Improving Direct Marketing Profitability with Neural Networks

Zaiyong Tang
Salem State University
Salem, MA 01970

ABSTRACT
Data mining in direct marketing aims at identifying the most promising customers to send targeted advertising. Traditionally, statistical models are used to make such a selection. The success of statistical models depends on the validity of certain assumptions about data distribution. Artificial intelligence inspired models, such as genetic algorithms and neural networks, do not need those assumptions. In this paper, we test neural networks with real-world direct marketing data. Neural networks are used for performance maximization at various mailing depth. Compared with statistical models, such as logistic regression and ordinary least squares regression, the neural network models provide more balanced outcome with respect to the two performance measures: the potential revenue and the churn likelihood of a customer. Given the overall objective of identifying the churners with the most revenue potential, neural network models outperform the statistical models by a significant margin.

General Terms
Direct marketing, linear regression, artificial neural networks, direct response modeling.

Keywords
Neural networks, data Mining, direct Marketing, profit modeling.

1. INTRODUCTION
Data mining aims to identify patterns or relationships that are of interest or value the data owners. With the speed of data creation today, it is not surprising that data mining techniques have attracted considerable interest in both business and academia. Lyman and Varian estimated that the current annual growth rate of unique data is between 1 and 2 exabytes, or "roughly 250 megabytes for very man, woman, and child on earth."[1] The idea of extracting information from these large masses of data is indeed appealing considering the commercial, industrial, and economic potentials.

Data mining in direct marketing seeks techniques that maximize returns from direct-mailing solicitations. Pollay and Mittal [2] studied the multiple dimensions of direct marketing advertising. Although consumer perception of direct marketing advertising has not always been enthusiastic, direct response marketing is widely used in the industry. The statistics from the Direct Marketing Association shows that an estimated $166.5 billion was spent on direct marketing in US in 2006. In 2007, direct response advertising accounted for more than half of all US advertising expenditures [3].

One of the key tasks in the direct marketing advertising is to identify the most promising individuals to solicit. Due to time and budgetary constraints, it is generally not feasible to target the entire customer segment. Thus direct marketing models are built to maximize potential returns by targeting certain groups of customers or potential customers. The identification of target audiences for specific marketing promotions involves detailed analyses of the customer database to seek out individuals most likely to respond and generate profits.

Various direct marketing models can be built using attributes characterizing potential responders to marketing promotions. Statistical techniques such as discriminant analysis, least squares regression and logistic models are commonly used [4]. Bult and Wansbeek used statistical regression for optimal selection of target mailing [5]. Haughton and Qulabi modeled direct marketing with CART and CHAID [6]. Zahavi and Levin applied neural networks for target marketing and compared performance of neural networks with statistical approaches [7, 8]. Ha, Cho, and MacLachlan used neural networks for response modeling [9]. Baesens et al. applied Bayesian neural networks to direct marketing [10]. Kaefer et al. deployed neural networks models to improve the timing of direct marketing activities [11]. Lee and Shih applied neural network models to identify profitable customers [12]. Torres, Hervás, and García used a hybrid approach that combines logistic regression and neural networks for classification problems [13]. While Zahavi and Levin showed that neural network did not do better than statistical methods, Bentz and Merunkay found that neural networks outperformed multinomial logistic regression [14].

Typically, developed models are used to score individuals in a customer file such that higher scores indicate greater mailing preference [15]. The model-obtained scores are then used to rank individuals, and the final mailing list determined through mailing-cost and budgetary considerations. Response models use discriminant analysis to classify individuals as responders and non-responders, with model scores pertaining to individuals' response likelihood. An alternate objective is to identify individuals with the highest response frequency in previous mailings, or those that have generated most revenue in earlier purchases. Here the dependent variable becomes continuous, and regression models are often used. When customer data contains information pertaining to profits/costs associated with individuals, an attractive modeling criterion is to identify individuals such that the overall profit from a mailing, considering promotional costs and purchase revenues, is maximized.

Given resource limitations, direct marketing models are used to target a fraction of individuals in the customer file. The proportion of the selected best individuals to be targeted is referred to as the mailing depth or depth-of-file. Suppose the budget allows mailing to 5000 customers out of a total of 20,000 in the customer database. Obviously, we want to select the most promising 5000 individuals. In this case, the best 25% of individuals, as ranked by the model, makes the 25% depth-of-file. Once a model is built, various depth-of-file mailing strategies can be deployed. Because the individuals are ranked in the customer file, the smaller the mailing depth, the larger the improvement over randomly selected customer list of the same size.

Although statistical techniques such as linear regression and logistic regression are commonly used in direct marketing analysis, those techniques have potential problems. For example, assumptions inherent in many commonly used statistical techniques may not be valid as the model building typically relies on data collected with low response rates. In this paper, we consider a direct incorporation of customer value together with the mailing depth in model development. We present a neural network based modeling approach that takes advantage of the robust, nonlinear modeling capability of neural networks. The main objective is to study the performance of neural network models in comparison to traditional statistical modes.

The following section discusses the performance analysis of direct marketing models. Section 3 gives a brief introduction to the neural network models used in our study. Section 4 presents the proposed direct marketing modeling approach. Experimental results are provided in Section 5, followed by a discussion of future research issues in Section 6 and conclusion in Section 7.

2. DECILE ANALYSIS

Given that direct marketing models are used to identify a subset of the total customers expected to maximize returns from a mailing solicitation, model performance is assessed at different mailing depths. Typically a decile analysis is used to examine model performance [16]. In a decile analysis, individuals are separated into 10 equal groups based on their ranking or respective model scores. In general, higher scores indicating better performance. Table 1 shows a typical decile analysis where the performance objective is profit maximization from a mailing. The first row, or top decile, indicates performance for the best 10% of individuals as identified by the model. The Cumulative Lifts at specific depths of file provide a measure of improvement over a random mailing, and is calculated as:

$$Cumulative\_Lift_{decile} = \frac{Cumulative\_average\_profit_{decile}}{Overall\_average\_profit} \times 100$$

Thus, in Table 1, a cumulative lift of 255 in the top decile indicates that the model in question is expected to provide a mailing profit that is 2.55 times the profit expected from a random mailing to 10% of the customers. Similarly, if 20% of the customers are to be mailed, the model is expected to perform 2.16 times better than a random mailing of 20% of the customers. The cumulative lift at the bottom decile is always 100 and corresponds to a mailing to the entire customer list. An ideal model should exhibit decreasing performance from the top through bottom deciles. As indicated in the table, the overall average customer profit is \$2.25. However, the average profit for the top 10% of the customer is \$5.75. The bottom 10% of the customer has an average profit of only \$0.84.

Table 1: Illustrative Decile Analysis

Decile	Number	Total Profit ($)	Average Profit ($)	Cumu. Average Profit ($)	Profit Lift (%)
1	500	2873.80	5.75	5.75	255
2	500	1990.32	3.98	4.86	216
3	500	1732.25	3.46	4.40	195
4	500	1231.55	2.46	3.91	174
5	500	885.30	1.77	3.49	155
6	500	627.10	1.25	3.11	138
7	500	513.35	1.03	2.82	125
8	500	504.18	1.01	2.59	115
9	500	480.62	0.96	2.41	107
10	500	420.78	0.84	2.25	100
Total	5000	11259.25	2.25		

3. NEURAL NETWORKS

Artificial neural networks are a broad class of computational models that have sparked wide interest in recent years [17, 18, and 19]. In contrast to conventional centralized, sequential processing, neural networks consist of massively connected simple processing units, which are analogous to the neurons in the biological brain. Through elementary local interactions (such as excitatory and inhibitory) among these simple processing units, sophisticated global behaviors emerge, resembling the high-level recognition process of humans [20, 21].

By virtue of their inherent parallel and distributed processing, neural networks have been shown to be able to perform tasks that are extremely difficult for conventional von Neumann machines, but are easy for humans. Neural networks have been used as an alternative approach to traditional optimization and statistical analysis, and have found successful applications in systems control, pattern recognition, classification, discriminant analysis, financial market, and forecasting [19].

Many neural network paradigms have been developed during the last two decades. One of the most widely used neural network models is the feedforward neural network, where neurons are arranged in layers [13]. Besides an input layer and an output layer, there are one or more hidden layers between the input and the output layer. Figure 1 gives a typical fully connected two-layer feedforward neural network (by convention, the input layer does not count) with N input nodes, H hidden nodes, and M output notes. It is common to refer the network as NxHxM network. The arrows represent the forward direction. Full connection means that each input node is connected to every hidden node, and each hidden node is connected to every output node. Note that it is possible to build neural network models with partial connections. Small networks (with small number of nodes and/or small number of connections) are generally preferred when the model needs to be able to generalize outside the sample data [22]. Input to the neural network are X = {xi | i = 1, 2, …, N} and output is Y = {yi | i = 1, 2, …, M}.

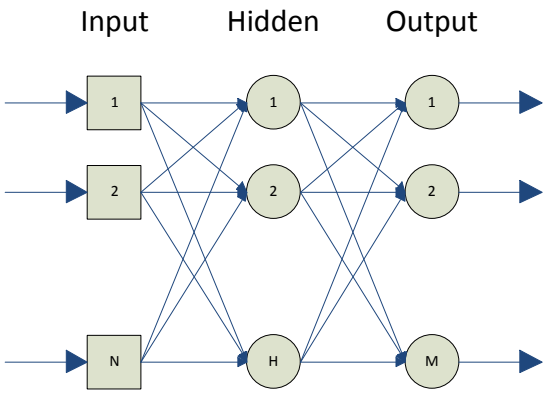

Fig 1. A Feedforward Network

A feedforward neural network is used by first training it with known examples (X, T), where X are the inputs and T are the target values. Training a neural net means modifying the weights on the links (connection strength) such that the network learns the underlying pattern(s) from the training examples. A widely used training algorithm for feedforward neural networks is known as the backpropagation algorithm. Backpropagation is essentially a gradient decent based algorithm that minimizes the error function, typically, the sum of squared differences of the network outputs and the target values.

$$E = \sum_i \sum_j (y_{ij} - t_{ij})^2 \text{ for i = 1, 2, …, P; j = 1, 2, …, M}$$

where P is the number of sample (x, t) pairs and M is the number of output nodes.

Output error is back propagated through the network, and the weights are modified to reduce the output error. When the error reaches a predetermined minimum we say the training is done. A trained neural network can be used to retrieve the input-output relationship of the training examples. More importantly, it can generalize from the limited training examples. In other words, a trained neural network can predict the target value given a new set of input data. For a complete coverage of the backpropagation training algorithm and many of its variations, the reader is referred to Fine [23]

4. DATA AND MODELING

A real world application is studied in this paper. The problem considered is that of a cellular-phone provider seeking to identify potential high-value churners so that they can be targeted with some appropriate intervention program. The specific objective is to identify high-value churners amongst new installs within the first year of service. Two dependent variables correspond to the two important measures of the objective: (1) a binary Churn variable indicating whether a customer churned (value 1) or not (value 0) within the first four months; and (2) a continuous variable measuring revenue ($) associated with the customer. The predictor variables considered pertain to standard measures used in the cellular industry. Four predicator variables used in this study are peak minutes-of-use, off-peak minutes-of-use, average charges, and payment information. The data were obtained after the usual variable transformation and reduction.

Cumulative lifts at the specified depths-of-file serve as a performance measure. As discussed in Section 1, cumulative lifts at specific depths of file provide a measure of improvement over a random mailing. For instance, a lift of 300 at the 10% depth of file indicates that the model in question is expected to provide a performance that is three times that of expected from a random mailing to 10% of the list.

Two cumulative lifts are used to gauge performance levels resulting from the two dependent variables. Churn-Lift at the desired decile shows the relative performance of the model in identifying churners. Revenue lift, denoted as $-Lift at the desired decile indicates the model performance in identifying high-value customers without regard for their churn likelihood. Note that a high Churn-Lift does not correspond to a high value of $-Lift. A model that does well with both performance measures is preferred. The maximization of the expected revenue that can be saved through identification of high-value churners is the overall modeling objective.

Churn-Lift and $-Lift are estimated at a specific decile d as follows: Consider R_d and C_d the cumulative total revenue and cumulative total number of churners respectively at the decile d, R the total revenue for the entire data, and C the total churners in the entire data. Then, if N denotes the overall total customers and N_d is the total customers up to the decile level d, the cumulative churn and revenue lifts are:

$$Churn - Lift = \frac{C_d / N_d}{C / N}$$

and

$$\$ - Lift = \frac{R_d / N_d}{R / N}$$

The expected revenue saved through identifying the churners up to the depth-of-file d is given by the product of average churn per customer and average revenue per customer.

$$\frac{C_d}{N_d} * \frac{R_d}{N_d} = (Churn - Lift * \$ - Lift)(\frac{R}{N} * \frac{C}{N})$$

The product of Churn-Lift and $-Lift value provides a measure for comparing the performance of models as it gives the cumulative lift on the expected revenue saved as:

$$(Churn - Lift * \$ - Lift) = (\frac{C_d}{N_d} * \frac{R_d}{N_d}) / (\frac{R}{N} * \frac{C}{N})$$

Feedforward neural networks are used to model the relationship between the independent variables and dependent variables (churn and revenue). In theory, a feedforward neural network with a single hidden layer is sufficient to approximate any continuous functions [24]. Empirical evidence shows that more than one hidden layer in the neural network models does not noticeably improve the performance. So we have used neural networks with 4X8X1 and 4X8X2 structure. That is, there are four input nodes (corresponding to the four input variables), eight hidden nodes, and one or two output nodes.

The numbers of input and output nodes depend on the data attributes, while the selection of the number of hidden nodes is often based on rule of thumb. Since we are using fully connected feedforward networks, the number of weights W depends on the number of hidden nodes. The general guideline in selecting the number of hidden nodes is to construct a neural network that is just large enough to solve the problem at hand. Not enough weights may render the model incapable of solving the problem, while too many weights tend to reduce the model's generalization ability [22].

After a few trial runs, the neural network training parameters are selected as follow: Number of training epochs = 1000, learning rate = 0.5, momentum = 0.7. Neural networks are initialized with random weights. Each set of experiment is carried out 10 times with different initial weights, and the average results are reported. A sample of 50,000 customer data was used for the modeling building and testing. This sample was divided into equal training and test sets of 25,000 observations each. The training set was used to build the models. No cross validation was used during training. All reported results are based on the test data.

A logistic regression model for Churn and an ordinary least squares regression model for revenue give us the baseline performances for the two objectives. While these models are expected to perform well on their respective single objectives, they may not provide effective solutions for the overall objective, i.e., maximization of the expected revenue that can be saved through targeted marketing to the high-value churners.

5. EXPERIMENT RESULTS

We tested the models at four different depths-of-file: 10%, 20%, 30% and 70%. Table 2 shows the Churn-Lift and $-Lift values from three neural network models. Model one uses the binary churn variable as training target with network structure 4x8x1. Revenue is not used in Model one. Model two is similarly constructed, but it uses the continuous revenue variable as the training target while the churn variable is omitted. Model three combines the two independent variables as the training target with a network structure of 4x8x2.

All three neural network models show significant improvement across various depth-of-files compared with the expected performance from random sampling. The result is encouraging considering that neural network models used are relatively simple. We have not conducted comprehensive search of optimal neural network structures. Not surprisingly, model one gives the largest churn-lift, as identifying the churners is the objective of this model. Model two aims to maximize the $-lift. The performance on Churn-Lift is not considered by the model, hence the poor performance results for Churn-Lift. When the two performance measures are combined, as in the case of model three, more balanced results are achieved, and the overall performance also improves.

Table 2. Neural network performance results

Neural Network		**10% depth**	**20% depth**	**30% depth**	**70% depth**
Model 1	**Churn-Lift**	365.5	343.9	291.1	138
	$-Lift	211.2	152.2	128.1	86.4
	Product of lifts	771.9	523.4	372.9	119.2
Model 2	**Churn-Lift**	106.1	101.9	95.2	86.8
	$-Lift	361.4	271.1	222.3	136.2
	Product of lifts	383.4	276.2	211.5	118.2
Model 3	**Churn-Lift**	253.0	185.2	149.9	93.6
	$-Lift	314.1	290.1	270.1	138.1
	Product of lifts	794.8	537.2	404.9	129.3

Table 3. Performance comparison: Neural network vs. Regression

Performance		**10% depth**	**20% depth**	**30% depth**	**70% depth**
Best NN model	**Churn-Lift**	253.0	185.2	149.9	93.6
	$-Lift	314.1	290.1	270.1	138.1
	Product of lifts	794.8	537.2	404.9	129.3
Logistic Regression	**Churn-Lift**	447.1	403.4	296.0	137.8
	$-Lift	111.8	72.6	57.4	66.7
	Product of lifts	499.8	292.7	170.0	91.9
OLS Regression	**Churn-Lift**	116.2	108.1	99.7	91.8
	$-Lift	360.5	271.7	223.2	136.2
	Product of lifts	418.8	293.7	222.5	125.1
Improvement over Logistic		**59.0%**	**83.5%**	**138.2%**	**40.7%**
Improvement over OLS		**89.8%**	**82.9%**	**82.0%**	**3.3%**

In terms of the overall performance measure: the product of lifts, Model 1 and 3 are significantly better than model 2. This indicates that high-revenue generating customers do not correspond to high churn rate. Model 1 suggests that churners may contribute to relative large revenue loss. Since Model 3 provides the highest overall performance, it should be the model of choice for this particular application. Table 3 gives performance comparisons between Model 3, our choice of neural network model, and traditional statistical approaches, namely, the logistic model and the least squares regression model.

Table 3 shows clearly the neural network model outperforms the logistic regression and OLS models. In particular, when the depth-of-file is limited to the top 30 percent, the neural network gives considerably better overall performance. Note that both of the regression models suffer skewed performance, as the logistic model overlooks the $-Lift while the OLS model overlooks the Churn-Lift. It is also noteworthy that the product of lifts generated by the neural network decreases in significant amount when the depth-of-file goes from 10 percent to 30 percent. However, the relative performance of the neural network model over the comparative models is still significantly better.

6. DISCUSSION

Feedforward neural networks are considered a general class of robust non-linear models. While linear models are widely used in real world applications, most real-world problems, nevertheless, exhibit non-linear relationship between the independent and dependent variables. Neural networks enable us to design nonlinear systems that are able to deal with complex problems without a priori knowledge of the input-output relationship. Because of their powerful modeling capability and relative ease of use, neural networks have found wide in various pattern recognition applications [23].

Linear regression models use linear functions to fit the data, based on the assumptions that the relationship between the dependent variable Y and independent variables X is linear; the values of Y are statistically independent of one another, and the distribution of possible values of Y for any X values is normal with equal variances. Those assumptions may not hold true for the all data sets. In contrast to the statistical models, neural networks make no such assumptions about the data; hence they can be applied to a wider range of problems. Furthermore, by changing the neural network structure and activation functions of the processing elements (nodes), we can use neural networks to approximate classification and regression models.

In the current application, we use neural networks to model the input-output relationship of the sample data. This input-output relationship is employed in the test data to "predict" the revenue potential and churn likelihood of a customer. The current neural network model does not directly incorporate the performance maximization at a given depth-of-file. Future research may consider modifying the standard neural network learning algorithm to explicitly seek performance maximization with specified mailing depth as an input. This will enable the decision maker to build optimal performance models geared towards specific depth-of-file requirements.

Building the best neural network for an application is still more of an art than a science. Zahavi and Levin [7] reported that neural networks did not outperform logistic regression. They suggested that two possible reasons for their results. One is that neural networks may be over fitting the training data. Another reason is that neural network models are typically built by trial and error approach. Further experiments exploring the use of other neural network models, such as modular neural networks, network with weight decay, and multiple objective models may lead to improved performance. More efficient neural network learning algorithms may also be used to improve the training efficiency. Techniques such as cross-validation can be used to increase the generalization ability of the trained neural network model.

7. CONCLUSION

We have applied one of the most popular neural network models, namely, the feedforward neural network, to performance maximization at desired mailing depths in direct marketing in cellular phone industry. Neural network based predictive model identifies the most promising individuals given a specified mailing depth. Compared with statistical models, such as logistic regression and ordinary least squares regression, the neural network models provide more balanced outcome regarding the two predicted measures, namely, the potential revenue and the churn likelihood of a customer. In terms of the overall objective, i.e., identifying the churners with the most revenue potential, neural networks models outperforms the statistical models by a significant margin. The performance of the neural network models is particularly well with low depth-of-file target levels.

8. ACKNOWLEDGMENTS

My thanks to Sid Bhattacharyya for providing the data and help with the direct response modeling and decile analysis.

9. REFERENCES

[1] Lyman, Peter and Hal R. Varian, 2000. "How Much Information?" Research report, School of Information Management and Systems, University of California at Berkeley.

[2] Pollay, R.W. and B. Mittal, 1993. "Here's the beef: factors, determinants and segments of consumer criticism of advertising", Journal of Marketing, 57. 99-114.

[3] DMA 2007 annual report: Working to keep every channel open and economically viable for all marketers. http://web.mac.com/asyracuse/Site/Corporate_Clips_files/annualreport.pdf. Retrieved March 22, 2008.

[4] Hand, D.J. 1981. Discrimination and Classification, John Wiley and Sons, New York, NY.

[5] Bult, J.R. and T.J. Wansbeek, 1995. Optimal selection for direct mail, Marketing Science, 14, 378-394.

[6] Haughton, D. and S. Oulabi. 1997. Direct marketing modeling with CART and CHAID, Journal of Direct Marketing, 11(4), 42-52.

[7] Zahavi, J., and Levin, N. 1997. Issues and problems in applying neural computing to target marketing. Journal of Direct Marketing, 11(4), 63–75.

[8] Zahavi, J. and Levin, N. 1997. Applying neural computing to target marketing, Journal of Direct Marketing, 11 (1), 5-22.

[9] Ha, K., Cho, S. and MacLachlan, D. 2005. Response models based on bagging neural networks. Journal of Interactive Marketing, 19(1). 17-30.

[10] Baesens, B., S. Viaene, D. van den Poel, J. Vanthienen, G. Dedene. 2002. Bayesian neural network learning for repeat purchase modelling in direct marketing. European Journal of Operations Research. 138(1) 191–211.

[11] Kaefer, Frederick, Heilman, Carrie M. and Ramenofsky, Samuel D. 2005. A Neural Network Application to Consumer Classification to Improve the Timing of Direct Marketing Activities. Computers and Operations Research, 32 (10), 2595-2615.

[12] Lee, Wan-I, Bih-Yaw Shih. 2009. Application of neural networks to recognize profitable customers for dental services marketing-a case of dental clinics in Taiwan. Expert System Applications. 36(1). 199-208.

[13] Torres, Mercedes, Cesar Hervás , Carlos García, 2009. Multinomial logistic regression and product unit neural network models: Application of a new hybrid methodology for solving a classification problem in the livestock sector, Expert Systems with Applications: An International Journal, 36(10),12225-12235.

[14] Bentz, Y. and Merunkay, D. 2000. "Neural Networks and the Multinomial Logit for Brand Choice Modeling: A Hybrid Approach", Journal of Forecasting, Vol. 19 (3), 177-200.

[15] Bhattacharyya, S. 1999. "Direct Marketing Performance Modeling using Genetic Algorithms", INFORMS Journal of Computing, 11(3). 248-257.

[16] David Shepard Associates, 2005. The New Direct Marketing: How to Implement a Profit-Driven Database Marketing Strategy, 2nd Edition, Irwin Publishing. 1995. 19(1), 17-30.

[17] Levine, D. S. and M. Aparicio, (editors), 1994. Neural Networks for Knowledge Representation and Inference, MIT press.

[18] Principe, J. C., N. R. Euliano and W. C. Lefebvre. 2000. Neural and Adaptive Systems: Fundamentals Through Simulations, John Wiley & Son, New York.

[19] Refenes, A. P. (editor), 1995. Neural networks in the capital markets, John Wiley & Sons, West Sussex, England,

[20] Rumelhart, D. E., James L. McClelland, and the PDP Research Group, 1986. Parallel Distributed Processing - Explorations in the microstructure of Cognition, Volume I: Foundations, The MIT Press.

[21] Rumelhart, D.E., G.E. Hinton and R.J. Williams, 1986. Learning Internal Representations by Error Propagation, in Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Volume I: Foundations, D.E. Rumelhart and J.L. McClelland (eds.), MIT Press, Cambridge, MA.

[22] Barrtlett, P. 1998. "The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network," IEEE Transactions on Information Theory, 44, 141-166.

[23] Fine, T. L. 1999. Feedforward Neural Network Methodology, Springer-Verlag, New York.

[24] Hornik, K.. 1991. Approximation capabilities of multiplayer feedforward networks. Neural Networks, 4, 251-257.