Pattern-based Target Selection applied to Fund Raising

Wim Pijls, Rob Potharst, and Uzay Kaymak

Erasmus University, P.O.Box 1738, 3000 DR Rotterdam
{pijls, potharst, kaymak}@few.eur.nl
http://www.few.eur.nl/few/research/eurfew21/

Abstract. This paper proposes a new algorithm for target selection. This algorithm collects all frequent patterns (equivalent to frequent item sets) in a training set. These patterns are stored efficiently using a compact data structure called a trie. For each pattern the relative frequency of the target class is determined. Target selection is achieved by matching the candidate records with the patterns in the trie. A score for each record results from this matching process, based upon the frequency values in the trie. The records with the best score values are selected. We have applied the new algorithm to a large data set containing the results of a number of mailing campaigns by a Dutch charity organization. Our algorithm turns out to be competitive with logistic regression and superior to CHAID.

1 Introduction

Since the first half of the nineties direct marketing has become an important application field for data mining. In direct marketing, companies or organizations try to establish and maintain a direct relationship with their customers in order to target them individually for specific product offers or for fund raising. Large databases of customer and market data are maintained for this purpose. The customers or clients to be targeted in a specific campaign are selected from the database given different types of information such as demographic information and information on the customer's personal characteristics like profession, age and purchase history. Apart from commercial firms and companies, charity organizations also apply direct marketing for fund raising. Charity organizations do not have customers in the regular sense of the word, but they must be able to trace people who are more likely to donate money in order to optimize their fund raising results.
Many techniques have been applied to select the targets in commercial applications, such as decision tree methods like CHAID or CART [7], statistical regression [5], neural computing [14] and fuzzy clustering [12]. In this paper, we propose a new algorithm which we apply to fund raising by a Dutch charity organization. There is a major difference between direct mailing in a commercial environment and direct mailing for the benefit of a charity organization: the response rate to a commercial direct marketing campaign seldom exceeds 5%,

whereas a charity campaign among a group of known supporters often triggers a much higher response. Modeling of charity campaigns/donations has recently been considered by Jonker et al. [8].

Target selection techniques can be roughly divided into two types: segmentation techniques and scoring techniques. Segmentation is the traditional tool of marketing science. The set of customers is divided into segments, each segment having characteristic value ranges for features such as gender, age, salary, etc. Algorithms based upon decision trees such as CHAID [9], C4.5 [11] and CART [4] fall into the segmentation category, since the subset of the input space defined by each leaf of the tree may be viewed as a market segment. The methods of this type have the advantage of comprehensibility. The other approach is to assign a score to each individual customer, indicating the likelihood that the customer is a responder. Such techniques, where each customer is given an individual score, appear to perform better in terms of response rates. Regression methods such as logistic regression, neural networks, and some fuzzy set approaches fall into this category, essentially. The technique we propose which we call PatSelect, is an offshoot of our classification algorithm [10], which is based on the data mining method of finding frequent item sets [1]. The new algorithm considers patterns of attribute values. Such patterns are similar to segment groups, but they are not segments in the proper sense in that they overlap strongly. Many overlapping patterns are examined to classify a client into a market group. In this sense, the new method may be regarded as a segmentation technique, exhibiting the advantage of comprehensibility. Due to the huge number of patterns considered, a much more finely grained picture of the market appears. As we will see, PatSelect performs on an equal level with pure scoring techniques such as logistic regression. So, it combines the transparency of segmentation with the scoring benefits of regression methods.
The outline of this paper is as follows. In section 2 we introduce the notion of frequent patterns which explains the foundation of our method. In section 3, the algorithm we propose is outlined and discussed. Finally, our proposed algorithm is applied to a large real world database of a well known Dutch charity organization in section 4. This case is extensively described including the modeling stage. The conclusions are given in section 5.

2 Frequent patterns

In this section, we discuss some definitions and facts on frequent patterns. A data set is a set of *records* or *cases*, where each record consists of an n-tuple (n is fixed) of discrete values. The positions in a record correspond to attributes. Suppose that we have a data set of customers with $n = 4$ and the four attributes are: 'income', 'married', 'children', 'credit worthy'. An instance of a record in this data set is *(5, yes, 3, yes)*, which means that the customer related to this record has income group equal to 5, is married, has 3 children and is credit worthy. As mentioned above, we require discrete values. If an attribute has continuous

a	b	c	$class$
1	1	3	2
1	1	3	2
1	2	1	1
1	2	1	1
1	2	3	2
1	2	3	1
2	1	3	2
2	1	3	2
3	1	2	1
4	2	1	1
4	2	1	2
4	2	3	1
4	2	3	1

Fig. 1. An example: a data set S

values, discretization is required. One attribute is appointed to be the target attribute. The values which are taken by the target attribute are called *classes* or *class labels*. Figure 1 shows a simple example of a data set, which has four attributes: a target attribute and three other ones called a, b and c.

Suppose that a data set D has $n-1$ non-target attributes $x_1, x_2, \ldots, x_{n-1}$, and target class label y. A *pattern* (also called an *item set* in data mining literature) is defined as a series of m equalities of the form $(x_{i_1} = v_{i_1}, x_{i_2} = v_{i_2}, \ldots, x_{i_m} = v_{i_m})$. where v_{i_k} is a value that can be taken by attribute x_{i_k}, $1 \leq k \leq m$. Note that a pattern refers to only non-target attributes. A record $r = (x_1, x_2, \ldots, x_{n-1}, y)$ is said to be a *supporter* of a pattern $P = (x_{i_1} = v_{i_1}, x_{i_2} = v_{i_2}, \ldots, x_{i_m} = v_{i_m})$, if r matches pattern P, i.e., attribute x_{i_k} occurs in r and has value v_{i_k} for each k, $1 \leq k \leq m$. The *support* of a pattern P, denoted by $supp(P)$, is defined as the number of supporters of P. Given a threshold or minimal support (denoted by *minsup*) a pattern is called *frequent* if $supp(P) \geq minsup$. A pattern P' is called a subpattern of P, if each equality in P' is also included in P. Clearly, any subpattern of a frequent pattern is frequent as well.

The best-known algorithm for finding frequent patterns is Apriori [1]. In this paper, a *hash tree* (a tree with a hash table in each node) is proposed to represent itemsets. We utilize a different data structure which replaces the hash nodes by nodes with completely filled arrays of dynamic length. This data structure is equivalent to a trie [2]. It stores the full collection of frequent patterns in an efficient and compact way. An example of a trie is shown in Figure 2, which displays the set of all frequent patterns in the data set of Figure 1 for *minsup*$= 2$. Each path from a square in the root of the trie to a square in any other node (not necessarily a leaf) represents a frequent pattern, e.g. $(a = 4, b = 2)$ or $(a = 4, b = 2, c = 1)$. The patterns $(a = 3)$ and $(c = 2)$ and any extensions of those patterns are infrequent and hence not included in the trie of Figure 2. Similarly to the hash tree in Apriori, the trie is built up level by level. First, the root consisting of frequent patterns of length 1 is constructed. Next, the

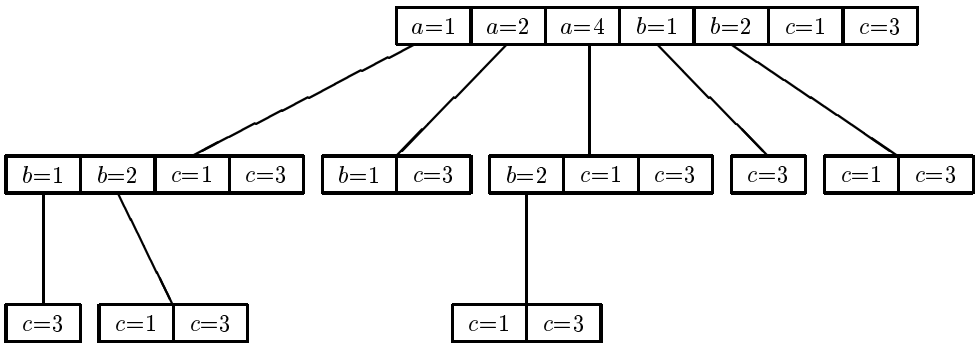

Fig. 2. The frequent patterns of S stored into a trie.

nodes representing the frequent patterns of length 2 (the children of the root) are added, etc. For each level of the trie a complete pass through the data is carried out. Thus, the maximum number of passes through the data is $n-1$.

As mentioned before, $supp(P)$ is equal to the number of the supporters of P. For a given class y and a pattern P, the numbers of supporters of P with a class label y is denoted by $supp(y, P)$, The *relative frequency* or briefly the frequency of a class y is denoted by $fr(y, P)$ and is defined as:

$$fr(y, P) = \frac{\text{number of supporters of } P \text{ with class } y}{\text{total number of supporters of } P} = \frac{supp(y, P)}{supp(P)} \qquad (1)$$

The following examples illustrate the definitions. Let $(c = 3)$ be pattern P and let $(b = 2, c = 3)$ be pattern Q in the data set of Figure 1. The following equalities hold for P: $supp(P) = 8$, $supp(1, P) = 3$, $supp(2, P) = 5$, $fr(1, P) = 3/8$, $fr(2, P) = 5/8$. For Q we have: $supp(Q) = 4$, $supp(1, Q) = 3$, $supp(2, Q) = 1$, $fr(1, Q) = 3/4$, $fr(2, Q) = 1/4$. The relative frequencies and the supports of each pattern are stored in the trie during the build-up of the trie.

In [10], it is shown that for classification a trie is an appropriate alternative to a decision tree as used in C4.5, CART or CHAID. The set of all possible values taken by the non-attribute values $(x_1, x_2, \ldots, x_{n-1})$, is called the input space, denoted by $\mathcal{X}$. In a decision tree as well as in a trie of patterns each node n or rather each path from the root to a node n corresponds to a subset of $\mathcal{X}$. However, there is a major difference between a trie of patterns and a decision tree. In a decision tree, the subsets corresponding to the leaves make up a partition of the input space $\mathcal{X}$. An algorithm using a decision tree classifies an input record $x = (x_1, x_2, \ldots, x_{n-1})$ by looking for the unique subset including x in that partition. In a trie, there may be many patterns that match x. When using a trie to classify x, one looks for all patterns matching x and one chooses one that is best according to a certain criterion.

3 A new algorithm for target selection

In this section, our new target selection algorithm *Patselect* is introduced. This algorithm is derived from a classification method, called Patmat, which was published recently in [10].
Target selection addresses the following problem: which records are likely to belong to a particular target class t? In order to describe the algorithm, we need to introduce a score function $score(r)$ for each record r. The function is defined as:

$$score(r) = \max\{fr(t, P) \mid r \textit{ is a supporter of } P\}. \quad (2)$$

This definition implies that, in order to calculate the score for a record r the entire trie must be scanned to find all frequent patterns that are supported by r. The pattern with the highest response frequency is picked out. Due to the efficient structure of the trie, the scan goes very fast. So, using the trie is an essential issue in the new algorithm.
To find a subset of N records which are likely to belong to t, perform the following steps:

procedure *selection* (*size* N)
    for every record r in the test set **do**
        compute $score(r)$;
    sort all records r by decreasing $score(r)$;
    select the topmost N records;

When sorting the records by decreasing $score(r)$, ties are resolved randomly. The only parameter that remains to be set when we run PatSelect on a data set is the minimum support parameter *minsup*. We have experimented extensively with many machine learning data sets and always found best results with a value of *minsup* of about 0.5% to 1% of the training set. With higher values the patterns become too coarse and with lower values the results on a test set deteriorate because of overfitting. For reasonably large data sets such a value of *minsup* also guarantees that the relative response frequencies we calculate for all the frequent patterns are stable estimates of the underlying response probabilities.

When comparing PatSelect to other Target Selection methods, it should be mentioned that all methods can be written in the form of the above procedure *selection*. What differs in different methods is the score function that is utilized. For instance, when we use linear regression, the score function is a linear function of the non-target attributes. When we use a segmentation method such as CHAID, the score function for a specific record can be defined as the relative response frequency in the segment that the record belongs to. Of course, what is meant is the relative response frequency in the training set. Now, we can see why PatSelect indeed is likely to perform better than a tree method: PatSelect does not look at one segment that the record belongs to, it looks at all patterns that match the record, and it picks the one with the highest response rate.

4 A fund raising application

In this section we apply the proposed target selection method to a large database from a well-known Dutch charity organization. This database contains information regarding the responses of more than 700 000 supporters to 26 mailing campaigns over a period of six years. For each mailing campaign, the supporters who were mailed in that campaign are recorded, as well as the amount they have donated (zero or more) in Dutch guilders. The mailing dates for each campaign are also known. Also recorded is the date at which the supporter has donated money in response to a particular mailing. The total recorded data amounts to a database of about 400MB.

Feature Selection The first modeling step we undertook with these data is to construct useful features from the raw data. In target selection, so-called RFM-variables (regarding the Recency, Frequency and Monetary value of the donations) capture relevant information for modeling the response behavior of customers. Constructing such RFM-variables is common practice in direct marketing, see for instance [3]. We constructed seven features from the supporter donation history data, of which two can be seen as recency features (R1 and R2), two as frequency features (F1 and F2), and three as features concerning monetary value (M1 to M3). Note that depending on the actual mailing campaign that is used for obtaining the modeling data, each of these features can be calculated for a different moment in time. We denote this moment as "*now*". In other words, the features must be re-computed for every mailing campaign that is considered, i.e. whose response is being modeled. The following list shows the seven features that we have used:

R1: the number of weeks since the supporter's last response to a mailing before *now*,
R2: the number of months since the supporter's first-ever donation,
F1: the fraction of the mailings the supporter has responded to,
F2: the median of the response times of the supporter in the period before *now* over all mailings the supporter has responded to,
M1: the average donated amount over all responded mailings before *now*,
M2: the amount the supporter donated at his/her last response before *now*,
M3: the average amount that the supporter has donated per year until *now*.

Data set construction We knew that the charity organization mails all active supporters in their database at least once a year. We decided to base our model on the two most recent full mailing campaigns: the first mailing in 1998 and the first mailing in 1999, the most recent year in the database. Before calculating the features for both years, we split the total raw data set randomly into two equal parts: a training set and a test set. For both of these we calculated the features for the 1998 and for the 1999 mailing. Thus we ended up with four different data sets: a training set and a test set for the 1998 mailing of about 166 000 records, and a training and test set for the 1999 mailing of size 186 000. The

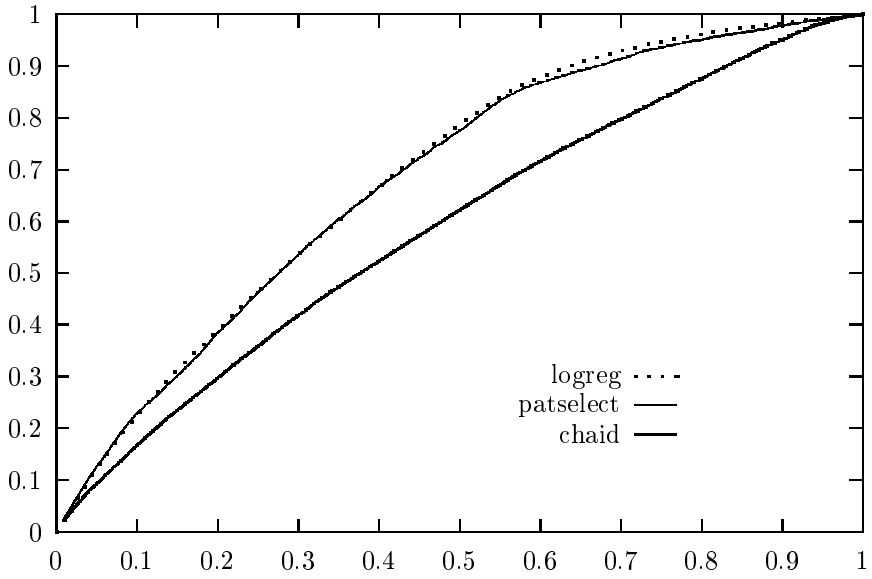

Fig. 3. The gain chart for the '98 test set

1999 training set was not used in the experiments described here. Each of these data sets contains data for the supporters on eight variables: the seven RFM features explained in the previous paragraph and calculated for the corresponding mailing, and the response (yes or no) to that mailing. In all of these data sets the percentage of responders is approximately 30%. The reason the final data set sizes are all well under half of 700 000 is that we left out the following categories: 1) the automatically paying supporters, 2) supporters that never received any mailing or never responded to any mailing, 3) supporters that had set a limit to their donations beforehand and 4) those supporters that did not take part in either the '98 or the '99 mailing.

Experiments and results Using the '98 training set we built a trie of frequent patterns and we used this trie two times: once we scored the '98 test set with it, using the PatSelect algorithm, and once we scored the '99 test set in the same way. We used a value of 800 for the parameter *minsup*, which amounts to about 0.5% of the training set. Building the trie costs around 10 seconds runtime, using an Apriori like method for building the trie and scoring a complete test set around 5 seconds on our Pentium III desktop computer running Windows NT. Patselect requires discrete attribute values. So, we needed to discretize the seven RFM features. First, we discretized the 1998 training set using the entropy-based method of [6]. The cut points provided by this process were also applied to the test sets. The number of categories was restricted to 8, since we noticed that a larger number gave poor results. For R1, F1, and F2, the maximum of 8 cate-

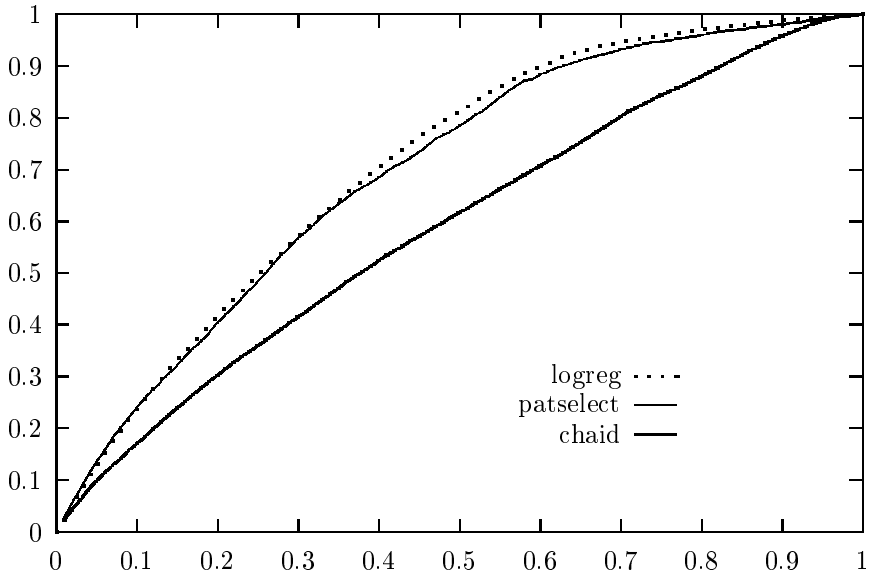

Fig. 4. The gain chart for the '99 test set

gories was achieved. Further, we had 5 categories for R2, 6 for M1 and M3 and 4 for M2. The series of n categories for each feature was numbered from 0 to $n-1$. The results are shown in Figure 3 and 4 in the form of so-called *gain charts*, an often used technique in the marketing world. An entry (x, y) in a gain chart means, that if a fraction x of the total database is mailed, then a fraction y of the responders is reached. Thus, we want the curve of a particular mailing to be as high as possible, since that means that we need to spend little effort to gain a high response. In order to be able to compare our method with commonly used methods, the same sequence of steps was also performed using logistic regression and CHAID. When ordering scores, ties were resolved randomly, as was the case in PatSelect. The results are shown in the same figures. Using CHAID with a maximum tree depth of 3 and a minimum leaf size of 200, we arrived at a tree with 89 leaves. If we use PatSelect or logistic regression, we learn from these gain charts that by mailing only 30% of the clients we will reach over 50% of the responders. It appears that PatSelect performs at a level comparable to logistic regression while CHAID lags behind clearly.

To illustrate the transparency of the PatSelect method, consider Table 1, in which we collected some results of the PatSelect selection process for the '98 test set. In particular, we show some of the highest-scoring groups. We read from this table *e.g.* that there were 1226 records in the '98 test set that received a score of .8852 associated with the pattern (R1 = 1, R2 = 4, F1 = 7). These records included 1084 responders, yielding a response rate of 88.4%. In this way,

score	R1	R2	F1	F2	M1	M2	M3	*number*	*hits*	*rate*
.8943	1	4	7	-	5	1	5	839	738	87.9
.8852	1	4	7	-	-	-	-	1226	1084	88.4
.8372	1	3	7	5	4	0	5	882	749	84.9
.8321	1	3	7	-	-	-	-	616	501	81.3
.8244	1	2	7	4	-	-	5	529	424	80.2
.8176	1	-	7	5	5	1	5	945	729	77.1
.8129	1	-	7	5	-	-	-	1618	1207	74.6

Table 1. Some high-scoring patterns of the '98 test set.

we identify the old-time (R2 = 4, the highest category), steadily donating (F1 = 7, also the highest category) supporters, who did not donate very recently (R1 = 1, the second lowest category) as a high-interest group for our next mailing. Thus, PatSelect not only assigns a score to each case in the database, but it also generates the pattern that resulted in this score. These patterns can be analyzed like we did in this example.

5 Concluding remarks

In this paper, we developed a new technique for target selection. This technique was applied successfully to a large real-world data set made available by a charity organization. It appears that the new technique is competitive with the best available methods for this problem, while at the same time it leads to transparent results by indicating high interest groups in the data. In the near future, we hope to apply our new method to data sets taken from other areas.

Acknowledgement We thank Jedid-Jah Jonker and Philip-Hans Franses for making available the data for this project and their pleasant and stimulating cooperation.

References

1. R. Agrawal, and R. Srikant, *Fast Algorithms for Mining Association Rules*, Proceedings of the 20th Int'l Conference on Very Large Databases, Santiago, Chili, September 1994.
2. A. V. Aho, J.E. Hopcroft and J.D. Ullman, *Data Structures and Algorithms*, pp. 163-169, ISBN 0-201-00023-7, Addison-Wesley Publishing Company, 1983.
3. C.L.Bauer, A direct mail customer purchase model, *Journal of Direct Marketing*, Vol. 2 (1988), pp. 16–24.
4. L. Breiman, J.H. Friedman, R.A. Olshen and C.J.Stone, *Classification and Regression Trees*, Wadsworth 1984, reprinted by Chapman and Hall, New York, 1993.
5. J.R. Bult and T.J. Wansbeek, *Optimal Selection for Direct Mail*, Marketing Science, Vol. 14, pp. 378–394, 1993

6. U. M. Fayyad and K.B. Irani, *Multi-interval discretization of continuous-valued attributes for classification learning,* IJCAI-93, pp. 1022-1027, 1993.
7. D. Haughton and S. Oulabi, *Direct Marketing Modeling with CART and CHAID*, Journal of Direct Marketing, Vol. 7 (1993) pp. 16–26.
8. J.-J. Jonker, R. Paap and P.H. Franses, *Modeling Charity Donations,* Technical Report 2000-07/A, Econometric Institute, Erasmus University Rotterdam, 2000.
9. G.V. Kass, *An Exploratory Technique for Investigating Large Quantities of Categorical Data*, Applied Statistics, Vol. 29 (1980) pp. 119–127.
10. W. Pijls and R. Potharst, *Classification based upon Frequent Patterns*, in: D. Lukose and G.J. Williams (eds), Proceedings of the Symposium on the Application of Artificial Intelligence in Industry, pp.87–94. Deakin University, Geelong, 2000, also available as: Technical Report ERS-2000-40-LIS, ERIM Report series, Erasmus University, the Netherlands. URL: `http://www.eur.nl/WebDOC/doc/erim/erimrs20001020162258.pdf`
11. R.Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1992.
12. M. Setnes and U. Kaymak, *Fuzzy modeling of client preference from large data sets: an application to target selection in direct marketing*, IEEE Transactions on Fuzzy Systems, Vol. 9, Nr 1, February 2001.
13. Ian H. Witten and Eibe Frank, *Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, 2000.
14. J. Zahavi and N. Levin, *Applying neural computing to target maketing,* Journal of Direct Marketing, Vol. 11, pp.5-22, 1997