Customer Targeting: A Neural Network Approach Guided by Genetic Algorithms

YongSeog Kim[†], W. Nick Street[†], Gary J. Russell[‡], Filippo Menczer[†]
†: Management Sciences Department, ‡: Marketing Department
University of Iowa
Iowa City, IA 52242 USA
{yong-s-kim,nick-street,gary-j-russell,filippo-menczer}@uiowa.edu

Abstract

One of the key problems in database marketing is the identification and profiling of households who are most likely to be interested in a particular product or service. Principal component analysis (PCA) of customer background information followed by logistic regression analysis of response behavior is commonly used by database marketers. In this paper, we propose a new approach that uses artificial neural networks (ANN's) guided by genetic algorithms (GA's) to target households. We show that the resulting selection rule is more accurate and more parsimonious than the PCA/logit rule when the manager has a clear decision criterion. Under vague decision criteria, the new procedure loses its advantage in interpretability, but is still more accurate than PCA/logit in targeting households.

1 Introduction

Due to the growing interest in micro marketing, many firms devote considerable resources to identifying households that may be open to targeted marketing messages. The availability of data warehouses combining demographic, psychographic and behavioral information further encourages marketing managers to use database-based approaches to develop and implement marketing programs.

Database marketers use different tools, depending upon what is known about particular households. Routine mailings to existing customers are typically based upon the RFM (recency, frequency, monetary) approach that targets households using knowledge of the customer's purchase history (Schmid and Weber, 1998). Mailings to households with no prior relationship with the firm are based upon the analysis of the relationship between demographics and the response to a test mailing of a representative household sample. Given the large number of potential demographics available, data dimension reduction is an important factor in building a predictive model that is easy to interpret, cost effective, and generalizes well to unseen cases. Commonly, principal component analysis (PCA) of demographic information (Johnson and Wichern, 1992) is used to prepare new variables for this type of analysis. These new variables are then used as predictors in a logistic regression on the test mailing responses.

In this study, we propose a new approach to building predictive models for identifying prospective households. The new methodology combines genetic algorithms (GA's) for choosing predictive demographic variables with artificial neural networks (ANN's) for developing a model of consumer response. ANN's (Riedmiller, 1994; Sarle, 1994) and GA's (Goldberg, 1989; Yang and Honavar, 1998; Krishna and Murty, 1999) have been widely used in machine learning, pattern recognition, image analysis and data mining. In particular, ANN's have been recognized as a relatively new approach in finance and marketing applications such as stock market prediction (Saad et al., 1998; Pan et al., 1997), bankruptcy prediction (Wilson and Sharda, 1994), customer clustering (Gath and Geva, 1988; Ahalt et al., 1990) and market segmentation (Hruschka and Natter, 1999; Balakrishnan et al., 1996). In this work, we exploit the desirable characteristics of GA's and ANN's to achieve two principal goals of household targeting: model interpretability and predictive accuracy. Our approach is different from previous studies on direct marketing because of our consideration of multiple objectives (Ling and Li, 1998) and data reduction (Bhattacharyya, 2000).

Data reduction of demographic information is performed via feature selection in our approach. Feature selection is defined as the process of choosing a subset of the original predictive variables by eliminating features that are either redundant or possess little predictive information. If we extract as much information as possible from a given data set while using the smallest number of features, we can not only save a great amount of computing time and cost, but also build a model that generalizes better to households not in the test mailing. Feature selection can also significantly improve the comprehensibility of the resulting classifier models. Even a complicated model - such as a neural network - can be more easily understood if constructed only from a few variables. In database marketing applications, it is important for managers to understand the key drivers of consumer response. A predictive model that is essentially a "black box" is not useful for developing comprehensive marketing strategies.

In our work, a specifically designed GA, the Evolutionary Local Search Algorithm (ELSA), is used to search through the possible combinations of features. Two quality measurements – hit rate (which is maximized) and complexity (which is minimized) – are used to evaluate the quality of each feature subset. ELSA performs a local search in the space of feature subsets by evaluating genetic individuals based on both their quality measurements and on the number of similar individuals in the neighborhood in objective space. The bias of ELSA toward diversity makes it ideal for multi-objective optimization, giving the decision maker a clear picture of Pareto-optimal solutions from which to choose. Previous research has demonstrated the effectiveness of ELSA for feature selection in both supervised (Menczer et al., 2000a) and unsupervised (Kim et al., 2000) learning.

The input features selected by ELSA are used to train an artificial neural network that predicts "buy" or "not buy." Using information from households with an observed response, the ANN is able to learn the typical buying patterns of customers in the dataset. The trained ANN is tested on an evaluation set, and a proposed model is evaluated both on the hit rate and the complexity (number of features) of the solution. This process is repeated many times as the algorithm searches for a desirable balance between predictive accuracy and model complexity. The result is a highly accurate predictive model that uses only a subset of the original features, thus simplifying the model and reducing the risk of overfitting. Because the algorithm identifies variables with no predictive value, it also provides useful information on reducing future data collection costs.

This paper is organized as follows. In Section 2, we explain ELSA in detail. In Section 3, we describe the structure of the ELSA/ANN model, and review the feature subset selection procedure. In Section 4, we present experimental results of both the ELSA/ANN and PCA/logit model algorithms. Using test-mailing responses on insurance policies, we show that there is a trade-off between model interpretability and predictive accuracy. In particular, we obtain both high model interpretability and high predictive accuracy only when the firm is specific about the way model forecasts will be used to select households in future mailings. In contrast, interpretability must be sacrificed to preserve predictive accuracy when the firm is vague about its selection rule. Section 5 concludes the paper and provides suggestions about future research directions.

2 Evolutionary Local Selection Algorithm (ELSA)

2.1 Local Selection and Algorithm Details

ELSA springs from algorithms originally motivated by artificial life models of adaptive agents in ecological environments (Menczer and Belew, 1996). Modeling reproduction in evolving populations of realistic organisms requires that selection, like any other agent process, be locally mediated by the environment in which the agents are situated.

In a standard evolutionary algorithm,[1] an agent is selected for reproduction based on how its fitness compares to that of other agents. In ELSA, an agent (candidate solution) may die, reproduce, or neither based on an endogenous energy level that fluctuates via interactions with the environment. The energy level is compared against a constant selection threshold for reproduction. By relying on such *local* selection, ELSA reduces the communication

[1]We use the terms genetic algorithms (GAs) and evolutionary algorithms (EAs) interchangeably.

```
initialize population of agents, each with energy θ/2
while there are alive agents and for T iterations
   for each energy source c
      for each v (0 .. 1)
         E^c_envt(v) ← 2vE^c_tot
      endfor
   endfor
   for each agent a
      a' ← mutate(clone(a))
      for each energy source c
         v ← Fitness(a', c)
         ΔE ← min(v, E^c_envt(v))
         E^c_envt(v) ← E^c_envt(v) − ΔE
         E_a ← E_a + ΔE
      endfor
      E_a ← E_a − E_cost
      if (E_a > θ)
         insert a' into population
         E_a' ← E_a/2
         E_a ← E_a − E_a'
      else if (E_a < 0)
         remove a from population
      endif
   endfor
endwhile
```

Figure 1: ELSA pseudo-code. In each iteration, the environment is replenished and then each living agent executes the main loop. In sequential implementations, the main loop calls agents in random order to prevent sampling effects. We stop the algorithm after T iterations.

among agents to a minimum. The competition and consequent selective pressure is driven by the environment (Menczer et al., 2000b). There are no direct comparisons with other agents and the search is biased directly by the environment. Further, the local selection scheme naturally enforces the diversity of the population, making ELSA appropriate for multi-objective optimization problems.

We now briefly describe the ELSA implementation for the feature selection problem. A more extensive discussion of the algorithm and its application to Pareto optimization problems can be found elsewhere (Menczer et al., 2000a; Menczer et al., 2000b). Figure 1 outlines the ELSA algorithm at a high level of abstraction.

2.2 Agents, Mutation and Selection

Each agent in the population is first initialized with some random solution and an initial reservoir of *energy*. The representation of an agent consists of D bits, with each of the bits indicating whether the corresponding feature is selected or not (1 if a feature is selected, 0 otherwise).

Mutation is the main operator used to explore the search space. The mutation operator randomly selects one bit of each agent and mutates it. At each iteration an agent produces a mutated clone to be evaluated. Each agent competes for a scare resource, energy, based on its multi-dimensional fitness and the proximity of other agents in the solution space. In the selection part of the algorithm, each agent compares its current energy level with a fixed threshold θ. If its energy is higher than θ, the agent reproduces: the mutated clone that was just evaluated becomes part of the population, with half of its parent's energy. When an agent runs out of energy, it is killed.

The population size is maintained dynamically over the iterations and is determined by the carrying capacity of the environment depending on the costs incurred by any action, and on the replenishment of resources both described below (Menczer et al., 2000b). The population size is also independent of the reproduction threshold, θ, which only affects the energy stored by the population at steady-state.

2.3 Energy Allocation and Replenishment

In each iteration of the algorithm, an agent explores a candidate solution (the mutated clone). The agent collects ΔE from the environment and is taxed with a constant cost E_{cost} ($E_{cost} < \theta$) for this "action." The net energy intake of an agent is determined by its fitness. This is a function of how well the candidate solution performs with respect to the criteria being optimized. But the energy also depends on the state of the environment. The environment corresponds to the set of possible values for each of the criteria being optimized.[2] We imagine an energy source for each criterion, divided into bins corresponding to its values. So, for criterion fitness F_c and bin value v, the environment keeps track of the energy $E^c_{envt}(v)$ corresponding to the value $F_c = v$. Further, the environment keeps a count of the number of agents $P_c(v)$ having $F_c = v$. The energy corresponding to an action (alternative solution) a for criterion F_c is given by

$$Fitness(a, c) = \frac{F_c(a)}{P_c(F_c(a))}. \tag{1}$$

[2]Continuous objective functions are discretized.

Candidate solutions receive energy only inasmuch as the environment has sufficient resources; if these are depleted, no benefits are available until the environmental resources are replenished. Thus an agent is rewarded with energy for its high fitness values, but also has an interest in finding unpopulated niches in objective space, where more energy is available. The result is a natural bias toward diverse solutions in the population.

When the environment is replenished with energy, each criterion c is allocated an equal share of energy:

$$E_{tot}^{c} = \frac{p_{max} E_{cost}}{C} \tag{2}$$

where C is the number of criteria considered. This energy is apportioned in linear proportion to the values of each fitness criterion, so as to bias the population toward more promising areas in objective space. Note that the total replenishment energy that enters the system at each iteration is $p_{max} \cdot E_{cost}$, which is independent of the population size p but proportional to the parameter p_{max}. This way we can maintain p below p_{max} on average, because in each iteration the total energy that leaves the system, $p \cdot E_{cost}$, cannot be larger than the replenishment energy.

3 ELSA/ANN Model for Customer Targeting

Our predictive model of household buying behavior is a hybrid of the ELSA and ANN procedures. In this approach, ELSA identifies relevant consumer descriptors that are used by the ANN to forecast consumer choice. We focus here on the structure of the approach and the criteria used to select an appropriate predictive model.

3.1 Structure of ELSA/ANN Model

The model setup is shown in Figure 2. ELSA searches for a set of feature subsets and passes them to an ANN. The ANN extracts predictive information from each subset and learns the patterns using a randomly selected 2/3 of the training data. Once an ANN learns the data patterns, the trained ANN is evaluated on the remaining 1/3 of the training data, and returns two evaluation metrics, $F_{accuracy}$ and $F_{complexity}$, to ELSA. It is important to note that in both the learning and evaluation procedures, the ANN uses only the selected features.

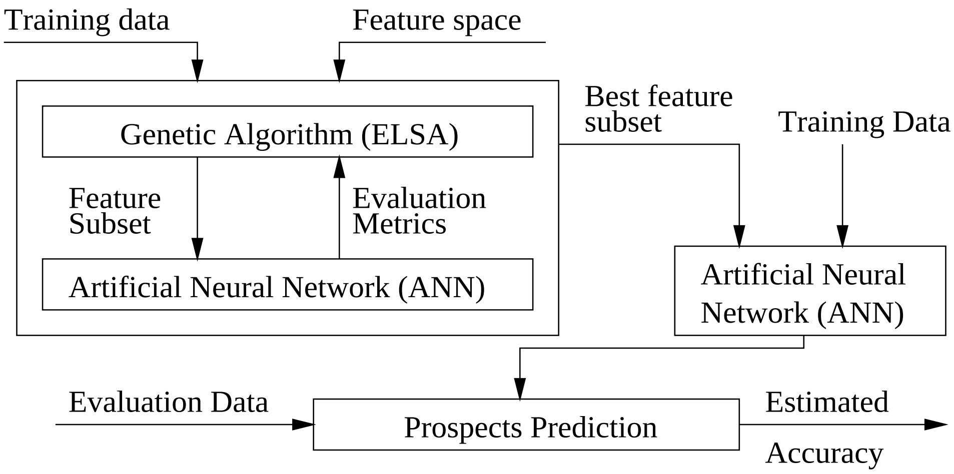

Figure 2: The structure of ELSA/ANN model. ELSA searches for a good subset of features and passes them to an ANN. The ANN calculates the "goodness" of each subset and returns two evaluation metrics to ELSA.

Based on the returned metric values, ELSA biases its search direction to maximize the two objectives. This routine continues until the maximum number of iterations is attained. All evaluated solutions over the generations are saved into an off-line solution set without comparison to previous solutions. In this way, high-quality solutions are maintained without affecting the evolutionary process.

Among all the evaluated subsets, we choose for further evaluation the set of candidates that satisfy a minimum hit rate threshold. With these chosen candidates, we start a more rigorous selection procedure, 10-fold cross validation. In this procedure, the training data is divided into 10 non-overlapping groups. We train an ANN using the first nine groups of training data and test the trained ANN on the remaining group. We repeat this procedure until each of the 10 groups is used as a test set once. We then take the average of the accuracy measurements over the 10 folds and call it an *intermediate* accuracy. We repeat the 10-fold cross validation procedure five times and average the five intermediate accuracy estimates. We call this the *estimated* accuracy through the following sections.

For evaluation purposes, we select a single "best" solution in terms of both estimated accuracy and complexity. We subjectively decided to pick a solution with the minimal number of features at the marginal accuracy level.[3] Once we decide on the best solution, we train the ANN using all the training data with the selected features only. The trained model is then used to rank the potential customers (the records in the evaluation set) in descending order by the probability of buying RV insurance, as predicted by the ANN. We finally select the top $x\%$ of the prospects and calculate the *actual* accuracy of our model using the actual choices of the evaluation set households.

3.2 Evaluation Metrics

We define two heuristic evaluation criteria, $F_{accuracy}$ and $F_{complexity}$, to evaluate selected feature subsets. Each objective, after being normalized into 25 intervals to allocate energy, is maximized by ELSA.

$\boldsymbol{F_{accuracy}}$**:** The purpose of this objective is to favor feature sets with a higher hit rate. Each ANN takes a selected set of features to learn data patterns and predicts which potential customers will actually purchase the product. In our application, we define two different measures, $F^1_{accuracy}$ and $F^2_{accuracy}$ for two different experiments. Experiment 1 assumes that the managers can specify in advance the rule to be used in select households for mailings. We select the top 20% of potential customers in descending order of the probability of purchasing the product and compute the ratio of the number of actual customers, AC, out of the chosen prospects, TC. We calculate $F^1_{accuracy}$ as follows:

$$F^1_{accuracy} = \frac{1}{Z^1_{accuracy}} \frac{AC}{TC} \tag{3}$$

where $Z^1_{accuracy}$ is an empirically derived constant to normalize $F^1_{accuracy}$.

In Experiment 2, we consider a generalization of Experiment 1. We first divide the range of customer selection percentages into 50 intervals with equal width (2%) and measure accuracy at the first m intervals only.[4] At each interval $i \leq m$, we select the

[3] If other objective values are equal, we prefer to choose a solution with small variance.

[4] This could be justified in terms of costs to handle the chosen prospects and the expected accuracy gain. As we select more prospects, the expected accuracy gain will go down. If the marginal revenue from an additional prospect is much greater than the marginal cost, however, we could sacrifice the expected accuracy gain. Information on mailing cost and customer value was not available in this study.

top $(2 \cdot i)\%$ of potential customers in descending order of the probability of purchasing the product and compute the ratio of the number of actual customers, AC_i, out of the total number of actual customers in the evaluation data, Tot. We multiply the width of interval and sum those values to get the area under the lift curve over m intervals. Finally we divide it by m to get our final metric, $F^2_{accuracy}$. We formulate it as follows:

$$F^2_{accuracy} = \frac{1}{Z^2_{accuracy}} \frac{1}{m} \sum_{i=1}^{m} \frac{AC_i}{Tot} \cdot 2 \tag{4}$$

where $Tot = 238$, $m = 25$ and $Z^2_{accuracy}$ is an empirically derived constant to normalize $F^2_{accuracy}$.

$\boldsymbol{F_{complexity}}$: This objective is aimed at finding parsimonious solutions by minimizing the number of selected features as follows:

$$F_{complexity} = 1 - \frac{d-1}{D-1} \tag{5}$$

where d and D represent the dimensionality of the selected feature set and of the full feature set, respectively. Note that at least one feature must be used. Other things being equal, we expect that lower complexity will lead to easier interpretability of solutions as well as better generalization.

4 Application

The new ELSA/ANN methodology is applied to the prediction of households interested in purchasing an insurance policy for recreational vehicles. To benchmark the new procedure, we contrast the performance of the ELSA/ANN methodology to an industry-standard logit approach that summarizes household background information using principal components analysis. We evaluate the ELSA/ANN approach using two experiments. In Experiment 1, we inform the algorithm of the way in which the predictive model will be used by managers to select households for a direct mail solicitation. In Experiment 2, we leave this information vague. We show that the new approach provides improvements in forecasting accuracy, but that model complexity is contingent on the amount of information about the managerial decision rule.

4.1 Data Description

The data are taken from a solicitation of 9,822 European households to buy insurance for a recreational vehicle. These data, taken from the CoIL 2000 forecasting competition (Kim and Street, 2000), provide an opportunity to assess the properties of the ELSA/ANN procedure in a customer prospecting application.[5] In our analysis, we use two separate datasets: a training set with 5822 households and an evaluation set with 4000 households. The training data is used to calibrate the model and to estimate the hit rate expected in the evaluation set. Of the 5822 prospects in the training dataset, 348 purchased RV insurance, resulting in a hit rate of $348/5822 = 5.97\%$. From the manager's perspective, this is the hit rate that would be obtained if solicitations were sent out randomly to consumers in the firm's database.

The evaluation data is used to validate the predictive models. Our predictive model is designed to return the top $x\%$ of customers in the evaluation dataset judged to be most likely to buy RV insurance. The model's predictive accuracy is examined by computing the observed hit rate among the selected households. It is important to understand that only information in the training dataset is used in developing the model. Data in the evaluation dataset is used exclusively for forecasting.

In addition to the observed RV insurance policy choices, each household's record also contains 93 additional variables, containing information on both socio-demographic characteristics (variables 1-51) and ownership of various types of insurance policies (variables 52-93). Details are provided in Table 1. The socio-demographic data are based upon postal code information. That is, all customers living in areas with the same postal code have the same socio-demographic attributes. The insurance firm in this study scales most socio-demographic variables on a 10-point ordinal scale (indicating the relative likelihood that the

[5] We use a dataset on consumer responses to a solicitation for "caravan" insurance policies. A "caravan" is similar to a recreational vehicle in the United States. For more information about the CoIL competition and the CoIL datasets, refer to the Web site http://www.dcs.napier.ac.uk/coil/challenge/.

socio-demographic trait is found in a particular postal code area). This 10-point ordinal scaling includes variables denoted as "proportions" in Table 1. For the purposes of this study, all these variables were regarded as continuous. The psychographic segment assignments (attributes 4-13), however, are household-specific and are coded into ten binary variables.

In our subsequent discussion, the word feature refers to one of the 93 variables listed in Table 1. For example, the binary variable that determines whether or not a household falls into the "successful hedonist" segment is a single feature. Accordingly, in the feature selection step of the ELSA/ANN model, the algorithm can choose to use any possible subset of the 93 variables in developing the predictive model.

4.2 Experiment 1

In Experiment 1, we maximize the hit rate when choosing the top 20% potential customers as in Kim and Street (2000). We select the top 20% of customers in the evaluation dataset using the model created by the ELSA/ANN procedure. The actual choices of these households provide a measure of the hit rate. For comparison purposes, we implemented a principal component analysis (PCA) of the household background characteristics followed by a logistic regression of the insurance policy choice data. PCA is analogous to our feature selection procedure to reduce data dimension. The logistic regression is, in fact, an example of a very simple ANN. The PCA/logit approach is commonly used by industry consultants in developing household selection rules.

We also implemented an intermediate model, ELSA/logit, for comparison purposes. The ELSA/logit model is different from ELSA/ANN in the sense that it uses only one hidden node.[6] We use the same criterion to select the final solution of ELSA/logit as in ELSA/ANN. The motivation behind the ELSA/logit model is the decomposition of the accuracy gain of ELSA/ANN into two sources: feature selection and response function approximation. The difference in results between PCA/logit and ELSA/logit can be attributed to characteristics of feature selection, while the difference in results between ELSA/logit and ELSA/ANN's can be attributed to the greater flexibility of ANN in approximating the response model.

[6] ELSA/ANN models use $\sqrt{node_{in}}$ where $node_{in}$ represents the number of input nodes. See the appendix for more details on ANN's.

Feature ID	Feature Description
1	Number of houses owned by residents
2	Average size of households
3	Average age of residents
4-13	Psychographic segment: successful hedonists, driven growers, average family, career loners, living well, cruising seniors, retired and religious, family with grown ups, conservative families, or farmers
14-17	Proportion of residents with Catholic, Protestant, others and no religion
18-21	Proportion of residents of married, living together, other relation, and singles
22-23	Proportion of households without children and with children
24-26	Proportion of residents with high, medium, and lower education level
27	Proportion of residents in high status
28-32	Proportion of residents who are entrepreneur, farmer, middle management, skilled laborers, and unskilled laborers
33-37	Proportion of residents in social class A, B1, B2, C, and D
38-39	Proportion of residents who rented home and owned home
40-42	Proportion of residents who have 1, 2, and no car
43-44	Proportion of residents with national and private health service
45-50	Proportion of residents whose income level is < \$30,000, \$30,000-\$45,000, \$45,000-\$75,000, \$75,000-\$123,000, >\$123,000, and average
51	Proportion of residents in purchasing power class
52-72	Scaled contribution to various types of insurance policies such as private third party, third party firms, third party agriculture, car, van, motorcycle/scooter, truck, trailer, tractor, agricultural M/C, moped, life, private accident, family accidents, disability, fire, surfboard, boat, bicycle, property, social security
73-93	Scaled number of households holding insurance policies for the same categories as in scaled contribution attributes

Table 1: Household background characteristics

Before discussing results, we first briefly summarize our implementation of the PCA/logit benchmark model in Figure 3. We selected 22 principal components. This is the minimum required to explain more than 90% of the variance in the training set. In order to get the estimated hit rate, we implement 10-fold cross validation on the training set as shown in Figure 4. In the cross validation procedure, the scores of the PC's are estimated using different portions of the data each time to get the estimated hit rate.

```
Apply PCA on training data D_train
Determine appropriate number of PCs, n
Reduce the dimensionality of D_train using n PCs, creating D'_train
Perform logistic regression on D'_train and save β̂_i and α̂ where i = 1,···,n.
Reduce the dimensionality of evaluation data D_eval using n PCs, creating D'_eval
Calculate p(not buy) for each record in D'_eval using
        p = exp(α̂+Σ_{i=1}^{n} β̂_i·PC_i) / (1+exp(α̂+Σ_{i=1}^{n} β̂_i·PC_i))
Select 20% of records, R, with lowest p
for each selected record r
  if r is an actual customer
    counter = counter + 1
  endif
endfor
Hit_rate = counter/R
```

Figure 3: The implementation procedure of PCA/logit model.

```
Divide training data D_train into 10 equal-sized subsets
for each subsets D_train[i], i = 1,···,10
  Define D_train[i]^c = D_train - D_train[i]
  Apply PCA on D_train[i]^c, and select n PCs
  Reduce the dimensionality of D_train[i]^c using n PCs
  Do logistic regression on reduced D_train[i]^c
  Reduce the dimensionality of D_train[i] using n PC scores
  Calculate p(not buy) using the formula in Figure 3
  Pick 20% of records, R[i], with lowest p
  for each selected record r
    if r is an actual customer
      counter[i] = counter[i] + 1
    endif
  endfor
endfor
Hit_rate = Σ_{i=1}^{10} counter[i] / Σ_{i=1}^{10} R[i]
```

Figure 4: The implementation procedure of cross-validation for PCA/logit model. We used the same number of PCs, $n = 22$, as we did in Figure 3.

We set the values for ELSA parameters in the ELSA/ANN and ELSA/logit models as follows: $\Pr(mutation) = 1.0$, $p_{max} = 1{,}000$, $E_{cost} = 0.2$, $\theta = 0.3$, and $T = 2{,}000$. We select the single solution which has the highest expected hit rate among those solutions that have fewer than 10 features selected in both models. We evaluated each model on the evaluation set. Our results are summarized in Table 2.

Model (# Features)	Training set	Evaluation set	
	Hit Rate ± s.d	# Correct	Hit Rate
PCA/logit (22)	12.83% ± 0.498%	109	13.63%
ELSA/logit (6)	15.73% ± 0.203%	115	14.38%
ELSA/ANN (7)	15.92% ± 0.146%	120	15.00%

Table 2: Results of Experiment 1. The hit rates from the three different models are shown as percentages with standard deviation. The column marked "# Correct" shows the number of actual customers who are included in the chosen top 20%. The number in parenthesis represents the number of selected features except for the PCA/logit model, where it represents the number of PCs selected.

In terms of the actual hit rate, all three models work very well. Even the model with lowest actual hit rate (PCA/logit) is 2.3 times better than the hit rate expected by mailing to these households at random (5.97%). The model generated by the ELSA/ANN procedure returns the highest actual hit rate. As noted earlier, the difference in actual hit rate between PCA/logit and ELSA/logit provides an estimate of the accuracy gain that comes from the ELSA feature selection procedure. The difference in actual hit rate between ELSA/logit and ELSA/ANN provides an estimate of the accuracy gain that comes from the additional flexibility that ANN provides in approximating the true response function. In this application, both aspects of the ELSA/ANN procedure contribute equally to the improved accuracy of the model.

Judging the interpretability of a model is necessarily subjective. An advantage of the ELSA/ANN approach is that predictive features are clearly highlighted. In contrast, the PCA/logit model uses all of the features in constructing the principal component scores. We show the seven features that the ELSA/ANN procedure selected in Table 3.

Feature Type	Selected Features
Demographic features	"Average Family" psychographic segment
Behavioral features	Amount of contribution to third party policy, car policy, moped policy and fire policy, and number of households holding third party policies and social security policies

Table 3: Selected features by ELSA/ANN in Experiment 1.

With the exception of the "Average Family" psychographic segment, all other features are reports of the insurance buying behavior of the household's postal code area. The feature reporting car insurance makes considerable sense, given the fact that the firm is soliciting households to buy insurance for recreational vehicles. Further evaluation shows that prospects with at least two insured autos are the most likely RV purchasers. Moped policy ownership is justified by the fact that many people carry their mopeds or bicycles on the back of RVs. Those two features are selected again by the ELSA/logit model.[7] Using this type of information, we are able to build a potentially valuable profile of likely customers (Kim and Street, 2000).

In general, the results are in line with marketing science work on customer segmentation, which shows that information about current purchase behavior is most predictive of future choices (Rossi et al., 1996). The fact that the ELSA/ANN model used only seven features for customer prediction also implies that the firm could reduce data collection and storage costs considerably. This is possible through reduced storage requirements ($86/93 \approx 92.5\%$), and the reduced labor and data transmission costs.

We also compare the three models in terms of lift curves.[8] Figure 5 shows the cumulative hit rate over the top $2 \leq x \leq 100$ % prospects. Clearly, our ELSA/ANN model is the best when the firm selects the top 20% of prospects for a direct mail solicitation. However, the performance of ELSA/ANN and ELSA/logit over all targeting percentages was worse than that of PCA/logit. This occurs because our solution is specifically designed to optimize the hit rate when managers select the top 20% of prospects. In contrast, the PCA/logit model is estimated without any knowledge of how model forecasts will be used in decision-making. This observation motivated a second experiment in which we attempt to improve the performance of ELSA/ANN model over a greater range of decision rules.

[7] The other four features selected by the ELSA/logit model are: contribution to bicycle and fire policy, and number of trailer and lorry policies.

[8] Lift is defined as the percentage of all buyers in the database who are in the group selected for a direct mail solicitation. Under random sampling, the lift curve is a 45-degree line starting at the origin of the graph.

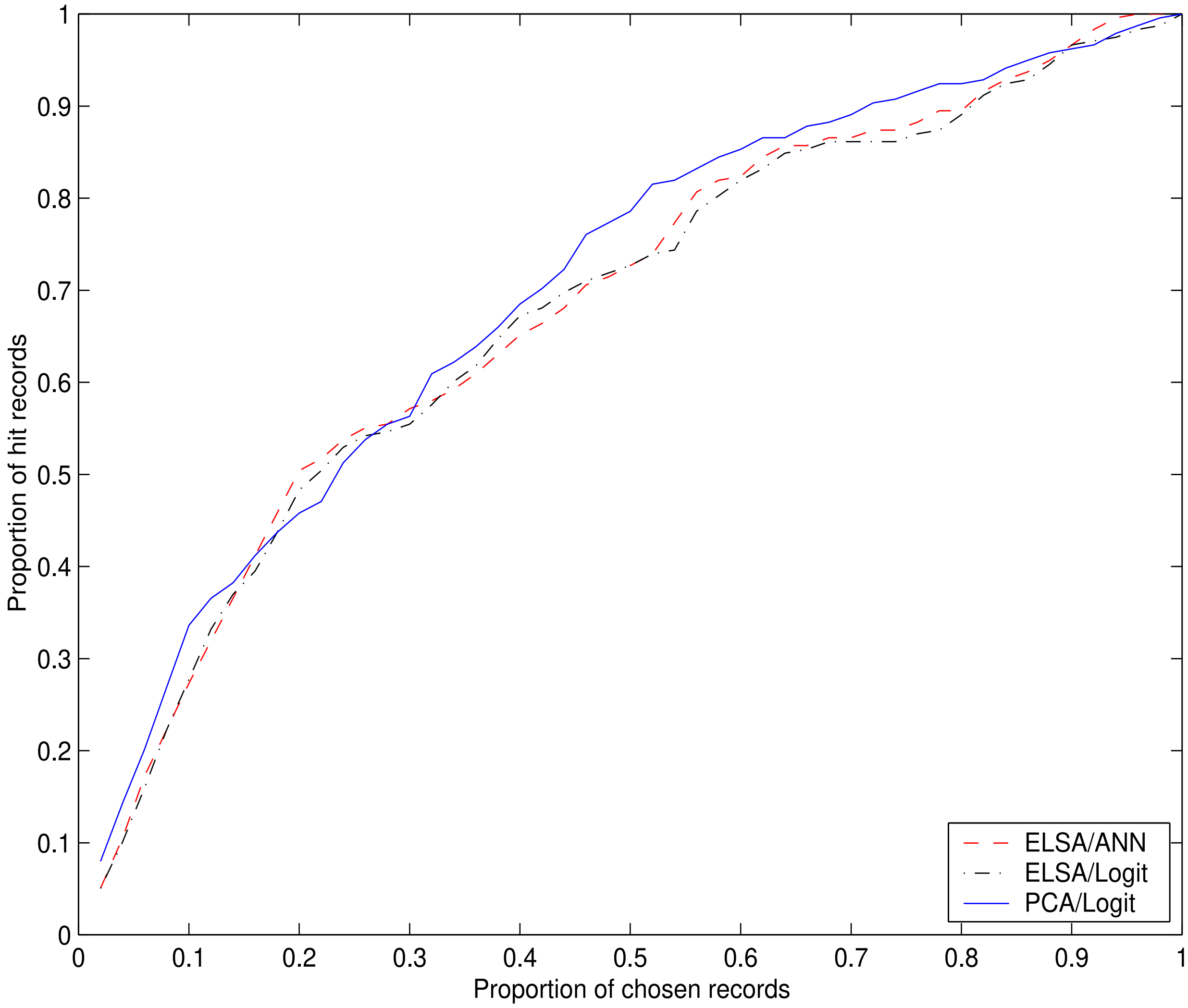

Figure 5: Lift curves of three models that maximize the hit rate when targeting the top 20% of prospects.

4.3 Experiment 2

In this experiment, we search for the best solution that maximizes the accuracy defined in a more global sense. The algorithm is designed to maximize the area under the lift curve, up to the top 50% of potential customers. Logically, the best solution from Experiment 1 is not necessarily the best solution in the more generalized environment of Experiment 2. In fact, our results are consistent with this observation. We also implemented the PCA/logit and the ELSA/logit model again for comparison purposes. We first show the generalized procedure of PCA/logit to get the estimated accuracy in Figure 6.

```
Apply PCA on training data D_train
Determine appropriate number of PCs, n
Reduce the dimensionality of D_train using n PCs, creating D'_train
Perform logistic regression on D'_train and save β̂_i and α̂ where i = 1, ···, n.
Reduce the dimensionality of evaluation data D_eval using n PCs, creating D'_eval
Calculate p(not buy) for each record in D'_eval using
        p = exp(α̂ + Σ_{i=1}^n β̂_i·PC_i) / (1 + exp(α̂ + Σ_{i=1}^n β̂_i·PC_i))
for each  i = 1 to int_num
   x = int_width · i
   Select x% records with lowest p
   for each  selected record r
      if  r is an actual customer
         counter = counter + 1
      endif
   endfor
   Hit_rate = counter/Tot
   Accuracy = Accuracy + Hit_rate * int_width
endfor
Accuracy = Accuracy/int_num
```

Figure 6: The generalized implementation of PCA/logit model. We use $n = 22$ (as in Experiment 1), $int_{num} = 25$, $int_{width} = 2$, and $Tot = 238$.

The ELSA/ANN and ELSA/logit models are adjusted to maximize the overall area under the lift curve over the same intervals as in PCA/logit. Because this new experiment is computationally much more expensive, we take a slightly different approach to choose the final solutions of ELSA/ANN and ELSA/logit. We used 2-fold cross validation estimates of all solutions and set the values of the ELSA parameters identically with the previous experiment except $p_{max} = 200$ and $T = 500$. Based on accuracy estimates, we choose a solution that has the highest estimated accuracy with less than half of original features in both models. We evaluate three models on the evaluation set and summarize results in Table 4 and in Figure 7.

Model (# Features)	% of Selected									
	5	10	15	20	25	30	35	40	45	50
PCA/logit (22)	20.06	20.06	16.04	13.63	12.44	11.20	10.81	10.22	9.87	9.38
ELSA/logit (46)	23.04	18.09	15.56	13.79	12.13	12.04	10.97	10.54	10.03	9.53
ELSA/ANN (44)	19.58	17.55	16.40	14.42	13.13	11.96	10.97	10.40	9.98	9.64

Table 4: Summary of Experiment 2. The hit rates of three different models are shown over the top 50% of prospects.

Table 4 shows that the ELSA/ANN model has higher hit rates than PCA/logit over the solicitation range between 15% and 50% of total households. In particular, ELSA/ANN is best when choosing 15%, 20%, 25% and 50% of the targeting points, and tied for the best at 30%, 35% and 45%. The overall performance of ELSA/logit is better than that of PCA/logit. We attribute this to the fact that both models benefit from the ELSA feature selection methodology.

The lift curves in Figure 7 show that the ELSA/ANN has much improved global characteristics relative to Experiment 1. We, however, note that there are significant costs associated with this improved performance. First, the hit rate of ELSA/ANN at the 20% solicitation rate is now lower than in Experiment 1 (14.42% versus 15.00%). Second, it is no longer clear which aspects of the ELSA/ANN model are responsible for the improved global performance. Note that the rank order of ELSA/logit and ELSA/ANN shows no consistent pattern across the various solicitation percentages. Third, the well-established parsimony and interpretability of the models selected by ELSA/ANN in Experiment 1 is largely lost in Experiment 2. We attribute this partially to the fact that different selection points may have related but different optimal subsets of features. Correlation among features seems to contribute to the loss of parsimony. For instance, a particular variable related to insurance policy ownership that is part of the optimal subset at a 20% selection rate could easily be replaced by a different, correlated feature at 30%. It should be noted that the ELSA/ANN model is superior to PCA/logit model in the sense that ELSA/ANN works with feature subsets, while PCA/logit always requires the whole feature set to construct PC's.

These aspects of the solution provide strong evidence that there exists a key trade-off in building a predictive model. By focusing on a specific decision scenario (as in Experiment

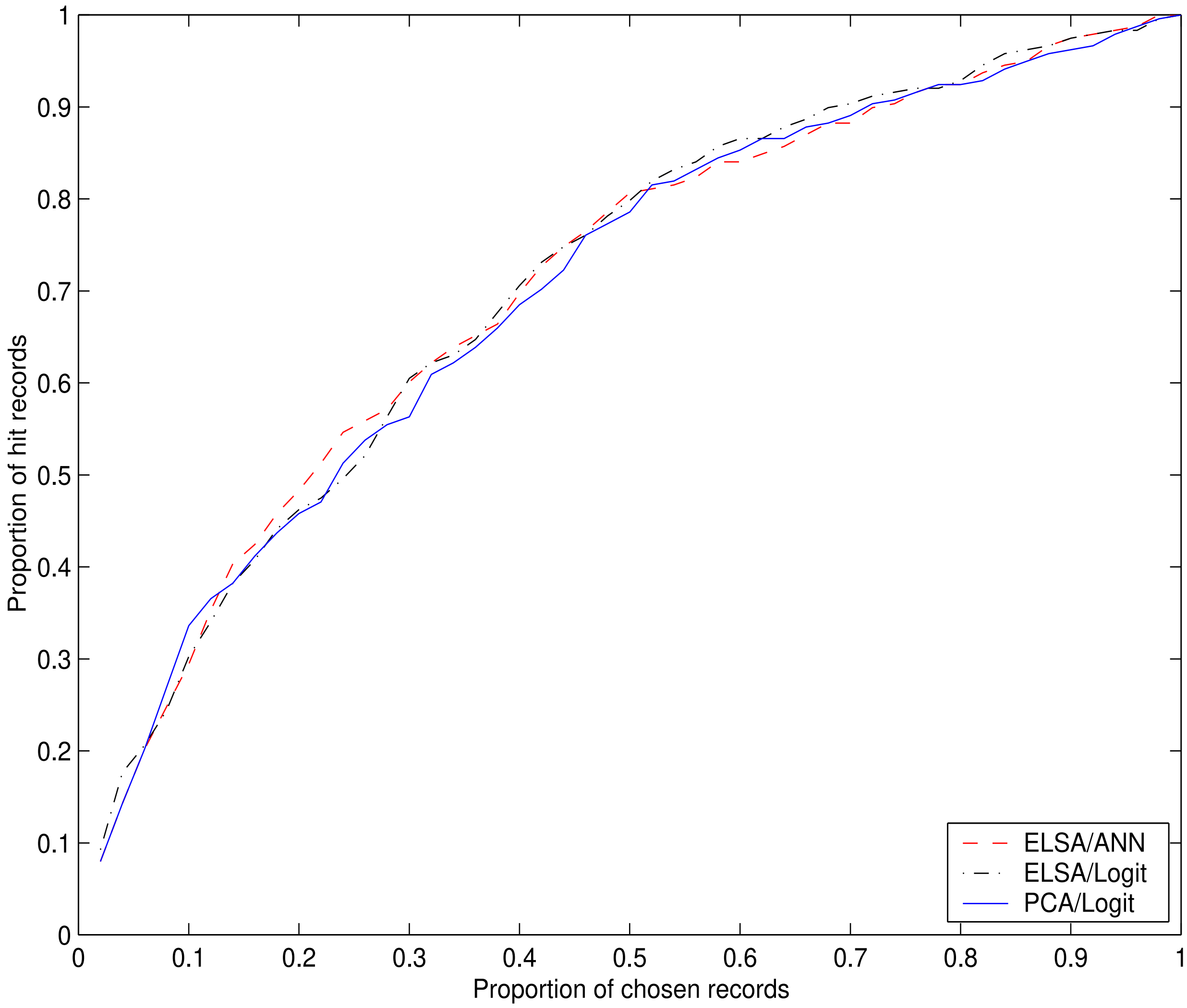

Figure 7: Lift curves of three models that maximize the area under lift curve when targeting upto top 50% of prospects. In practice, we optimize over the first 25 intervals which have the same width, 2%, to approximate the area under the lift curve.

1), we are able to construct a procedure that is parsimonious and has superior predictive performance. When the decision scenario is more ambiguous (as in Experiment 2), we can improve predictive performance over a broad range, but sacrifice model interpretability.

5 Conclusion

In this paper, we presented a novel approach for customer targeting in database marketing. We used an evolutionary algorithm, ELSA, to search for possible combinations of features and an artificial neural network (ANN) to score customers. When the decision rule was precise, the overall performance of ELSA/ANN was superior to the industry standard PCA/logit model both in terms of accuracy and in terms of interpretability. However, this superiority in interpretability is confined to specific decision conditions defined during model development and calibration. Under a more general decision scenario, ELSA/ANN yielded a more accurate model over a broad selection percentage range at the cost of increasing the number of predictive features in the specification.

One of the clear strengths of the ELSA/ANN approach is its ability to construct predictive models that reflect the direct marketer's decision process. Unlike a standard statistical approach like PC/logit, the ELSA/ANN procedure can be easily modified to take into account different objectives. With information of campaign costs and profit per additional actual customer, a direct marketer could use ELSA/ANN to choose the best selection point where expected total revenue is maximized. In this way, it would be possible to determine the type of decision rule that the marketer should adopt, both in terms of solicitation percentage as well as predictive rule. Because all mailing lists do not all have the same potential for the marketer, this approach would allow a predictive model and solicitation-mailing rule to be customized as the firm's database changes.

Our work provides additional evidence that there exists strong dependencies between model specification and managerial decision-making. When managers are clear about how a model will be used, the analyst can construct a highly specialized model that does better than general approaches (such as PC/logit). When managers are vague, a less parsimonious model can be constructed which does better under some region of the decision space. The

ELSA/ANN approach provides a new tool in which these trade-offs can be understood in the context of direct mail marketing applications.

Acknowledgments

The authors wish to thank Peter van der Putten and Maarten van Someren for making the CoIL data available for this paper. This work is partially supported by NSF grant IIS-99-96044.

References

Ahalt, S., Krishnamurthy, A., Chen, P., and Melton, D. (1990). Competitive learning algorithms for vector quantization. *Neural Networks*, 3:277–290.

Balakrishnan, P., Cooper, M., Jacob, V., and Lewis, P. (1996). Comparative performance of the FSCL neural net and K-means algorithm for market segmentation. *European Journal of Operation Research*, 93(10):346–357.

Bhattacharyya, S. (2000). Evolutionary algorithms in data mining: Multi-objective performance modeling for direct marketing. In *Proc. 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining (KDD-00)*, pages 465–473.

Gath, I. and Geva, A. (1988). Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):773–781.

Goldberg, D. (1989). *Genetic algorithms in search, optimization and machine learning.* MA: Addison-Wesley, New York.

Hruschka, H. and Natter, M. (1999). Comparing performance of feedforward neural nets and K-means for market segmentation. *European Journal of Operational Research*, 114:346–353.

Johnson, R. and Wichern, D. (1992). *Applied multivariate statistical analysis*. Prentice Hall, New Jersey, 3 edition.

Kim, Y. and Street, W. (2000). Coil challenge 2000: Choosing and explaining likely caravan insurance customers. Technical Report 2000-09, Sentient Machine Research and Leiden Institute of Advanced Computer Science. http://www.wi.leidenuniv.nl/~putten/library/cc2000/.

Kim, Y., Street, W., and Menczer, F. (2000). Feature selection in unsupervised learning via evolutionary search. In *Proc. 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining (KDD-00)*, pages 365–369.

Krishna, K. and Murty, M. (1999). Genetic K-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics - PartB: Cybernetics*, 29(3):433–439.

Ling, C. and Li, C. (1998). Data mining for direct marketing: Problems and solutions. In *Proc. 4th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining (KDD-98)*, pages 73–79.

Menczer, F. and Belew, R. (1996). Latent energy environments. In Belew, R. and Mitchell, M., editors, *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Addison Wesley, Reading, MA.

Menczer, F., Degeratu, M., and Street, W. (2000a). Efficient and scalable pareto optimization by evolutionary local selection algorithms. *Evolutionary Computation*, 8(2):223–247.

Menczer, F., Street, W., and Degeratu, M. (2000b). Evolving heterogeneous neural agents by local selection. In Honavar, V., Patel, M., and Balakrishnan, K., editors, *Advances in the Evolutionary Synthesis of Neural Systems*. MIT Press, Cambridge, MA.

Pan, Z., Liu, X., and Mejabi, O. (1997). A neural-fuzzy system for forecasting. *Journal of Computational Intelligence in Finance*, 5(1):7–15.

Riedmiller, M. (1994). Advanced supervised learning in multi-layer perceptrons - from backpropagation to adaptive learning algorithms. *International Journal of Computer Standards and Interfaces*, 16(5):265–278.

Rossi, P., McCulloch, R., and Allenby, G. (1996). The value of household information in target marketing. *Marketing Science*, 15(3):321–340.

Saad, E., Prokhorov, D., and Wunsch, D. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks*, 9(6):1456–1470.

Sarle, W. (1994). Neural networks and statistical models. In *Proc. 19th Annual SAS Users Group International Conference*, pages 1538–1550. SAS Institute.

Schmid, J. and Weber, A. (1998). *Desktop Database Marketing.* NTC Business Books.

Wilson, R. and Sharda, R. (1994). Bankruptcy prediction using neural networks. *Decision Support Systems*, 11(3):545–557.

Yang, J. and Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems and their Applciations*, 13(2):44–49.

Appendix

In this appendix, we briefly explain our neural network model and describe the learning procedure. Our neural network model is a multi-layer neural network consisting of a number of neurons (nodes) which are connected by weighted links. We specifically build it with three layers, an input layer, a hidden layer, and an output layer. We also note that the time complexity grows geometrically with the number of hidden nodes. We empirically set the number of hidden nodes as $\sqrt{node_{in}}$ where $node_{in}$ represents the number of input nodes. We represent our neural network model as follows:

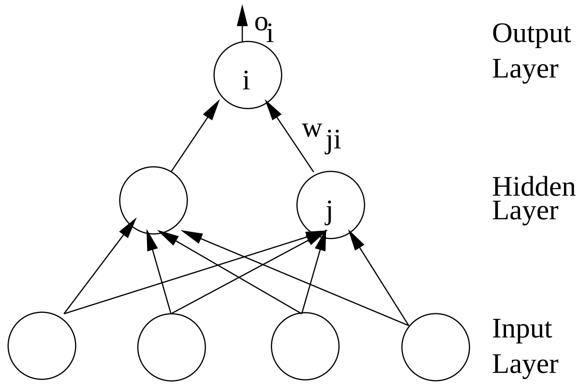

Figure 8: The structure of neural network model. It consists of three layers; input layer, hidden layer, and output layer. There is only one output node for concept learning and $\sqrt{node_{in}}$ nodes in the hidden layer.

As shown in Figure 8, each node in the input layer receives a corresponding feature from the input data, and passes it via weighted connections to neurons in the hidden layer. Each neuron i computes its activation level o_i using an activation function, the sigmoid logistic function

$$o_i = f(net_i) = \frac{1}{1 + e^{-net_i}} \tag{A-1}$$

where net_i is defined as follows:

$$net_i = \sum_{j \in pred(i)} o_j w_{ji}. \tag{A-2}$$

In the above equation, $pred(i)$ and w_{ji} denotes the set of predecessors of unit i and the connection weight from unit j to unit i respectively.

Learning in neural network is done by adjusting network weights in order to map input to output through examples in the training data set, N. Each example n consists of feature valuess, $\overline{x}_n$, and its corresponding class label t_n. When an example with $\overline{x}_n$ is presented to the network, the distance between the target t_n and the actual output vector o_d is measured as follows:

$$E = \frac{1}{2}\sum_{n \in N}(t_d - o_d)^2. \tag{A-3}$$

Fulfilling the learning goal now is equivalent to finding a minimum of E. The weights in the network are changed along a search direction $\delta(t)$, driving the weights in the direction of the estimated minimum:

$$w(t+1) = w(t) + \eta * \delta(t) \tag{A-4}$$

where the learning rate η determines the step size of weight changes and the negative gradient $-\frac{\partial E}{\partial w}$ is used for the search direction $\delta(t)$.

By propagating the error back from the output layer towards the input layer and applying the chain rule repeatedly, the backpropagation algorithm computes $\frac{\partial E}{\partial w_{ji}}$ for each weight in the network as follows:

$$\begin{aligned} \frac{\partial E}{\partial w_{ji}} &= \frac{\partial E}{\partial o_i}\frac{\partial o_i}{w_{ji}} \\ &= \frac{\partial E}{\partial o_i}\frac{\partial o_i}{\partial net_i}\frac{\partial net_i}{\partial w_{ji}} \\ &= \frac{\partial E}{\partial o_i} f'(net_i) o_j. \end{aligned} \tag{A-5}$$

Based on whether i is an output unit or not, the value of $\frac{\partial E}{\partial o_i}$ is computed as follows:

- Case 1: i is an output unit:

$$\frac{\partial E}{\partial o_i} = \frac{1}{2}\frac{\partial (t_i - o_i)^2}{\partial o_i} = -(t_i - o_i). \tag{A-6}$$

- Case 2: i is not an output unit:

$$\begin{aligned}
\frac{\partial E}{\partial o_i} &= \sum_{k \in upper(i)} \frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial o_i} \\
&= \sum_{k \in upper(i)} \frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial net_k}\frac{\partial net_k}{\partial o_i} \\
&= \sum_{k \in upper(i)} \frac{\partial E}{\partial o_k} f'(net_k) w_{ik} \qquad \text{(A-7)}
\end{aligned}$$

where $upper(i)$ denotes the set of all units k in upper layers and the gradient information is passed down from the output layer to input layer successively. Once the gradient information is known, the weight update is computed as follows:

$$\triangle w_{ji}(t) = -\eta * \frac{\partial E}{\partial w_{ji}}(t). \tag{A-8}$$