in Section 3.1. A focused association rule makes use of only items that have higher frequency and correlation in the “respond” class. The search space is determined by “respond” records and items that occur infrequently in the “not_respond”. This prunes all “not_respond” records (to deal with the scarcity of the target class) and all items that occur frequently in the “not_respond” class (to deal with the high dimensionality).

In the second step, we convert focused association rules into a model for predicting the donation amount for a given customer. This involves determining how to cover customers using rules, pruning over-fitting rules that do not generalize to the whole population, and estimating the donation amount for rules, therefore, for customers. In the presence of Challenge 1, innovative solutions are needed because statistically insignificant rules could generate a significant profit. Our approach is to push the customer value into the model building/pruning so that the estimated profit over the whole population is maximized.

In the rest of the paper, the following terms are interchangeable: customer and record, responder and “respond” record, non-responder and “not_respond” record.

3 Algorithm

Historical records are stored in a relational table of m non-target variables $A_1, \ldots, A_m$ and two target variables $Class$ and V. $Class$ takes one of the “respond”/“not_respond” classes as the value. V represents a continuous donation amount. Given a set of records of this format, our task is to build a model for predicting the donation profit over current customers represented by the validation set in the KDD-CUP-98 dataset. Precisely, we want to maximize $\Sigma(V - \$0.68)$ over the current customers who are predicted to have a donation greater than the mailing cost $0.68. An implicit assumption is that current customers follow the same class and donation distribution as that of historical records. Since the donation amount V for a current customer is not known until the customer responds, the algorithm is evaluated using a holdout subset from the historical data, i.e., the validation set.

Algorithm 1 outlines the algorithm for building the model. There are three main steps: *Rule Generating*, *Model Building*, and *Model Pruning*. The Rule Generating step finds a set of good rules that capture features of responders. The Model Building step combines such rules into a prediction model for donation amount. The Model Pruning step prunes over-fitting rules that do not generalize to the whole population. We discuss these steps in detail.

Algorithm 1 The overall algorithm

```
Input: The learning set, minimum support and maximum support
Output: A model for predicting the donation amount
1: Rule Generating;
2: Model Building;
3: Model Pruning;
```

3.1 Step 1: Rule Generating

We discretize continuous non-target variables using the utility at http://www.sgi.com/tech/mlc before generating rules. After discretization, each value a_{i_j} is either a categorical value or an interval. We are interested in "respond" rules of the form

$$A_{i_1} = a_{i_1}, \ldots, A_{i_k} = a_{i_k} \rightarrow respond$$

that are potentially useful for discriminating responders from non-responders. The generality of a rule is defined by the percentage of the records that contain all the equalities on the left-hand side of the rule, called *support*. Despite many efficient algorithms for mining association rules (see (Agrawal et al., 1993, 1996, Agrawal and Srikant, 1994), for example), we encountered a significant difficulty in this step: to find "respond" rules, we have to set the minimum support well below 5%, i.e., the percentage of "respond" records in the dataset; however, with 481 variables and 95% records in the "not_respond" class, the number of "not_respond" rules satisfying the minimum support is so large that finding "respond" rules is similar to searching a needle from a haystack. Sampling techniques cannot reduce the "width" of records that is the real curse behind the long running time. This situation of extremely high dimensionality and extremely low target proportion also occurs in computational biology (Rigoutsos and Floratos, 1998), network intrusion detection and fraud detection (Joshi et al., 2001). We consider a simple but efficient solution to this problem by focusing on items that *occur frequently in "respond" records but occur infrequently in "not_respond" records*. Let D_r be the set of "respond" records and let D_n be the set of "not_respond" records.

Definition 3.1 (Focused association rules) The *support* of item $A_i = a_i$ in D_r or D_n is the percentage of the records in D_r or D_n that contain $A_i = a_i$. The *support* of a rule in D_r or D_n is the percentage of the records in D_r or D_n that contain all the items in the rule. Given a minimum support for D_r and a maximum support for D_n, an item $A_i = a_i$ is *focused* if its support in D_n is not more than the maximum support and its support in D_r is not less than the minimum support. A "respond" rule is a *focused association rule (FAR)* if it contains only focused items and its support in D_r is not less than the minimum support.■

Algorithm 2 Rule Generating

Input: D_r, D_n, the minimum support for D_r and the maximum support for D_n

Output: FARs

```
1:  /* compute the support in $D_n$ for items in $D_r$ */
2:  for all tuple $t$ in $D_r$ do
3:      for all item in $t$ do
4:          create a counter for the item if not yet created;
5:      end for;
6:  end for;
7:  for all tuple $t$ in $D_n$ do
8:      for all item in $t$ do
9:          increment the counter for the item if found;
10:     end for;
11: end for;
12: /* remove the items from $D_r$ whose support in $D_n$ exceeds the maximum support */
13: for all tuple $t$ in $D_r$ do
14:     remove the items from $t$ whose support in $D_n$ exceeds the maximum support;
15: end for;
16: /* find frequent "respond" rules in $D_r$ */
17: find "respond" rules above the minimum support in $D_r$ such as in (Agrawal et al., 1993);
```

In words, a FAR occurs frequently in D_r (as per the minimum support) but none of its items occurs frequently in D_n (as per the maximum support). Notice that FARs exclude the “respond” rules that as a whole do not occur frequently in D_n but some of its items does. This “incompleteness” trades for the data reduction achieved by pruning all non-focused items. For the KDD-CUP-98 dataset, this prunes all “not_respond” records, which accounts for 95% of the dataset, and all items that occur frequently in D_n, which accounts for 40%-60% of all items. In fact, for our purpose, the completeness of rules is not a concern, but finding some rules that can influence the final profit is. Our experiments show that the notion of FARs works exactly towards this goal.

Algorithm 2 finds FARs for given minimum support in D_r and maximum support in D_n. First, it computes the support in D_n for the items in D_r (line 1-11) and removes those items from D_r for which this support exceeds the maximum support (line 12-15). Then, it applies any association rule mining algorithm

Table 1. (a) Before applying maximum support. (b) After applying maximum support

D_r				
TID	A	B	C	V
p_1	a_1	b_1	c_3	$30.68
p_2	a_1	b_2	c_3	$50.68
p_3	a_1	b_2	c_1	$40.68
p_4	a_2	b_2	c_2	$20.68
p_5	a_2	b_1	c_3	$20.68

D_n				
TID	A	B	C	V
n_1	a_1	b_1	c_1	$0.00
n_2	a_2	b_1	c_3	$0.00
n_3	a_2	b_2	c_1	$0.00
n_4	a_2	b_1	c_3	$0.00
n_5	a_3	b_2	c_1	$0.00

(a)

D_r				
TID	A	B	C	V
p_1	a_1	/	c_3	$30.68
p_2	a_1	b_2	c_3	$50.68
p_3	a_1	b_2	/	$40.68
p_4	/	b_2	c_2	$20.68
p_5	/	/	c_3	$20.68

Count of items		
Item	Count in D_n	Count in D_r
a_1	1	3
a_2^*	3	2
b_1^*	3	2
b_2	2	3
c_1^*	3	1
c_2	0	1
c_3	2	3

(b)

such as (Agrawal et al., 1993) to the updated D_r to find "respond" rules above the minimum support (line 16-17). This association rule mining is expensive, but is applied to only "respond" records and only items whose support in D_n is not more than the maximum support. After finding the FARs, we add to the rule set the (only) "not_respond" rule of the form

$$\emptyset \rightarrow not\_respond.$$

This rule, called the *default rule*, is used only if a customer matches no FAR.

Example 3.1 Consider the database in Table 1(a). There are 10 records (tuples), 5 in D_r and 5 in D_n. Each record has 3 attributes A, B, C and donation V. Suppose that both minimum support for D_r and maximum support for D_n are 40%. The lower table in Table 1(b) shows the support count for each item in D_r. The items exceeding the maximum support in D_n (i.e., occur in more than 2 records in D_n) are marked with

Table 2. The FARs generated with minimum support and maximum support of 40%.

RID	Rules	Support in D_r
r_1	$\emptyset \rightarrow not\_respond$	$5/5 = 100\%$
r_2	$A = a_1 \rightarrow respond$	$3/5 = 60\%$
r_3	$B = b_2 \rightarrow respond$	$3/5 = 60\%$
r_4	$C = c_3 \rightarrow respond$	$3/5 = 60\%$
r_5	$A = a_1, B = b_2 \rightarrow respond$	$2/5 = 40\%$
r_6	$A = a_1, C = c_3 \rightarrow respond$	$2/5 = 40\%$

Table 3. Two models of $profit(r, t)$ (V is the donation amount in t)

	Rule r	Record t	Reward/Penalty model	Profit model
Case 1	"respond" rule	"respond" record	$V - 0.68$ (earned)	$V - 0.68$
Case 2	default rule	"respond" record	$-(V - 0.68)$ (not earned)	0
Case 3	"respond" rule	"not_respond" record	-0.68 (wasted)	-0.68
Case 4	default rule	"not_respond" record	0.68 (saved)	0

"*". The upper table of Table 1(b) shows D_r with such items removed. Table 2 shows the FARs found from D_r, plus the default rule. ■

In the rest of the paper, a "rule" refers to either a FAR or the default rule, $Supp(r)$ denotes the support of rule r in $D_r \cup D_n$, i.e., the percentage of all records containing both sides of the rule, $lhs(r)$ denotes the set of items on the left-hand side of rule r, $|lhs(r)|$ denotes the number of items in $lhs(r)$. We say that a rule r *matches* a record t, or vice versa, if t contains all the items in $lhs(r)$. We say that a rule r is *more general than* a rule r' if $lhs(r) \subseteq lhs(r')$.

3.2 Step 2: Model Building

Given a customer, we need to choose one rule to predict the donation amount on the customer. To maximize the profit generated, we prefer the rule that matches the customer and has the largest observed profit on the learning set. The observed profit of a rule r is the average profit generated on the learning records that match the rule. Let $profit(r, t)$ denote the profit generated by the prediction of r on a learning record t. The observed profit of r is defined as:

$$O\_avg(r) = \Sigma_t profit(r, t)/M,$$

where t is a learning record that matches r and M is the number of such records. A large $O\_avg(r)$ means that the customers (in the learning set) matched by r make a large donation on average.

Table 3 gives two models for defining $profit(r, t)$. The *Reward/Penalty* model rewards each dollar saved or earned and penalizes each dollar wasted or not earned. The *Profit* model simply measures the net profit generated. Let us explain each case in Table 3.

- Case 1: a responder t is predicted as a responder. Both models reward this decision by the net profit earned, i.e., $V - 0.68$.

- Case 2: a responder t is predicted as a non-responder. The Reward/Penalty model penalizes this decision by the loss of the supposedly earned dollars, i.e., $-(V - 0.68)$, and the Profit model does it by the zero profit generated.

- Case 3: a non-responder t is predicted as a responder. Both models penalizes this decision by the mailing cost wasted, i.e., -0.68.

- Case 4: a non-responder t is predicted as a non-responder. The Reward/Penalty model rewards this decision by the mailing cost saved, i.e., 0.68, and the Profit model does it by the zero profit generated (i.e., no mailing cost wasted).

The difference between the two models is that, for each non-responder prediction (i.e., Case 2 ad 4), there is zero profit generated in the Profit model, but there is the mailing cost saved (if the prediction is correct) or the customer value lost (if the prediction is wrong) in the Reward/Penalty model.

To maximize the profit on a current customer, we prefer the matching rule of the largest possible O_avg. The effect is predicting the profit using the most profitable customer group that matches a current customer. We formalize this preference below.

Definition 3.2 (Covering rules) For any two rules r and r', r is *ranked higher than* r'

- (Average profit) if $O\_avg(r) > O\_avg(r')$, or

- (Generality) if $O\_avg(r) = O\_avg(r')$, but $Supp(r) > Supp(r')$, or

- (Simplicity) if $Supp(r) = Supp(r')$, but $|lhs(r)| < |lhs(r')|$, or

- (Totality of order) if $|lhs(r)| = |lhs(r')|$, but r is generated before r',

in that order. Given a record t, a rule r is the *covering rule* of t, or r *covers* t, if r matches t and has the highest possible rank. ■

Table 4. Coverage and rank of rules

RID	Records matched	Records covered	O_avg	Ranking
r_5	p_2, p_3	p_2, p_3	\$45.00	1^{st}
r_6	p_1, p_2	p_1	\$40.00	2^{nd}
r_2	p_1, p_2, p_3, n_1	n_1	\$29.83	3^{rd}
r_3	p_2, p_3, p_4, n_3, n_5	p_4, n_3, n_5	\$21.73	4^{th}
r_4	p_1, p_2, p_5, n_2, n_4	p_5, n_2, n_4	\$19.73	5^{th}
r_1	p_1-p_5, n_1-n_5	$\emptyset$	\$0.00	6^{th}

Given a current customer, we use the covering rule of the customer to estimate the profit. We will discuss the profit estimation shortly. Though possibly matched by more than one rule, each record is covered by exactly one rule (i.e., the covering rule). To find the covering rule of a given record, we store rules of size k in a hash tree of depth k (Agrawal et al., 1993). Associated with each rule is the quadruple $< O\_avg, Supp, |lhs|, generation\_time >$ that determines the rank of the rule. Given a record t, we find the covering rule of t by finding all matching rules using the hash trees of depth smaller than the size of t. The covering rule of t is the matching rule of the highest possible rank.

A rule is useless if it matches a record, some rule of a higher rank also matches the record. Therefore, a useless rule has no chance to cover any record and can be removed without affecting prediction. Precisely, a rule is *useless* if some other rule is more general and ranked higher.

Example 3.2 Continue with Example 3.1. Rules are ranked by O_avg in Table 4, where the Profit model for $profit(r, t)$ is used. For example, r_2 matches 4 records p_1, p_2, p_3, and n_1. $O\_avg(r_2) = \sum_t profit(r_2, t)/4 = (\$30 + \$50 + \$40 - \$0.68)/4 = \$29.83$. O_avg for other rules is similarly computed. p_2 is matched by all 6 rules and is covered by r_5, the matching rule of highest rank. Similarly, the covering rules of other records can be determined. ■

3.3 Step 3: Model Pruning

The above rule ranking favors specific rules that match a small number of customers of a high profit. In the classic classification problem, such rules are pruned due to statistical insignificance. In the presence of inverse correlation between the likelihood to respond and the dollar amount generated by a response, extra care should be taken because valuable customers do not show up very often and pruning their rules could lead to the loss of significant profit. To address this issue, we propose pruning rules *on the basis of* *increasing the estimated profit over the whole population*. Below, we describe this new pruning method.

First, we explain how to estimate the profit of a rule r over the whole population; then, we give a method for pruning rules based on this estimation. The profit of r (over the whole population) can be estimated in two steps. First, we estimate the "hits" of r over the whole population. Second, we compute the profit of the estimated hits using the observations in the learning set. We borrow the *pessimistic estimation* (Clopper and Pearson, 1934, Quinlan, 1993) for estimating the "hits" of r.

Definition 3.3 Let $Cover(r)$ denote the set of learning records covered by r. Let M denote the number of records in $Cover(r)$, E of which do not match the class in r. ■

E/M is the observed error rate of r on the learning sample. To estimate the error rate of r over the whole population, we regard these E errors as observing E events in M trials, assuming that such events follow the binomial distribution. Given a confidence level CF, the probability that the real error rate of r in the whole population exceeds the upper limit $U_{CF}(M,E)$ is no more than $CF/2$.The exact computation of $U_{CF}(M,E)$ is less important and can be found in the C4.5 code (Quinlan, 1993), and a theoretical account can be found in (Clopper and Pearson, 1934). The idea is that a smaller sample size M is penalized by a larger upper limit $U_{CF}(M,E)$ to guarantee the specified confidence level CF. The default value of CF in C4.5 is 25%. If we use r to classify M customers randomly chosen from the whole population, we have $1 - CF/2$ confidence that the number of "hits" is at least $M \times (1 - U_{CF}(M,E))$, and the number of "misses" is at most $M \times U_{CF}(M,E)$

Consider a "respond" rule r. The average profit per hit in $Cover(r)$ is

$$avg_h(r) = \Sigma_t(V - 0.68)/(M - E)$$

for the "respond" records t in $Cover(r)$, where V is the donation amount in t. The average profit per miss in $Cover(r)$ is the cost of mailing to a non-responder, i.e., 0.68. We extend these averages to the above estimated hits and misses.

Definition 3.4 (Estimated profit) Assume that r covers M learning records, E incorrectly. The *estimated profit* of r is

$$Estimate(r) = \begin{cases} M \times (1 - U_{CF}(M,E)) \times avg_h(r) - M \times U_{CF}(M,E) \times 0.68 & \text{if } r \text{ is a "respond" rule} \\ 0 & \text{if } r \text{ is the default rule} \end{cases}$$

The *estimated average profit* of r, denoted $E\_avg(r)$, is $Estimated(r)/M$. The *estimated profit* of a model is $\Sigma_r Estimated(r)$ over all rules r (for $|D_r|+|D_n|$ customers randomly chosen from the whole population). ■

Algorithm 3 Model Pruning

Input: A set of rules

Output: The pruned covering tree

```
1: build the covering tree;
2: for all node $r$ in the bottom-up order do
3:     compute $Estimated(r)$;
4:     if $r$ is a non-leaf node and $E\_tree(r) \leq E\_leaf(r)$ then
5:         prune the subtree at $r$;
6:     end if;
7: end for;
8: return the unpruned rules;
```

Notice the difference between $O\_avg(r)$ and $E\_avg(r)$. $O\_avg(r)$ is the average profit *observed* for the learning records that are *matched* by r. The matching rule of largest $O\_avg(r)$ is the covering rule of a given record. $E\_avg(r)$ is the average profit *estimated* for the records in the whole population that are *covered* by r. We use $E\_avg(r)$ to estimate the profit generated by each prediction of r over the whole population. $E\_avg(r)$ depends on $O\_avg(r)$ to define the notion of covering rules.

Now we return to the main topic of pruning over-fitting rules to maximize $\Sigma_r Estimated(r)$ over unpruned rules r. If a specific rule is pruned, we choose the general rule of highest rank to cover the records that were covered by the pruned rule. This specific/general rule relationship is defined by the covering tree below.

Definition 3.5 (Covering tree) In the *covering tree*, a rule r' is the *parent* of a rule r if r' is more general than r and has the highest possible rank. If a rule r is pruned, the parent of r replaces r as the covering rule of the records previously covered by r. ■

A child rule always has a higher rank than its parent; otherwise, the parent rule will cover all records matched by the child rule and the child rule is useless, which contradicts the fact that all useless rules have been removed. The most general default rule is the root of the covering tree. As we walk down the tree, rules are increasingly more specific and ranked higher.

Algorithm 3 shows the algorithm for pruning rules in a bottom-up order of the covering tree. It first builds the covering tree (line 1). This can be done as follows. Assume that rules of size k are stored in a hash tree of depth k. We examine rules of larger size before examining rules of smaller size. For each rule r of size k, we find all general rules by searching the hash trees of depth smaller than k. If the general rule of highest

Table 5. $Estimated(r)$ **before and after pruning**

	Before pruning at the rule	After pruning at the rule
RID	$Cover(r)$ (M, E), $Estimated(r)$	$Cover(r)$ (M, E), $Estimated(r)$
r_5	p_2, p_3 (2, 0), \$44.32	
r_6	p_1 (1, 0), \$6.99	
r_2	n_1 (1, 1), -\$0.68	p_1, p_2, p_3, n_1 (4, 1), \$70.50
r_3	p_4, n_3, n_5 (3, 2), \$2.10	
r_4	p_5, n_2, n_4 (3, 2), \$2.10	
r_1	$\emptyset$ (0, 0), \$0.00	$\emptyset$ (0, 0), \$0.00 (pruning not performed)

possible rank has a lower rank than r, we make it the parent rule of r; otherwise, we discard r because it is useless. In this step, we also compute M and E for every rule r in the covering tree. For this, we scan every record t in $D_r \cup D_n$, find the covering rule r of t using the hash trees, and increment M for r. If t and r does not match in class, we also increment E for r.

After building the covering tree, the algorithm examines the nodes in the bottom-up order. At a leaf node r, it computes $Estimated(r)$. At a non-leaf node r, it computes the estimated profit for the subtree at r, denoted by $E\_tree(r)$, and the estimated profit of r after pruning the subtree, denoted by $E\_leaf(r)$. $E\_tree(r)$ is $\Sigma Estimated(u)$ over all nodes u within the subtree at r. $E\_leaf(r)$ is the same as $Estimated(r)$, except that r now covers all the learning records covered by the subtree at r. If $E\_tree(r) \leq E\_leaf(r)$, it prunes the subtree at r by making r a new leaf node in the covering tree and removing the rules in the subtree from the hash trees. If $E\_tree(r) > E\_leaf(r)$, it does nothing at r. The nodes outside the subtree at r are not considered because their estimated profit remains unchanged. Essentially, the bottom-up pruning has the effect of cutting off some lower portion of the covering tree to maximize $\Sigma_r Estimated(r)$ over remaining rules r.

Example 3.3 Let us build the covering tree for Example 3.2. Consider rule r_5 for example. r_1, r_2 and r_3 are more general than r_5, but r_2 has the highest rank among them. So, r_2 is the parent of r_5. In this way, we build the covering tree on the left of Figure 1.

Table 5 shows $Estimated(r)$ before and after the pruning at r. For example, r_5 covers correctly p_2 and p_3, so $M = 2$ and $E = 0$. The estimated number of misses is $2 \times U_{CF}(2, 0) = 2 \times 0.50 = 1.00$, and the estimated number of hits is $2 \times (1 - U_{CF}(2, 0)) = 2 \times 0.50 = 1.00$. $avg_h(r_5) = [(50.68 - 0.68) + (40.68 - 0.68)]/(2 - 0) = \$45.00$. From Definition 3.4, $Estimated(r_5) = 1.00 \times avg_h(r_5) - 1.00 \times 0.68 = \$44.32$.

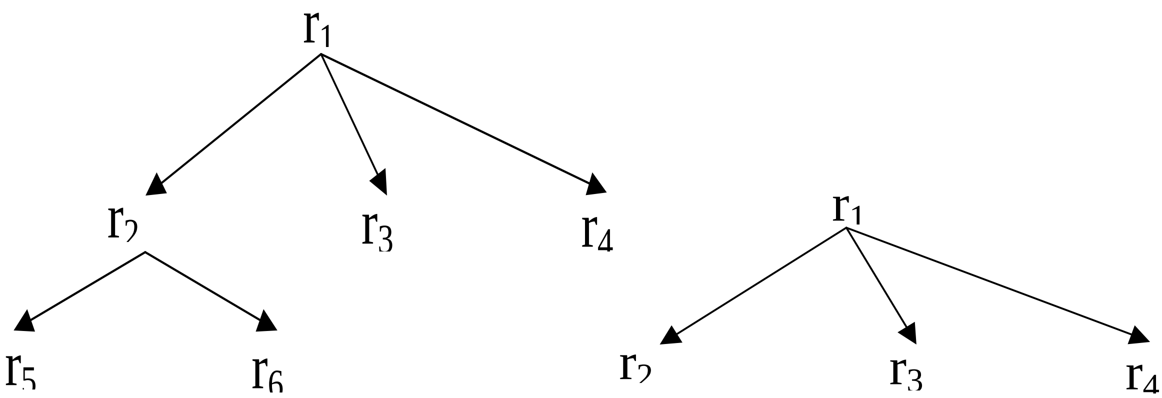

Figure 1. Left: the covering tree before pruning. Right: the covering tree after pruning.

After examining nodes r_5 and r_6, the bottom-up pruning examines the node r_2. $E\_tree(r_2) = Estimated(r_2) + Estimated(r_5) + Estimated(r_6) = -0.68 + 44.32 + 6.99 = \$50.63$. Pruning the subtree at r_2 makes r_2 cover p_1, p_2, p_3 and n_1, and $M = 4$ and $E = 1$. In this case, the estimated number of misses is $4 \times U_{CF}(4,1) = 4 \times 0.55 = 2.20$, the estimated number of hits is $4 \times (1 - U_{CF}(4,1)) = 4 \times 0.45 = 1.80$, and $avg_h(r_2) = [(50.68 - 0.68) + (40.68 - 0.68) + (30.68 - 0.68)]/(4 - 1) = \$40.00$. Following Definition 3.4,

$$E\_leaf(r_2) = Estimated(r_2) = 1.80 \times 40.00 - 0.68 \times 2.20 = \$70.50.$$

Since $E\_tree(r_2) \leq E\_leaf(r_2)$, the subtree at r_2 is pruned.

After examining nodes r_2, r_3, r_4, the bottom-up pruning examines the root r_1. We have

$$E\_tree(r_1) = \textstyle\sum_{i=1}^{4} Estimated(r_i) = 0.00 + 70.50 + 2.10 + 2.10 = \$74.70.$$

If the subtree at r_1 is pruned, r_1 would cover all records in $D_r \cup D_n$. Since r_1 is a "not_respond" rule, $Estimated(r_1) = 0$ (Definition 3.4), and $E\_leaf(r_1) = 0$. We have $E\_tree(r_1) > E\_leaf(r_1)$. So, no pruning is done at r_1. The final pruned covering tree is shown on the right of Figure 1. ■

We can prove the following optimality of the above bottom-up pruning. A *cut* of a tree contains exactly one node on each root-to-leaf path in the tree. A cut generates a tree by making the nodes in the cut as the leaf nodes.

Theorem 3.1 The pruned covering tree has the maximum $\Sigma_r Estimated(r)$ among all pruned covering trees generated by a cut of the given covering tree.

Proof: It essentially follows from the fact that the pruning decision at a sibling node is independent of the decisions at other sibling nodes. This implies that, if the pruning at a sibling node increases estimated profit, so does it in any "optimal cut" because it does not affect the pruning at other sibling nodes. Therefore, the pruning should be done in any “optimal cut”. ■

The cost of Model Pruning consists of building the covering tree and pruning the tree. Pruning the covering tree takes only one scan of the nodes in the tree. The cost of building the covering tree involves finding all general rules for each rule and finding the covering rule for every learning record. These costs are comparable to the cost of counting the support of candidates for itemsets in the Apriori algorithm (Agrawal and Srikant, 1994).

3.4 Choosing threshold values

A remaining issue is how to choose the minimum support (for D_r) and the maximum support (for D_n). Our method is less sensitive to specific rules because of its own pruning step, i.e., Step 3. For this reason, a smaller minimum support is preferred to avoid losing profitable rules. The choice of maximum support is dictated by how much resource we can afford for mining “respond” rules. The rule of thumb is that, for a smaller minimum support, the Rule Generating step becomes more time/space-consuming, and a smaller maximum support should be used to exclude more items.

We suggest the following procedure for choosing the minimum support and maximum support as follows. We split the learning set into *building set* and *testing set*, and run Algorithm 1 on the building set. Some initial minimum support, usually 1%, and initial maximum support, which is usually the percentage of “respond” records in the learning set, are used. After building the model, we compute the *sum of actual profit*, as defined in Section 4, on the testing set. If the current run results in an increase in the sum of actual profit, we rerun the algorithm with a reduced minimum support and, if necessary, a reduced maximum support to allow efficient rule generating. This procedure is repeated several times until the sum of actual profit cannot be increased “significantly”. The model built in the last run is returned. We shall experimentally study this procedure in Section 4. The testing set serves to tune parameters in our method, therefore, should not be confused with the validation set.

3.5 Making prediction

The prediction model is given by the set of unpruned rules returned in the last run. To make prediction on a customer t, we use the hash trees to find the covering rule r of t. The decision on the customer is “contact the customer” if and only if r is a “respond” rule and the predicted profit $E\_avg(r)$ (Definition 3.4) is positive.

4 Validation

In this section, we validate the proposed method using the standard split of the KDD98-learning-set (95,412 records) and KDD98-validation-set (96,367 records) used by the KDD competition (KDD98, 1998a). The KDD98-learning-set is used for learning a model. In our method, we split the KDD98-learning-set randomly into 70% for the building set (66,788 records, 3,390 "respond" records) and 30% for the testing set (28,624 records, 1,453 "respond" records), as described in Section 3.4. The testing set is used for tuning the minimum and maximum support in our method, not for evaluation purpose. The evaluation is performed using the standard KDD98-validation-set, which is held out from the learning phase of all algorithms. The competition criterion is the *sum of actual profit* on the KDD98-validation-set, defined as $\Sigma_t(V - 0.68)$ for all validation records t predicted to have a positive profit, where V is the donation amount in t. We report our results based on the Profit model in Table 3. No significant difference is found on the Reward/Penalty model.

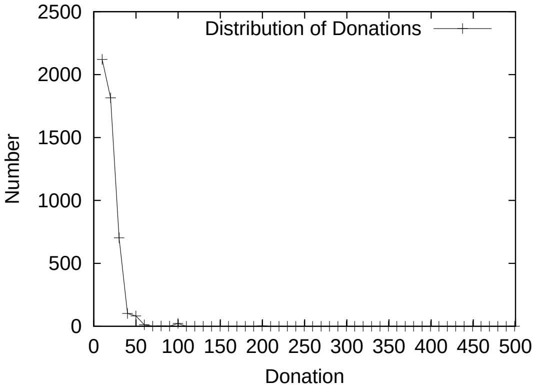

Figure 2. The distribution of donation

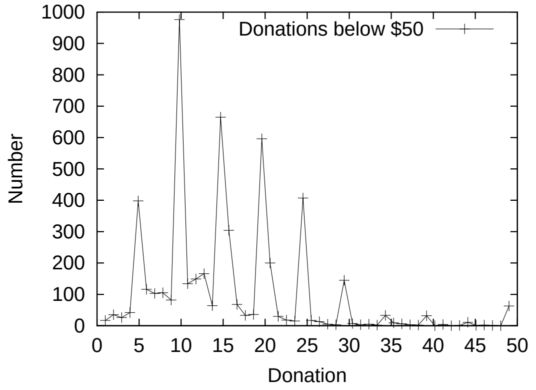

Figure 3. The distribution of donation below $50

Table 6. Comparison with published results

Category	Algorithm	Sum of Actual Profit	# Mailed	Average Profit
	Our Algorithm	**$20,693**	**23,437**	**$0.88**
KDD-CUP-98 Results *in* (KDD98, 1998b)	GainSmarts (The winner)	$14,712.24	56,330	$0.26
	SAS/Enterprise Miner (#2)	$14,662.43	55,838	$0.26
	Quadstone/Decisionhouse (#3)	$13,954.47	57,836	$0.24
	ARIAI/CARRL (#4)	$13,824.77	55,650	$0.25
	Amdocs/KDD Suite (#5)	$13,794.24	51,906	$0.27
MetaCost *in* (Domingos, 1999) and (Zadrozny and Elkan, 2001)	Smoothed C4.5 (sm)	$12,835		
	C4.5 with curtailment (cur)	$11,283		
	Binned naive Bayes (binb)	$14,113		
	Average (sm, cur)	$13,284		
	Average (sm, cur, binb)	$13,515		
Direct Cost-Sensitive *in* (Zadrozny and Elkan, 2001)	Smoothed C4.5 (sm)	$14,321		
	C4.5 with curtailment (cur)	$14,161		
	Binned naive Bayes (binb)	$15,094		
	Average (sm, cur)	$14,879		
	Average (sm, cur, binb)	$15,329		
	Maximum possible profit	$72,776	4,873	$14.93
	Mail to Everyone	$10,548	96,367	$0.11

We compare our method with three categories of published results. The first includes the top five results from the KDD-CUP-98 competition. As pointed out by (KDD98, 1998b)), these contestants used state-of-the-arts techniques such as *2-stage*, *multiple strategies*, *combined boosting and bagging*. The second category includes the results produced by the MetaCost technique (Domingos, 1999). The third category includes the results produced by the direct cost-sensitive decision-making (Zadrozny and Elkan, 2001). The results from the latter two categories are taken from (Zadrozny and Elkan, 2001), which implemented MetaCost and direct cost-sensitive decision-making using advanced techniques for probability estimation and donation estimation, including *multiple linear regression, C4.5, naive Bayes, smoothing, curtailment, binning, averaging*, and *Heckman* procedure. Interested readers are referred to (Zadrozny and Elkan, 2001) for more details.

The evaluation results in Sections 4.1-4.4 are based on the KDD98-validation-set, which has 96,367 records and 4,873 “respond” records. Figure 2 and Figure 3 show the distribution of donation amount for “respond” records. There is a clear inverse correlation between the probability that a customer responds and the dollar amount generated by a response.

4.1 Sum of actual profit

The summary of comparison is shown in Table 6 based on the KDD98-validation-set. The first row (in bold face) is our result. Next come the three categories of published results: the top five contestants of the KDD-CUP-98 as reported in (KDD98, 1998b), five algorithms of MetaCost and five algorithms of direct cost-sensitive decision making as reported in (Zadrozny and Elkan, 2001).

Our method generated the sum of actual profit of \$20,693. This is 41% more than the KDD-CUP-98 winner (\$14,712.24), 47% more than the best profit of MetaCost (\$14,113), and 35% more than the best profit of direct cost-sensitive decision making (\$15,329). According to the analysis in (Zadrozny and Elkan, 2001), a minimum difference of \$1,090 is required to be statistically significant. Our performance gain far exceeds this requirement. Our average profit per mail is \$0.88. This is 3.38 times that of the KDD-CUP-98 winner, and 8 times that of the Mail to Everyone Solution. Compared to the KDD-CUP winner, we generated 41% more profit by predicting less than an half number of contacts. (Zadrozny and Elkan, 2001) did not report the number of mailed, so we cannot compute their average profit. These higher total profit and average profit suggest that the proposed method is highly successful in focusing on valuable customers. This success is credited to the novel feature extraction based on the global search of association rule mining, and the profit estimation that pushes the customer value as the first class information.

4.2 Profit lift

We extend the concept of “lift” in the literature (Ling and Li, 1998, Masand and Shapiro, 1996) to evaluate the “profit lift” of our result. In the *cumulative lift curve* (Ling and Li, 1998, Masand and Shapiro, 1996), validation records are ranked by the estimated probability of belonging to the “respond” class, and each point (x, y) on the curve represents that the top x percent of the ranked list contains y percent of all actual responders. In the *cumulative profit lift curve*, each point (x, y) represents that the top x percent of the ranked list generates y percent of the total profit. Thus, the cumulative lift curve is a special case of the cumulative profit lift curve when every responder generates the same profit. Figure 4 shows the cumulative profit lift curve of our result. For example, the top 20% of the ranked list generates 42% of the total actual profit, giving the profit lift of 2.1. The bend toward the upper-left corner suggests that our method ranks valuable customers toward the top of the list.

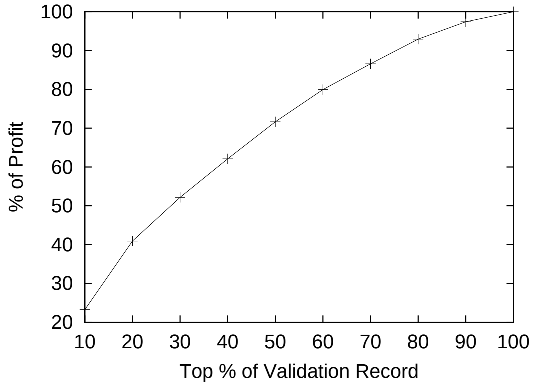

Figure 4. The accumulative profit lift curve

4.3 Classification

Table 7 shows the confusion matrix for the KDD98-validation-set. 2,813 of the 4,873 responders are predicted as responders (i.e., contacted), and 71,389 of the 91,494 non-responders are predicted as non-responders (i.e., not contacted), giving the "hit rate" of 57.7% on responders and 78.0% on non-responders. In other words, the hit rate for responders is more than 10 times the percentage of responders in the data (i.e., 5%). This strongly suggests that our method has achieved the goal of identifying valuable customers. This is further confirmed by the striking similarity between the number of identified responders in Figure 5 and the number of actual responders in Figure 3.

Table 7. The confusion matrix

	not contacted	contacted
non-responder	$71,389$	$20,105$
responder	$2,060$	$2,813$

4.4 The top 10 rules

Figure 6 shows the 10 "respond" rules in terms of the profit generated on the KDD98-validation-set. The top portion describes the involved variables, copied from the data source. Each rule is listed in the following format:

$$Rule\#: \; Profit \;\; Supp \;\; Conf \;\; A_{i_1} = a_{i_1}, \ldots, A_{i_k} = a_{i_k}$$

where $Profit$ is the total profit generated on validation records by the rule, $Supp$ and $Conf$ are the support (in number of records) and confidence of the rule. A_{i_j} is a non-target variable and a_{i_1} is either a single value

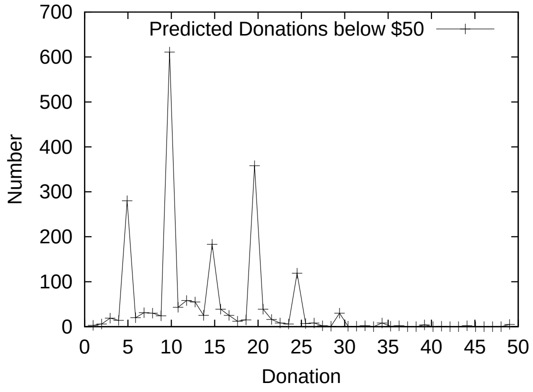

Figure 5. The predicted responders below $\$$50

Table 8. Choosing maximum support/minimum support

Max sup.	Min sup.	#Rules	#Rules Aft. Prune	Time (second)	#Mailed	Sum of Act. Prof.
5% (40% remain)	3%	956	714	825	5470	$3757
	2%	31850	1117	1019	6270	$4566
	1.5%	46348	1363	2227	8559	$4690
3% (25% remain)	1%	1819	973	716	6785	$5722
	0.8%	2510	1064	754	6898	$5640
	0.5%	6530	1303	913	7796	$6733
	0.3%	16446	1522	1067	7812	$6760
	0.2%	**47665**	**1863**	**2178**	**8739**	**$7382**
	0.15%	71626	2317	2841	8383	$7103
1% (8% remain)	0.1%	4451	793	1001	5802	$4757
	0.05%	13871	1173	975	6017	$5128

or a range of the form $[a, b]$. We have omitted the right-hand side $respond$ from all rules.

Examining these rules reveals some interesting points. First, neither the most general rule 8 (with the largest support) nor the most confident rule 9 is the most profitable rule, i.e., rule 1. Second, the support of rules is very small. Recall that the learning set, $D_n \cup D_r$, has 66,788 records, 3,390 of which are “respond” records, D_r. The smallest support of 7 here corresponds to 0.01% of $D_n \cup D_r$ and 0.2% of D_r. With 481 variables, mining the whole learning set with the minimum support of 0.01% is infeasible according to our experience. However, by mining FARs from D_r only, we can set the minimum support to 0.2%, in which case the mining task is feasible. Third, these rules are explicit in terms of customer demographic information, thus, are potentially useful for devising campaign strategies.

4.5 Choosing threshold values

Now we report how the maximum support/minimum support were selected in our algorithm. At first, we included all items by setting maximum support at 100%. The rule generating (i.e., association rule mining)

```
Variables              Meanings
---------------------------------------------------------------------------------
AGE904                 Average Age of Population
CHIL2                  Percent Children Age 7 - 13
DMA                    DMA Code
EIC16                  Percent Employed in Public Administration
EIC4                   Percent Employed in Manufacturing
ETH1                   Percent White
ETH13                  Percent Mexican
ETHC4                  Percent Black < Age 15
HC6                    Percent Owner Occupied Structures Built Since 1970
HHD1                   Percent Households w/ Related Children
HU3                    Percent Occupied Housing Units
HUPA1                  Percent Housing Units w/ 2 thru 9 Units at the Address
HVP5                   Percent Home Value >= $50,000
NUMCHLD                NUMBER OF CHILDREN
POP903                 Number of Households
RAMNT_22               Dollar amount of the gift for 95XK
RFA_11                 Donor’s RFA status as of 96X1 promotion date
RFA_14                 Donor’s RFA status as of 95NK promotion date
RFA_23                 Donor’s RFA status as of 94FS promotion date
RHP2                   Average Number of Rooms per Housing Unit
TPE11                  Mean Travel Time to Work in minutes
WEALTH2                Wealth Rating
```

```
Rule Profit  Supp  Conf   Rule
---------------------------------------------------------------------------------
1    $81.11  13    0.11   ETHC4=[2.5,4.5], ETH1=[22.84,29.76], HC6=[60.91,68.53]
2    $61.73  8     0.17   RFA_14=f1d, ETH1=[29.76,36.69]
3    $47.07  12    0.12   HHD1=[24.33,28.91], EIC4=[33.72,37.36]
4    $40.82  16    0.12   RFA_23=s2g, ETH13=[27.34,31.23]
5    $35.17  11    0.16   EIC16=[11.25,13.12], CHIL2=[33,35.33], HC6=[45.69,53.30]
6    $28.71  7     0.16   RHP2=[36.72,40.45], AGE904=[42.2,44.9]
7    $24.32  10    0.14   HVP5=[56.07,63.23], ETH13=[31.23,35.61],
                                          RAMNT_22=[7.90,10.36]
8    $19.32  31    0.08   NUMCHLD=[2.5,3.25], HU3=[66.27,70.36]
9    $17.59  8     0.25   RFA_11=f1g, DMA=[743,766.8], POP903=[4088.208,4391.917],
                                          WEALTH2=[6.428571,7.714286]
10   $9.46   9     0.23   HUPA1=[41.81+,], TPE11=[27,64,31.58]
---------------------------------------------------------------------------------
```

Figure 6. The top 10 “respond" rules found

took significantly long time for any minimum support small enough to produce sufficient "respond" rules; or to finish the rule generating within a reasonable amount of time, we had to use a high minimum support that generated very few "respond" rules, therefore, very low profit. Therefore, we have to exclude unpromising items using a lower maximum support. Our algorithm iteratively adjusted the maximum support and minimum support based on the feedback on the testing set. The last column in Table 8 shows the sum of actual profit on the testing set for several settings of maximum support/minimum support. Recall that the testing set is a 30% random sample of the KDD98-learning-set.

In general, reducing maximum support/minimum support increases the sum of actual profit. Reducing the minimum support increases the number of rules, and reducing the maximum support allows more efficient mining. After reaching 3% for maximum support and 0.2% for minimum support, i.e., the row in bold face, further reducing these thresholds will decrease the sum of actual profit, due to the excessive over-fitting of specific rules. Therefore, our algorithm chooses 3% and 0.2% for maximum support and minimum support.

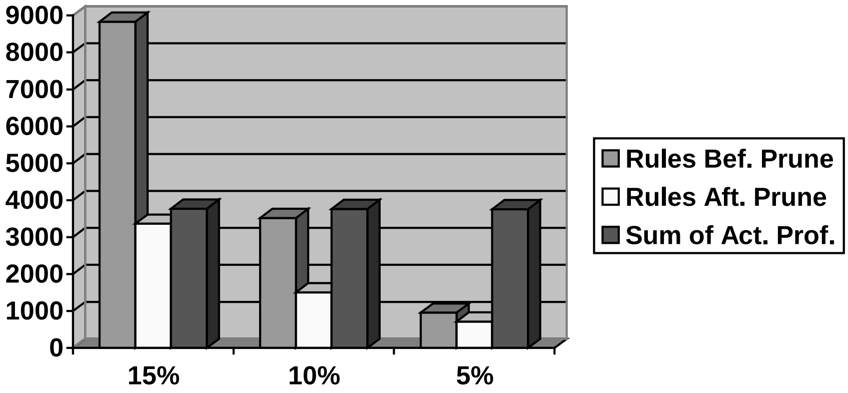

Figure 7. The effectiveness of maximum support

This study reveals the effectiveness of maximum support in reducing the number of items. The first column of Table 8 gives the remaining file size in percentage after applying the maximum support. A large portion of items was removed by using maximum support, which is extremely important for scaling up the association rule mining. A question is whether such dimension reduction will reduce the profitability of the final model. To answer this question, Figure 7 compares the models built at varied maximum support of 15%, 10% and 5%, with minimum support fixed at 3%. A smaller maximum support sharply reduces the number of rules, but not the sum of actual profit (on the testing set). In fact, many rules pruned by the maximum support are ranked lower than some general rules. Such rules are never used according to our ranking of rules.

Table 8 also shows the number of rules before and after the model pruning in Step 3 (i.e., columns 3 and

4). The pruning effect is dramatic, especially when the initial model is large. For example, at the maximum support of 3% and the minimum support of 0.2%, the model has 47,665 rules before the pruning and only 1,863 rules after the pruning.

5 Conclusion

Direct marketing becomes increasingly important in retail, banking, insurance and fundraising industries. In a recent KDnuggets poll on the question “where do you plan to use data mining in 2002”, direct marketing/fundraising was the second most voted among 15 application areas (KDnuggets, 2001). A challenge to the prediction problem in direct marketing is the inverse correlation between the likelihood to buy and the dollar amount to spend, which implies that the class probability based ranking will rank valuable customers low rather than high! Previous approaches are “after the fact” in that they re-rank the probability based ranking using the customer value in the second step. Another challenge is the extremely high dimensionality and extremely low proportion of the target class. In such cases, finding rules to distinguish the target class from non-target classes is similar to finding a needle from a haystack.

In this paper, we push the customer value as the first class information. Our approach is to estimate directly the profit generated on a customer without estimating the class probability. This methodology opens up new possibilities for profit estimation. In particular, we use association rules to summarize customer groups and to build a model for profit prediction. The advantage of the association rule approach is its scalability of finding correlated features that may never be found in a local search. The evaluation on the well known, large, and challenging KDD-CUP-98 task shows a breakthrough result.

References

Agrawal, R., T. Imilienski, and A. Swami (1993). Mining association rules between sets of items in large datasets. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD’93)*, Washington, D.C., USA, pp. 207–216.

Agrawal, R., H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo (1996). Fast discovery of association rules. In *Advances in knowledge discovery and data mining*, pp. 307–328. AAAI/MIT Press.

Agrawal, R. and R. Srikant (1994). Fast algorithm for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB’94)*, Santiago de Chile, Chile, pp. 487–499.

Bitran, G. and S. Mondschein (1996). Mailing decisions in the catalog sales industry. *Management Science 42*, 1364–1381.

Brijs, T., G. Swinnen, K. Vanhoof, and G. Wets (1999). Using association rules for prodcut assortment decisions: a case study. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'99)*, San Diego, CA, USA.

Bult, J. R. and T. Wansbeek (1995). Optimal selection for direct mail. *Marketing Science 14*, 378–394.

Clark, P. and T. Niblett (1989). The CN2 induction algorithm. *Machine Learning Journal 3*(4), 261–283.

Clopper, C. and E. Pearson (1934). The use of confidence or fiducial limits illustrated in the case of the binomial (http://www.jstor.org/journals/bio.html). *Biometrika 26*(4), 404–413.

Desarbo, W. and V. Ramaswamy (1994). Crisp: customer response based iterative segmentation procedures for response modeling in direct marketing. *Journal of Direct Marketing 8*, 7–20.

Domingos, P. (1999). Metacost: A general method for making classifiers cost sensitive. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'99)*, San Diego, CA, USA, pp. 155–164.

Domingos, P. and M. Pazzani (1996). Beyond independence: conditions for the optimality of the simple bayesian classifier. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML'96)*, Bari, Italy.

Joshi, M., R. Agarwal, and V. Kumar (2001). Mining needles in a haystack: classifying rare classes via two-phase rule induction. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data (SIGMOD'01)*, Santa Barbara, California, USA, pp. 91–102.

KDD98 (1998a). The kdd-cup-98 dataset. In *http://kdd.ics.uci.edu/databases/kddcup98/kddcup98.html*.

KDD98 (1998b). The kdd-cup-98 result. In *http://www.kdnuggets.com/meetings/kdd98/kdd-cup-98.html*.

KDnuggets (2001). Kdnuggets poll results: data mining applications in 2002. In *http://www.kdnuggets.com/news/2001/n25*.

Levin, N. and J. Zahavi (1996). Segmentation analysis with managerial judgement. *Journal of Direct Marketing 10*, 28–37.

Ling, C. and C. Li (1998). Data mining for direct marketing: problems and solutions. In *The Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*, New York, New York, USA, pp. 73–79.

Masand, B. and G. P. Shapiro (1996). A comparison of approaches for maximizing business payoff of prediction models. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, Portland, Oregon, USA, pp. 195–201.

Michalski, R. S. (1969). On the quasi-minimal solution of the general covering problem. *3*, 125–128.

Potharst, R., U. Kaymak, and W. Pijls (2002). Neural networks for target selection in direct marketing. *Neural Networks in Business: Techniques and Applications*, 89–110.

Quinlan, J. (1993). *C4.5: programs for machine learning*. San Mateo, CA, USA: Morgan Kaufmann.

Quinlan, R. J. (1983). Learning efficient classification procedures and their application to chess endgames. *Machine Learning: An Artificial Intelligence Approach 1*, 463–482.

Rigoutsos, I. and A. Floratos (1998). Combinatorial pattern discovery in biological sequences. *Bioinformatics 14*(2), 55–67.

Savasere, A., E. Omiecinski, and S. Navathe (1998). Mining for strong negative associations in a large database of customer transactions. In *Proceedings of the Fourteenth International Conference on Data Engineering (ICDE'98)*, Orlando, Florida, USA, pp. 494–502.

Tan, P. N., V. Kumar, and J. Srivastav (2000). Indirect association: mining higher order dependencies in data. In *The Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'00)*, Lyon, France, pp. 632–637.

Wang, K. and M. Y. Su (2002). Item selection by hub-authority profit ranking. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'02)*, Edmonton, Alberta, Canada, pp. 652–657.

Wang, K., S. Zhou, and J. Han (2002). Profit mining: from patterns to actions. In *Proceedings of the Eighth International Conference on Extending Database Technology (EDBT'02)*, Prague, Czech Republic, pp. 70–87.

Zadrozny, B. and C. Elkan (2001). Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (SIGKDD'01)*, San Francisco, CA, USA, pp. 204–213.