Combining Target Selection Algorithms in Direct Marketing

Sjoerd van Geloven

April, 2002

Abstract

The aim of this work is to show that the performance of target selection models can be improved by using a combination of existing target selection algorithms. We present an approach in which combinations of algorithms provide better results than algorithms used stand-alone. Combining algorithms can be seen as an intelligent operation. The generalizing power of an algorithm is bounded by the assumptions underlying the algorithm. Systematic errors are made accordingly because the real world does not obey this assumptions in general. By using several algorithms at once, we believe that these systematic errors can be partly averaged out.
Four algorithms are used: linear and logistic regression, a feed forward back propagation neural network and a fuzzy modeling algorithm. The combinations are tested on a real-life data set. The fruitful combinations are shown and their suitability in general is discussed.

Keywords: Classification, response modeling, performance improvement.

1 Introduction

In direct marketing, it is important to know which prospects are interested in a specific offer and which customers would be annoyed by receiving the same mailing. If these two groups can be properly distinguished, this will on the one hand increase profit, because the costs per order drop, and on the other hand decrease customer annoyance. Several target selection approaches have been proposed through the years: traditional human-driven segmentation methods involving RFM (recency, frequency and monetary) variables [13], tree-structured "automatic" segmentation models such as Automatic Interaction Detection (AID) [8] [11] [12], Chi Square AID (CHAID) [4], Classification and Regression Trees (CART) [7] and C4.5 [18], linear statistical models such as linear [21], multiple regression [17] and discriminant analysis [19], non-linear discrete choice logit [5] [14] and probit [16] models, neural networks [15] and fuzzy modeling techniques [3].

This paper tries to give insight in a specific approach, by which target selection in direct marketing could obtain better results. The approach suggested is to combine two or more stand-alone algorithms, so that the limitations of applying one specific algorithm can be surpassed. The combination of Chaid and logit has been proposed by Magidson [10]. Levin and Zahavi [20] evaluated the performance of automatic tree classifiers (including CHAID) versus the judgmentally-based RFM and FRAC (Frequency, Recency, Amount of money and Category of product) methods and logistic regression. While there has been intensive research on the algorithms used stand-alone (for example [22]), other possible combinations are not reported.

A target selection algorithm tries to give each prospect a score which represents the likelihood of responding. The approach presented here, is to apply different techniques, hence algorithms, to score the prospects. Subsequently, these scores are assigned a weight, which represents an indication of importance, to each algorithm before summing them up resulting in a final indication of prosperity.

The outline of this paper is as follows. Section 2 describes the target selection problem in direct marketing. In section 3 suggestions are made regarding a number of configurations in which algorithms could co-operate. The most attractive one is explained, and an application of this combination of multiple algorithms in a real world business problem is given in section 4. Finally, in section 5 the results are compared to previously obtained results and section 6 gives conclusions and recommendations for further research.

2 Target Selection in Direct Marketing

Direct marketing is a powerful tool for companies to increase profit, especially when one knows his customers well.

The advantage of direct marketing compared to other marketing activities is that it seeks a direct response from pre-identified prospects. It allows companies to direct their energies toward those individuals who are most likely to respond. This selectivity is necessary to enable the use of high cost – high impact marketing programs such as personal selling in the business-to-business environment [23]. A second fundamental characteristic of direct marketing is that it generates individual response by known customers. This enables companies to construct detailed customer purchase histories, which tend to be far more useful than traditional predictors based on geodemographics and psychographics [23].

How can the most valuable customers or prospects be targeted? This question corresponds to target selection in direct marketing. The analysis of a direct marketing campaign usually entails two stages. At first, *feature selection* must determine the variables that are relevant for a specific target selection problem. Secondly, the rules for selecting profitable prospects should be determined, given the relevant features. We shall refer to this second stage by *client scoring.*

Target selection models are evaluated by using gain chart analysis. Gain chart analysis orders the members of the target population by the index prosperity given by a target selection method, from high to low [1]. Prospects with similar index values can then be divided in groups of equal size (this size can be one) and the average response per group is calculated. If the costs per mailing and the profit per order are known, those prospects can be selected for the future mailing for whom the average group returns exceed expected costs of the mailing.

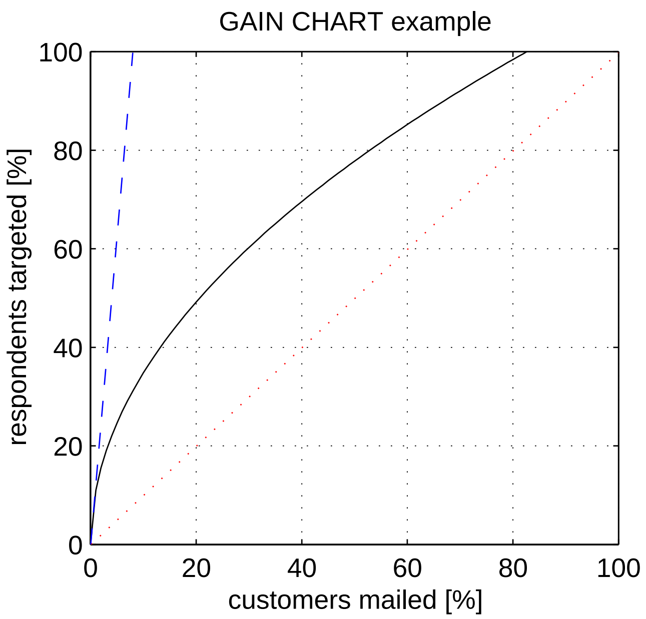

Figure 1: Example gain chart

In this paper, we stop after constructing the so-called gain chart. Each client is labeled and given a score by a target selection method which represents the likelihood of response, also referred to as index of prosperity. Subsequently, the prospects are ordered by this score from high to low. In a data set used to test different target selection algorithms is listed whether a prospect responded or not. This information can be used to calculated the number of respondents in every group of customers. In a gain chart the ordered customers are plotted against the cumulative number of respondents targeted. So, if all respondents are assigned a high prosperity index and none-respondents a low degree of likelihood to purchase, this would represent the ideal case plotted in figure 1 as the dashed line. This line indicates that in this example circa 8 percent of the customers are respondents and they are all successfully predicted. A purely random mailing or no model used, results in the dotted straight line from the lower left to the upper right. The goal of target selection is to determine a target selection model such that the corresponding gain chart shifts from the dotted reference line to the ideal dashed line. A typical gain chart is drawn as the solid curve.

3 Combining the Algorithms

The justification for investigating the generalizing power of the presented configuration of multiple algorithms originated from the observation that a stand-alone algorithm is bounded by the assumptions underlying that algorithm. Although the extra gain can be marginal, every improvement is worth to look into because the typical response is low and especially in the high cost – high impact environment every extra properly targeted client means an increase of profit. Put in other words, the idea behind target selection is that the behaviour of prospects can be predicted by a theoretical relation. The combination of multiple algorithms tries to overcome algorithm specific limitations by deploying several ones in constructing the final client score.

As mentioned in the previous section, target selection can be divided into two stages: feature selection and client scoring. Although in some cases both stages can be handled in the same computational procedure, it is instructive to treat the stages separately. Magidson [10] reported that selecting features using Chaid and scoring by logit is a powerful combination. This procedure can be applied in general: one configuration selects the important features and a second configuration uses these features for the client scoring. Because there are usually a lot of attributes in a direct marketing database, feature selection is necessary to identify the important features. Every algorithm selecting features, however, suffers from algorithm specific assumptions: normally distributed data, for instance. Because the total number of attributes is high, several feature selecting algorithms can be applied. These rarely come up with exactly the same feature subsets. If we assume that we have sufficient feature selecting algorithms (a domain expert can also be an "algorithm"), we have the availability of several feature subsets which inhibit relevant attributes to the target selection problem. The next question is how to assign a score to each client, given these feature subsets. One option is to construct one large feature set, with all features present in the feature sets given by the feature selecting algorithms. The disadvantage of this approach is that the unique structures of the former feature sets are lost. Another related problem, is what configuration should be used to score the clients only using the features present in this set? Other options of combining feature sets before scoring the prospects will have to conquer the same problems. The solution is simple: score all feature sets separately by algorithms which are known or expected to do well. This results in different scores for each client in the database. Assign all scores a weight depending on the expected performance by the specific configuration and multiply the scores with these weights before adding them up resulting in a final client score.

Mathematically expressed, if we have a data set with N customers and K attributes, and R feature selecting algorithms (FSA_r) selecting R feature subsets FT_r with $k(r) < K$ features each, we also need R scoring algorithms (SA_r) giving each customer $n \leq N$ a score sc_{nr} and the total score for client n is given by:

$$SC_n = \sum_r \alpha_r sc_{nr}, \tag{1}$$

in which α_r is the weight factor for scoring algorithm r. In the ideal case, the sum of these weights equals 1:

$$\sum_r \alpha_r = 1. \tag{2}$$

There are three special cases:

1. The feature selecting algorithm can be the same for all R scoring algorithms, hence the same feature (sub)set is used for different scoring configurations. Example: finding the optimal number of neurons in the hidden layer of a Artificial Neural Network.

2. The scoring algorithm can be the same for all R feature selecting algorithms. Example: finding the best feature subset for logistic regression as scoring algorithm.

3. The feature selecting algorithm is absent or the same as the scoring algorithm and R equals one. This is target selection in the "old-fashioned" way.

Next to the special cases, by manipulating the weight factors, one can build an algorithm selecting–target selection configuration. The total method is summarized in the flowchart given in AD1. The only problem left is how to determine appropriate values for the α's.

Setting the Weight Factors

The weight factor gives a configuration a degree of importance with respect to the other configurations used in a combination. Different configurations rarely come up with scores on the same value scale. So, the second task of the weight factor is transforming the scores given by the different configurations to the same scale. In order to establish a proper distinction between both tasks the weight factor can be split up into two factors: one representing the relative degree of importance and the other to scale the scores.

$$\alpha_r = \beta_r \gamma_r \tag{3}$$

In equation 3 the weight factor α is split into β, which represents the degree of importance and γ which is the scaling factor. The latter can easily be computed:

$$\gamma_r = \frac{N}{\sum_n sc_{nr}} \tag{4}$$

Four different sets of weight factors are used. The first, straightforward, way is to give all α's the same value: $1/R$. The second way is to set all β_r to 1 and calculate the γ_r's according to equation 4. In order to satisfy (2), the resulting scores SC_n can be divided by the total sum of the scale factors, such that new scaled SC_n values are obtained:

$$SC_{n,new} = \frac{SC_{n,old}}{\sum_r \gamma_r} \tag{5}$$

The third way to determine the weight factors is to apply a procedure which optimizes the β's, such as linear regression. The drawback of this approach is that some weight factors can become negative. This can be prevented by using a linear regression with positive coefficients.The final way to set the β weights is based on domain expertise. A domain expert can be someone with expert knowledge of or intensive experience with the algorithms, the (kind of) data set or, ideally, both. If there are indications that one algorithm performs better on a data set than others, the weight factor β for this algorithm can be increased accordingly.

4 Application: CoIL Challenge

The approach presented in the previous section has been used on the data set subject of study in the CoIL Challenge 2000. The task in this competition was to predict potential caravan policy buyers from the client data of an insurance company. The train set consist of 5822 client records each with 85 attributes plus the dependent variable. The score set, which will be used for an out-of-sample test has 4000 entries. In order to raise the generalizing power of the model, the train set is randomly divided into three sets, such that the overall percentage of respondents was kept. Two sets are, in turn, used for training and the third as a first validation. Finally, a fourth model is built by adding all three sets. By doing so, four different models are used, which on turn can be treated as one model by assigning each model weights. In figure 2 the procedure is drawn. The models mentioned in this figure represent the configuration of figure 3.

The performance measures on the first validation sets (”val” 1 to 3) are not numerically given in this paper. They can be a first indication which model trained on the cross sets performs best. The performances that we will report are labeled ”sc” 1 to 3, the performance of the three models based on the cross sets, ”score” 4, the performance of the model based on the entire train set and ”avg score”, the three cross models added up in the same way we combine different algorithms. We give two examples.

Example 1

In this combination four algorithms are used. The first algorithm is linear regression applied on all 85 variables (no feature selection). The second one

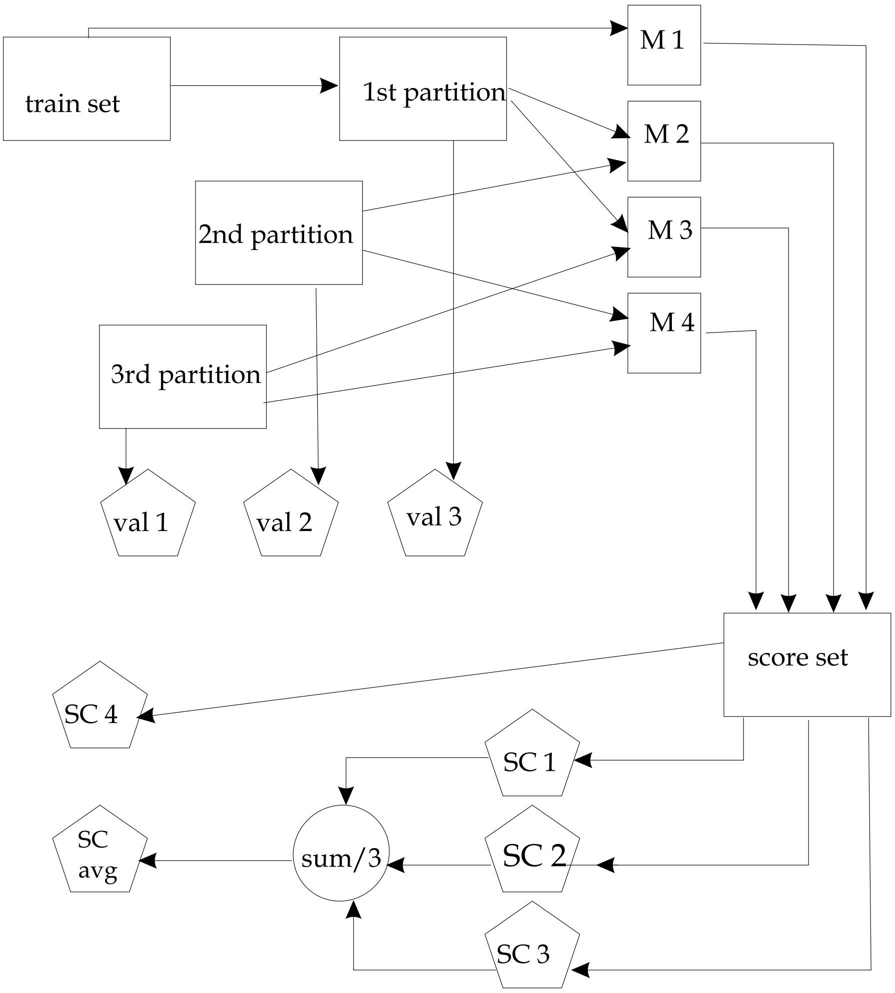

Figure 2: CoIL example procedures

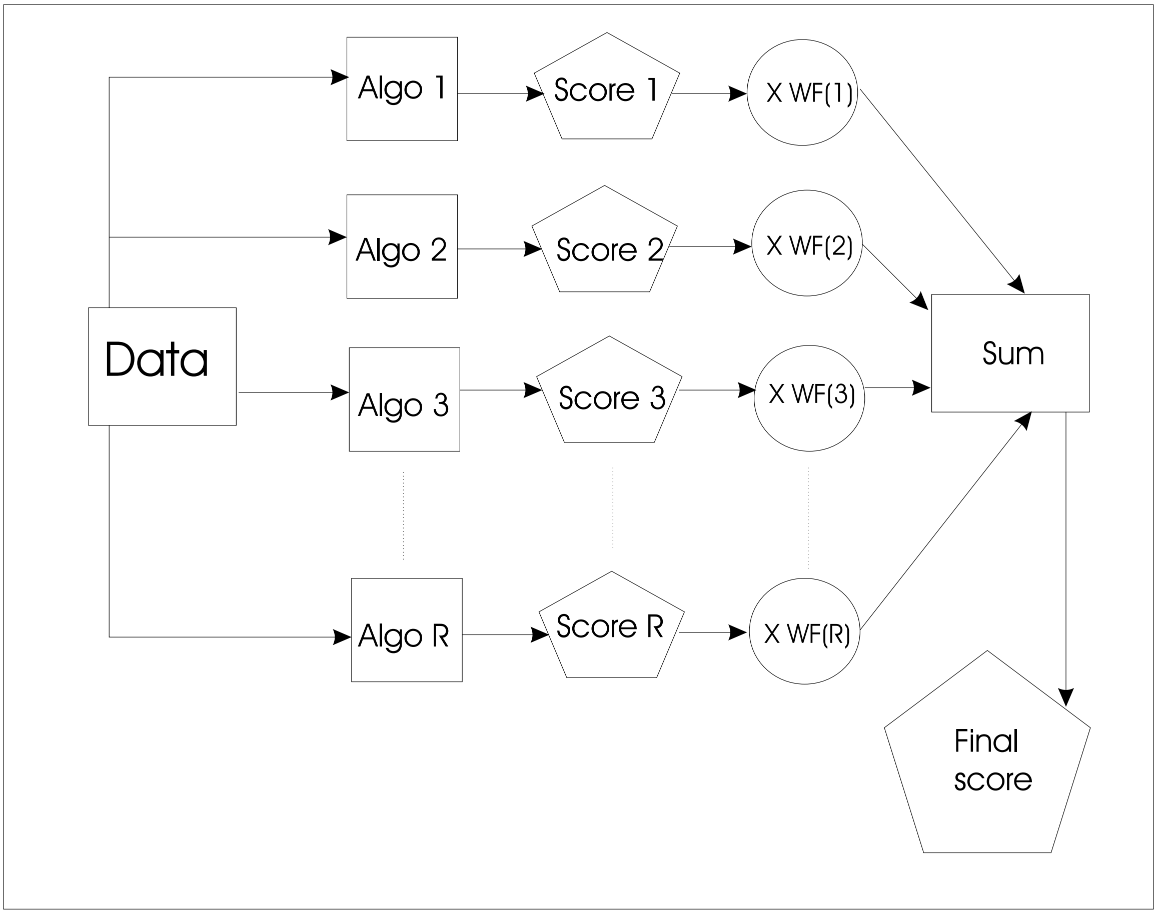

Figure 3: Methodology

is logistic regression, on a feature subset ($k(2) = 8$) also build by logit using a t-test to test whether a coefficient significantly differed from zero. The third one is a feed-forward back-propagation neural network with 2 neurons in the hidden layer deployed on a feature subset based on the degree of absolute correlation to the dependent variable where interaction effects between variables are accounted for ($k(3) = 8$), trained in 500 epochs. The fourth and final one is given by a fuzzy modeling technique [3] with 5 initial clusters for each variable ($k(4) = 8$). The different values for the weight factors are given in table 1. In the first column the four different sets of weight factors are listed, the γ's are the same for all listed β values. Note that the resulting α values are scaled except the "dom exp" -β values. The resulting α's can be scaled by applying equation 4, where the γ_r is substituted by α_r. In this way, the weights set by the domain expert (labeled "dom exp") are easier to explain. They are chosen given the following observations. First of all, the logit model seems to be the weakest link in this combination. The procedure to optimize the weight factor even excludes logit. This seems to be a bit exaggerated, but a relative low value is desirable. Furthermore, the weight factor for the linear regression is set a bit higher because it is built on all features, and subsequently is expected to represent more information. The resulting gain charts for the configuration with the weight factors determined by a domain expert are drawn in figure 4.

	equal	scaled	optimized	dom exp		all
β_1	$0.25/\gamma_1$	1	2.1061	1.5	γ_1	0.2131
β_2	$0.25/\gamma_2$	1	0	0.5	γ_2	0.2038
β_3	$0.25/\gamma_3$	1	2.3412	1	γ_3	0.2066
β_4	$0.25/\gamma_4$	1	0.1793	1	γ_4	0.3765

Table 1: Weight factors for example 1

The numerical performance measure used in the CoIL Challenge was the number of respondents found in the first 20 % of the customers sorted by the SC_n scores. The total number of respondents in the score set equals 238. A random mailing (no model used) results in a performance measure of circa 48 (47.6). So, every value greater than this one is a gain to random mailing and the maximum number is 238. The values for the four algorithms used stand-alone and the combinations with the four sets of weight factors can be found in table 2.

	lin	log	nn	fuz	equal	scaled	optim	dom exp
score 1	121	110	110	113	118	118	119	123
score 2	110	108	109	90	114	115	113	114
score 3	115	104	111	111	115	113	117	116
score 4	118	103	114	108	114	116	117	120
avg score	-	-	-	-	116	115	119	119

Table 2: Numerical performance measures example 1

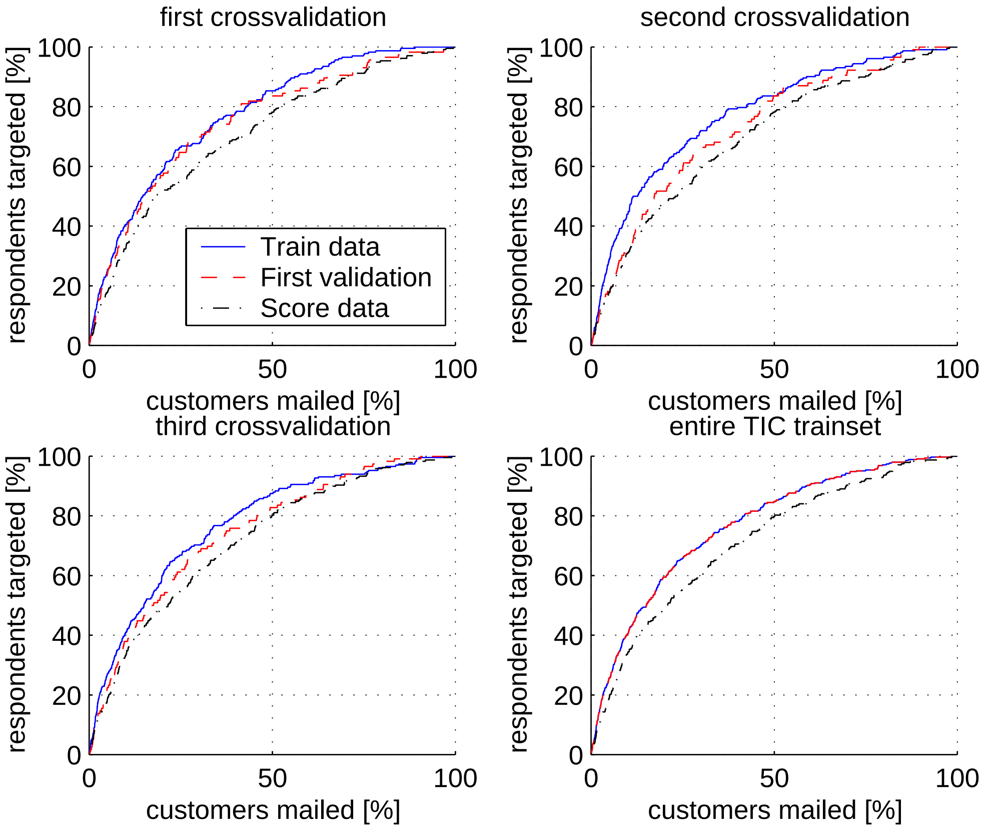

Figure 4: Gain charts example 1

Example 2

The second combination exists of two algorithms: the same logit configuration as in example 1 and a neural network with 6 neurons in the hidden layer on a feature subset based on correlation to the dependent variable. This second feature subset contains 11 features. The different values for the weight factors are listed in table 3. The dissentient choice of the domain expertise weights can be explained by the fact that this particular neural network performs really well on the data and logit does less, as can be seen in example 1. The combination works very well as can be observed by looking at the numerical performances which are listed in table 4 and the gain charts for the domain expertise configuration which are drawn in figure 5.

	equal	scaled	optimized	dom exp		all
β_1	$0.5/\gamma_1$	1	1.3897	0.5	γ_1	0.6456
β_2	$0.5/\gamma_2$	1	0.2929	1.5	γ_2	0.3544

Table 3: Weight factors for example 2

	log	nn	equal	scaled	optim	dom exp
score 1	110	103	105	107	115	104
score 2	108	98	106	106	110	104
score 3	114	95	106	108	114	106
score 4	103	121	122	115	123	125
avg score	-	-	108	108	115	108

Table 4: Numerical performance measures example 2

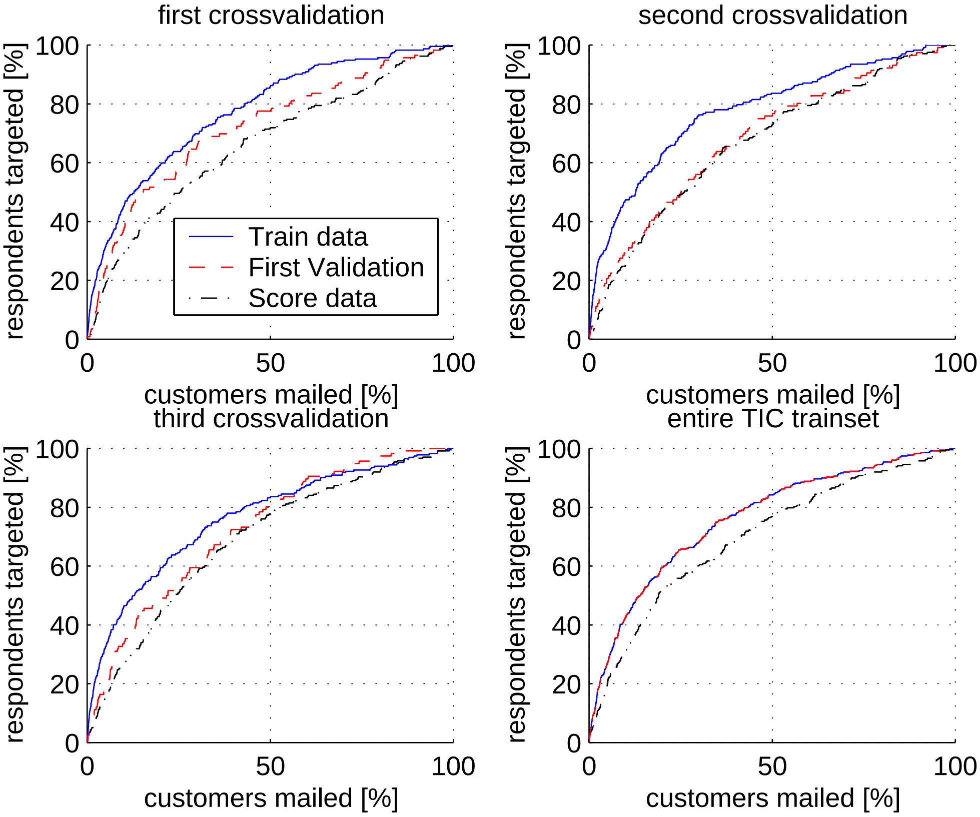

Figure 5: Gain charts example 2

5 Discussion

In this section, the results of the previous section are discussed and compared to the results of the contestants of the CoIL challenge 2000. From table 2, we can see that the model based on the entire train set correctly scores 120 in the first 20 % ranked prospects in the out-of-sample test. The model on the first cross validation does even better: 123 respondents. In figure 6 the results of the 43 participants are visualized. The best score is 121. Even the other models do very well (in the best 5% of the CoIL participants).

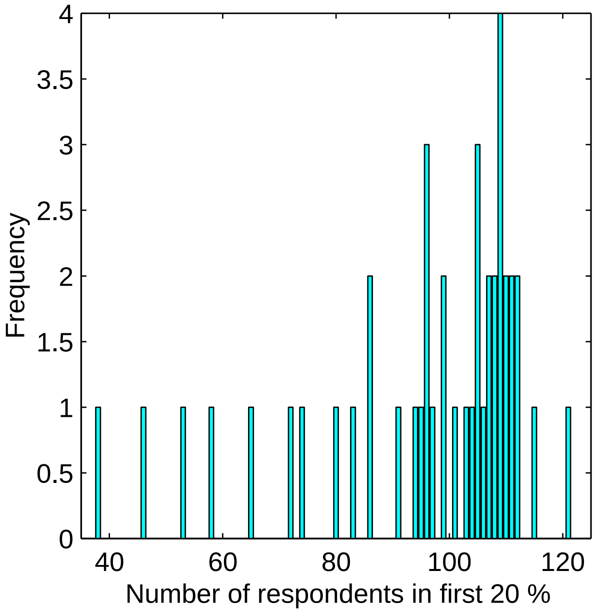

Figure 6: Results participants of CoIL 2000

In example 2 even a higher value is reached: 125 for the combination on the entire train set. Although the configuration with the neural network scored 121 stand-alone, the combination with logit adds another 4 successfully predicted respondents.

The models build on the three cross validation sets (”avg score”) do not have a better result than the combinations on the entire train set.

Although some configurations used in the two examples do very well applied stand-alone, there is a significant gain in using approach of combining target selection algorithms.

Every configuration, hence algorithm, introduces its own systematic errors because the assumptions underlying the algorithm do not hold to their full extend in the real world. Optimizing the weight factors using linear regression treats each systematic error as a random error in the addition sum (equation 1). Following this observation, optimizing the weights is more justified when a larger

number of algorithms are combined. Systematic errors imply that all values in a set are shifted in the same direction in the same amount - in unison. This is in contrast to random errors where each value fluctuates independently of the others. So, when it comes to optimizing the weight factors, a combination of more than a few algorithms with *different* assumptions is the best option.
The results subscribe this statement: the combinations of the models build on the three cross sets (”avg score”) do not outperform the combinations on the entire train set, because the same assumptions hold. On the other hand , the β weights of the optimized set in example 1 are closer to the weights set by the domain expert than in example 2. The domain expert obtained the best results, so in example 1 the weights are nearer to his weights, because more algorithms, hence different assumptions are used.

6 Conclusions and Recommendations

Despite the fact that many algorithms are developed for the purpose of target selection, no universal algorithm exist. The strength of the approach presented in this paper is that the structure and specific characteristics of each feature subset are maintained and scored individually. If proper action is taken to prevent over-training in building the algorithms, combining them will not introduce any kind of over-training. The quality of the train samples are equally important in preventing over-training. Spurious relations can be found, especially in case of a large number of algorithms in a combination.
By adequately combining the scores, results are achieved out-performing all participants in the CoIL challenge 2000. The best results are obtained by using the weights chosen by a domain expert. If such a person is not available, the best way seems to be optimizing the weights using a linear regression method with positive coefficients. A fairly large number of algorithms with different assumptions is needed for adequately optimized weights.

Further research has to be done to find out where the limits of combining algorithms are at. To put in other words, how sensitive this approach is to adding a large number of different combinations. Another question is how cross validation could better be integrated in this technique.

References

[1] J.R. Bult, *Target selection for direct marketing*, PhD thesis, Rijks Universiteit Groningen, The Netherlands, 1993

[2] W.H. Greene, *Econometric Analysis*, Macmillan Publishing Company New York, 1993

[3] 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, No. 1, 2001

[4] G.V. Kass, "An exploratoty technique for investigating large quantaties of categorical data", *Applied Statistics* vol. 29, No. 2, pp. 119-127, 1980

[5] M. Ben-Akiva and S.R. Lerman, *Discrete choice analysis: Theory and Application to Travel Demand*, MIT Press England, 1985 1996

[6] R. Babuska, *Introduction to Neural Networks*, Delft University of Technology, 2000

[7] L. Breiman, J.H. Friedman, R.A. Olshen and C.J. Stone, *Classification and Regression Trees*, Wadsworth: Belmont USA, 1984

[8] D.M. Hawkins and G.V. Kass, "Automatic Interaction Detection", *Topics in Applied Multivariate Analysis*, pp. 267-302, Cambridge Univ Press: Cambridge, 1982

[9] E.B. Moser, "Chaid-like Classification Trees", *Multivariate Statistics, course material*, www.stat.lsu.edu.faculty/moser/exst7037/exst7037.html, 2000

[10] J. Magidson, "Improved statistical techniques for response modeling-Progression beyond regression", *Journal of Direct Marketing* vol. 2 No. 4 pp 6-18, 1988

[11] J. N. Morgan and J. A. Sonquist, "Problems in the analysis of survey data, and a proposal", *Journal of the American Statistical Association*, vol. 58 pp. 415-434, 1963

[12] J. A. Sonquist, E. L. Baker and J. N. Morgan, *Searching for Structure*, Institute for Social Research, University of Michigan, Ann Arbor, 1973

[13] D. Shepard, *The New Direct Marketing*, 2^{nd} edition, Irwin Professional Marketing, 1995

[14] S. J. Long, *Regression Models for Categorical and Limited Dependent Variables*, Sage Publications, 1997.

[15] J. Zahavi and N. Levin, "Applying neural computing to target marketing", *Journal of Direct Marketing*, vol. 11 No. 4 pp. 77-93, 1997

[16] D. J. Finney, *Probit Analysis*, Cambridge University Press, UK, 1964

[17] N. Draper and H. Smith, *Applied regression analysis*, New York: John Wiley & Sons, 1966 (Revised ed., 1981)

[18] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Ca, USA: Morgan Kaufmann Publishing, 1993

[19] D. G. Morrison, "On the Interpretation of Discriminant Analysis", *Journal of Marketing Research*, pp. 156-163, 1969

[20] N. Levin and J. Zahavi, *Predictive modeling using segmentation*, http://www.urbanscience.com/ Predictive_Modeling_Using_Segmentation.pdf , 1999

[21] S. Quigley, "Regression analysis", *SST Help guide* http://emlab.berkely.edu/sst/regression.html

[22] T-S Lim, W-Y Loh and Y-S Shih, "A comparison of prediction accurancy, complexity and training time of thirty-three old and new classification algorithms", *Journal of Machine Learning*, vol. 40 pp. 203-229, 2000.

[23] V.G. Morwitz and D.C. Schmittlein, "Testing new direct marketing offerings: The interplay of management judgment and statistical models", *Management Science* vol. 44, No. 5, pp. 610-628, May 1998