Data Mining Problems and Solutions for Response Modeling in CRM

Cho, Sungzoon · Shin, Hyunjung · Yu, Enzhe · Ha, Kyoungnam · MacLachlan, L. Douglas

Abstract

This paper presents three data mining problems that are often encountered in building a response model. They are robust modeling, variable selection and data selection. Respective algorithmic solutions are given. They are bagging based ensemble, genetic algorithm based wrapper approach and nearest neighbor-based data selection in that order. A real world data set from Direct Marketing Educational Foundation, or DMEF4, is used to show their effectiveness. Proposed methods were found to solve the problems in a practical way.

Keywords : Response Modeling, CRM, Data Mining, Stability, Ensemble, Variable Selection, Data Selection, Nearest Neighbor, Genetic Algorithm, Diversity, Accuracy

1. Introduction

Response modeling is concerned with computing the likelihood of a customer to respond to a marketing campaign. Lack of mental models for decision making leaves theoretical models unattainable. Statistical models are often developed from a historic data set which contains customers' past purchasing behavior and demographic information. The marketers can then use the model to compute the likelihood values or scores to reduce the number of customers whom they target in actual campaign. As far as the number of the responding customers is reasonably similar, the effectiveness of the campaign can be raised. Conventional statistical methods include logistic regression and multi-normal regression. Recently, however, machine learning based data mining models such as neural networks and support vector machines have been increasingly employed with significantly better accuracies. There are several data mining related issues in response modeling. They include instability of complex nonlinear models, many potential predictor variables to consider, and many data in the database. Each of these issues is not easy to tackle. Left unsolved, however, they result in poor performance. In this work, we present practical remedies for these problems : bagging ensemble, GA wrapper variable selection and Neighborhood Property based pattern selection, respectively. Experimental results involve a real world data set from Direct Marketing Educational Foundation(DMEF). Four remedies are applied and shown to have the desired effect.

This paper comprises four sections. The next three sections cover each issue, its remedy and empirical results : instability, variable selection and data selection, in that order and a conclusion is presented.

2. Model Instability

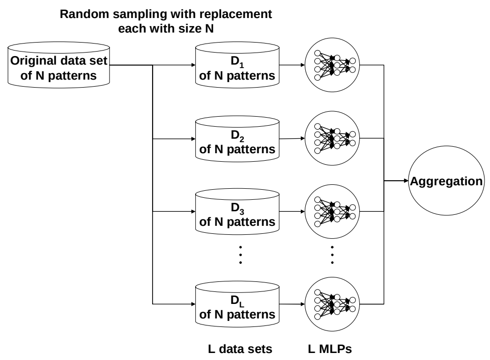
Traditionally, a linear statistical method such as logistic regression has been used to model response based on a test of a random sample of customers from the complete list. In order to overcome the limitations of logistic regression, other approaches such as ridge regression, stochastic RFM response models and hazard function models have been proposed recently. Neural networks, a class of non-linear models that mimic brain function, have been employed in marketing because no a priori knowledge or assumption about the error distribution is required[19]. It has been shown in one instance that neural network models improved the response rate up to 95% in direct marketing[2]. In another application, bank customers' response was predicted using a neural network[8], yielding superior results. A neural network was also shown to outperform multinomial logistic regression[1].

Cho, Sungzoon : Professor, Industrial Engineering, Seoul National University
Shin, Hyunjung : Researcher, Friedrich Miescher Laboratory
Yu, Enzhe : Consultant, Samsung Data Systems, Inc.
Ha, Kyoungnam : Ph. D. Candidate, Texas A & M University
MacLachlan, L. Douglas : Professor, University of Washington Business School

However, there is a tricky process involved in using neural networks or any complex nonlinear classifiers such as decision tree. An overly complex neural network is said to have a large variance; performance of the network varies greatly over different data sets from an identical population distribution. Simple models such as logistic regression would not have such a problem. When models have a large discrepancy between the true target and the expectation of the model output over different data sets, the models are said to have a large bias. Both bias and variance create classification error. A complex model has a large variance and a small bias while a simple model has a large bias and a small variance. One can typically improve one type of error at the expense of the other, thus the "bias-variance" dilemma[5]. It is a difficult task to determine the "optimal" complexity of a neural network given a finite training data set. What is usually practiced is a model selection process, a tedious, time-consuming trial-and-error search for the optimal complexity. Numerous research papers have been written about ways to find the right complexity, but it appears impossible to find an easy-to-understand and easy-to-follow procedure.

One way to alleviate the instability of a single neural network is to combine multiple networks. One of the most popular methods is bagging, or bootstrap aggregating, proposed in late nineties[3]. The bagging process begins with bootstrapping the training data(Step 1 in Figure 1). Given a data set of N patterns(i.e., customer records), a bootstrap sample is constructed by randomly sampling N patterns with replacement. Due to replacement, some patterns are picked more than once while some other patterns are not picked at all. It is estimated that a bootstrap sample contains about 67% of the original data set patterns.

Next, each bootstrap data set is used to train a neural network(Step 2 in Figure 1). Then one has L neural networks. The outputs of L neural networks are all potentially different. The final model output is computed by averaging L outputs for a regression problem or by a majority voting for a classification problem(Step 3 in Figure 1). It is obvious that bagging training takes at least L times the duration of a single neural network training, plus the bootstrap time. One positive aspect of bagging in terms of computation time is that the training of L neural networks can be done in parallel, if sufficient hardware and/or software systems are available.



Step 1. Generate L bootstrap samples

$\{B_i\},\ i=1,2,\cdots,L$

: A set of bootstrapping replicantes of original data set B

$\{f_i\},\ i=1,2,\cdots,L$

: A set of Networks in the ensemble

Step 2. Training L different neural networks

$f_i(x,y),\ i=1,2,\cdots,L,\ (x,y)\in B_i$

: Train each learner based on the bootstrap sample

Step 3. Aggregating L outputs

$$f_{Bag}(x)=\sum_{i=1}^{L}\alpha_i f_i(x)$$

: Aggregate the ouputs of learners based on weigh α_i or majority v oting

Figure 1. Conceptual model of bagging with L MLPs, each trained with N patterns

Bagging reduces variance[3] or model variability over different data sets from a given distribution, without increasing bias, which results in a reduced overall generalization error and an improved stability. Although the method is quite simple, it was found to be effective in many applications. The other advantage of using bagging is related to model selection. Since bagging transforms a group of over-fitted networks into a better-than-perfectly-fitted network, the tedious time-consuming model selection is no longer necessary. This could even offset the computational overhead introduced by bagging that involves training L neural networks.

DMEF4 is a public-domain data set provided by the Direct Marketing Educational Foundation(www.the-dma.org/dmef/). The data set is from an upscale gift business that mails general and specialized catalogs to its customers several times a year. There are two time periods, "base time period" from

December 1971 through June 1992, and "later time period" from September 1992 through December 1992. Every customer in the "later time period" received at least one catalog in early autumn of 1992. The fact that the data set is old does not matter since we focus on comparing the relative performance of different approaches. Now we are to build a response model for period October 1992 through December 1992 time period. That data set consists of 101,532 customers or records and 91 predictor variables or columns. The response rate is 9.4%. Since the experiments were repeated 30 times, the number of data or customers had to be reduced. Instead of randomly selecting 20% of customers, we chose to select those 20% of customers who are "important" in terms of their recent purchase. Recency was implemented using "weighted dollar amount spent" defined as

$$\begin{aligned}\text{Weighted Dollar} \;=\; & 0.4 \times \text{Dollars of this year}\\ & + 0.3\times \text{Dollars of last year}\\ & + 0.1\times \text{Dollars of 2 years ago}\\ & + 0.1\times \text{Dollars of 3 years ago}\\ & + 0.1\times \text{Dollars of 4 years ago}\end{aligned}$$

The particular values of weights were arbitrarily determined. However, a particular choice of the values would not alter the outcome of the experiments. The reduced data set now has 20,300 "important" customers with 18.9% response rate. We randomly chose 90% of these data for training. The remaining 10% or 2,030 records were set aside for test. The response rate of the test data set was 17.73%. The same data set was used in the following sections.

Figure 2 displays ROC charts for worst case out of 20 repetitions. The curve was formed by connecting adjacent points, each of which corresponds to a(false positive, false negative) pair obtained from a particular threshold value among nine, ranging from 0.1 through 0.9. A total of four models were built: the proposed bagging MLP(BMLP), a single MLP(SMLP), logistic regression with balanced data set (BLR) and logistic regression with unbalanced data set(UBLR). Three observations can be made. First, the bagging MLP(BMLP) did best. That was exactly what we set out to demonstrate by using BMLP in the first place: building a model that is good as well as stable in fit. Second, the single MLP did worst. Third, the BLR and UBLR did not seem to differ much in terms of false positive and false negative.

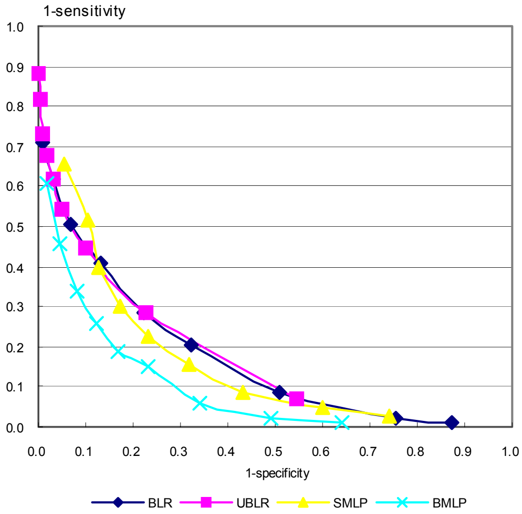


Figure 2. Worst case ROC curves for four models : BLR, UBLR, SMLP and BMLP

3. Variable Selection

A second issue involves which variable to choose in response modeling. A customer related data set usually contains hundreds of features or variables, many of which are irrelevant and heavily correlated with others. Without prescreening or feature selection, they tend to deteriorate performance of the model, as well as increase the model training time. Feature subset selection can be formulated as an optimization problem which involves searching the space of possible features to identify a subset that is optimum or near-optimal with respect to performance measures such as accuracy. Various ways to perform feature subset selection exist[16]. They can be classified into two categories based on its relation with base learner: filter and wrapper. The filter approach usually chooses features independently of base learner by human experts or statistical methods such as principal component analysis(PCA). Generally, It is computationally more efficient than a wrapper approach, but its major drawback is that an optimal selection of features may not be independent of the inductive and representational biases of the learning algorithm that is used to construct the classifier. The wrapper approach, on the other hand, evaluates candidate feature subsets by training a base learner such as a neural network or support vector machine with a given training data set using each feature subset under consideration and then testing it against a separate validation data set. Since the number of feature subsets is usually very large, this scheme is feasible only if training is relatively fast[16].

Furthermore, since wrapper approach usually involves search, choosing a proper way of searching is also an important issue. A simple exhaustive search is computationally infeasible in practice. Therefore, suboptimal yet practical search method is desired. The well known stepwise regression is a wrapper approach based on a greedy search. It is computationally efficient yet achieves suboptimal solution. Since more computing power is available now, randomized or probabilistic search such as genetic algorithm(GA) is used more often. It is computationally more involved, yet promises a solution close to the optimal solution.

Most literature has treated the two issues separately. Model building procedure usually deals with feature selection first, and then with ensemble creation[7]. However, separation of the issues is not desirable since a feature subset selected may be optimal for a single classifier, but may not be optimal for an ensemble. In fact, it may be better off if each member classifier of an ensemble employees different feature subsets. Since they are interrelated, these two issues need to be addressed at the same time. In this paper, we propose a framework where these two issues are addressed together with a goal of obtaining a globally optimal solution, not a combination of two locally optimal solutions. In particular, we propose an ensemble of classifiers trained with different subsets of features chosen by a GA based wrapper feature selection approach.

In a GA based wrapper approach[16], each feature is encoded as a gene and a subset of features as a chromosome. Presence or absence of a feature corresponds to 1 or 0 value respectively in the chromosome. To each chromosome does base learner correspond which is trained with only those features that are present in the chromosome. Initially, a population of chromosomes is randomly created. The validation accuracy of the trained base learner is considered as the fitness of the corresponding chromosome. Selection operation screens out those chromosomes that correspond to a low fitness value. A next generation of chromosomes is created from the selected chromosomes through cross-over and mutation. These operations are designed so that the next generation is "fitter". With this process being repeated, the population of chromosomes evolves with an expectation that fitter chromosomes emerge and survive. In most practical situations, chromosomes do improve and become fitter. In the end, only the fittest ones do survive and they qualify as "good" feature subsets.

The base learner can be any classifier with a reasonably good generalization performance and quick training capability. We found that Support Vector Machine(SVM) suits the requirement very well[15]. In one of our previous studies, an SVM resulted in a comparable performance with that of four layer multilayer perceptron neural network while its learning time was three orders of magnitude faster than that of neural network[18]. A GA wrapper approach with SVM as a base learner is depicted in Figure 3.

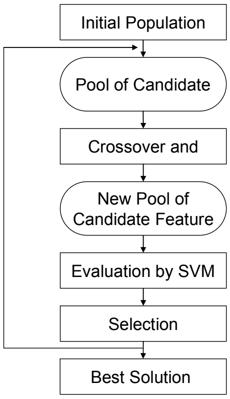


Figure 3. GA-SVM Wrapper Approach for Feature Subset Selection

Previous researches on ensemble are either based on the difference of training sets, such as bagging[3] and boosting [11], or on difference of classifiers[17]. The ensemble creation method proposed here is based on the difference among feature subsets which correspond to different chromosomes. Since GA wrapper approach involves generating a population of "good" subset features, it is quite natural to base it for our ensemble member selection process. There is one caveat, however. One needs to balance between accuracy and diversity. An ensemble is effective only when the member classifiers are both accurate and diverse. If one selects only those classifiers with a high accuracy, the ensemble may not have a sufficient diversity. If, on the other hand, one focuses on the diversity of classifiers or their corresponding subset of features, their individual members' accuracy may not be high. Thus, it is critical how to balance these two contradictory objectives in a structured way. Our approach is based on the following observation that in an early stage of genetic evolution, the chromosomes tend to be more diverse but less accurate while in a latter stage of genetic evolution, the surviving chromosomes tend to be more accurate but less diverse. Thus, we propose to stop the genetic evolution process early so that diversity is still kept while accuracy is acceptable. Then, "diverse" subset of chromosomes is chosen from the finalists. The proposed Feature Selection Ensemble (FSE) procedure is described in Figure 4.

Phase 1 (Early stopped GA)

Step 1. Evolve a population of classifiers each of which employs a subset of features for(1 − ε)·100% of time T, where T is the time necessary for complete convergence and 0<ε <1.

Step 2. Select all those "premature" classifiers.

Phase 2 (Selection of diverse Classifiers)

Step 1. Identify k groups of SVM classifiers, each of which employs the same set of features. (Even if a same set of features were used, different classifiers could result in general, depending on the particular SVM parameters such as γ and cost c used during training.)

Step 2. Choose for each group the SVM classifier f_i that has the smallest validation error, resulting in a total of k (≤ n) classifiers.

Step 3. For each classifier f_i, compute the validation output vector $\vec{f_i}$ = (f_i(1), f_i(2), …, f_i(N)), where f_i(k) is the output of classifier i for the k-th validation pattern.

Step 4. Compute hamming distance *HD(i,j)* between every pair of validation output vectors $\vec{f_i}$ and $\vec{f_i}$. (There are a total of ${}_kC_2$ such distances.)

Step 5. Select the classifiers involved in the largest HD, the second largest HD, and so on until the pre-determined number of classifiers is obtained. (If *HD(3,7)* were the largest, for instance, classifiers 3 and 7 would be obtained.)

Figure 4. Proposed Feature Selection Ensemble(FSE) procedure

The proposed Feature Selection Ensemble(FSE) procedure consists of two phases, GA phase and classifier selection phase. First, in GA phase, a typical GA wrapper is run to find T. Then, those chromosomes that emerge at (1−ε) × 100% of T are identified for phase 2. As mentioned earlier, if the GA process is let to run to converge, very accurate but similar classifiers will result, thus lowering the diversity. They are considered to be reasonably accurate and fairly diverse.

Second, in the classifier selection phase, we try to select a subset of classifiers from phase 1 for an ensemble. Since they are already reasonably accurate, we focus here the diversity. The problem of choosing the most diverse subset of classifiers based on their validation predictions can be transformed into a set packing problem, which is known as NP-complete. Thus, we employed a greedy heuristic approach where a pair of classifiers with a largest difference is selected, then another pair with a second largest difference is selected, and so on. The difference between two classifiers is estimated by the hamming distance between two respective validation output vectors corresponding to the two classifiers. The first step in phase 2 identifies k groups of SVM classifiers, each of which employs the same set of features. Note that even if a same set of features were used, different classifiers could result in general, depending on the particular SVM parameters such as γ and cost c used during training. In the second step, the SVM classifier f_i that has the smallest validation error for each group is chosen, resulting in a total of k (≤n) classifiers. Then, for each classifier f_i, compute the validation output vector $\vec{f_i}$ = (f_i (1), f_i (2), …,f_i(N)), where f_i (k) is the output of classifier i for the k-th validation pattern. Based on these vectors, hamming distance *HD(i,j)* between every pair of validation output vectors $\vec{f_i}$ and $\vec{f_i}$ is computed (There are a total of ${}_kC_2$ such distances.). Finally, at step 5, those classifiers are selected which are involved in the largest HD, then those involved in the second largest HD, and so on until the pre-determined number of classifiers is obtained. If HD(3,7) were the largest, for instance, classifiers 3 and 7 would be obtained. Or alternatively, the top χ% of the HDs could be considered and all the classifiers involved would be obtained. In the latter, the exact number of classifiers can not be determined a priori.

The proposed FSE procedure employs three kinds of parameters which must be set and tuned by the user. First is the set of parameters related to GA such as the number of generations, crossover and mutation rates. In particular, GA is usually let to run until the population fitness change becomes very small. Here, we set the generation number to 100 Second is the set of parameters related to the base learner or SVM. For Gaussian kernels, its width parameter γ and cost parameter α need to be set. They could have been put into a chromosome and "optimal" values could have been obtained through GA search. Due to an increased computational time, their ranges were set and tuned within the ranges in a randomized fashion. The third kind involves those parameters particular to our proposed approach : the number of ensemble members and "early stop" parameter ε. We set the number of ensemble size as 10 in this study. Parameter ε controls the level of accuracies and diversity of the population. If ε is too large, the chromosomes will show a high level of diversity while the quality is low. On the contrary, if ε is too small, the chromosomes will have a high quality, but the level of diversity will be low. Setting parameter ε requires an empirical justification through a priori studies on the problem complexity against GA convergence. The values of ε's were set between 10% to 20%, that is, the final chromosomes were 80% to 90% mature.

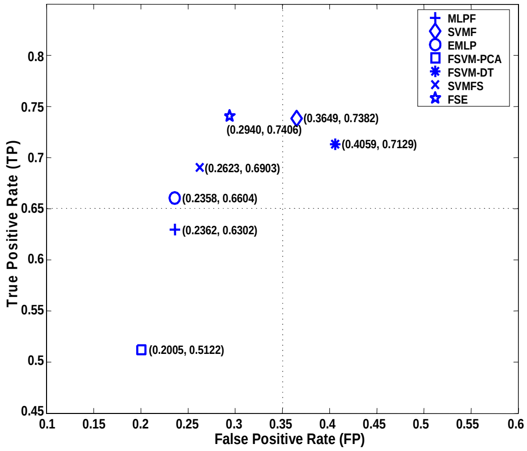


Figure 5. ROC points of the seven response models

Table 1. Seven different models to be compared

MLP-FF	Single MLP with Full Feature Set
SVM-FF	Single SVM with Full Feature Set
EMLP	Ensemble MLP with Full Feature Set
FSVM-PCA	SVM with feature subsets selected by PCA
FSVM-DT	SVM with feature subsets selected by DT
SVM-FS	SVM with Feature Selection (by GA wrapper)
FSE (proposed)	SVM Ensemble based on Feature Selection

A popular measure of performance is an ROC graph which plots true positive rate(TP) against false positive rate(FP)(see Figure 5). Point(0,1) in the figure corresponds to perfect classification. SVM's performance is denoted by a single(FP, TP) pair point in the figure. Compared to other measurements, an ROC curve has the following characteristics. An ROC curve or point is independent of class distribution or error costs. An ROC graph encapsulates all information contained in the confusion matrix, since FN is the complement of TP and TN is the complement of FP. ROC curves provide a visual tool for examining the tradeoff between the ability of a classifier to correctly identify positive cases and the number of negative cases that are incorrectly classified. Figure 5 presents (FP, TP) pairs for the seven models(see Table 1). The perfect model would fall on (0, 1). The proposed FSE lies closet to it while SVMFS and EMLP look pretty close. SVMF has a good TP but is FP is too large.

4. Pattern Selection

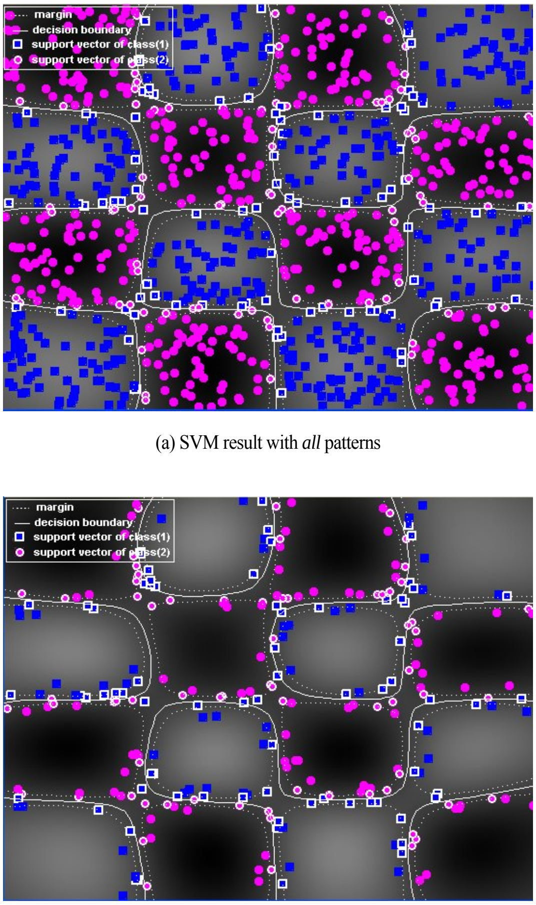
A third major problem encountered in response modeling is sheer volume of data or patterns. Generally speaking, retailers keep huge amounts of customer data. Moreover, a new customer's record is added on top of it on and on. Even though data mining algorithms are designed to deal with the problem, it is always desirable to sample the data and work on a subset of the huge data set. The problem is more severe when the most powerful classification model of our time is employed, the support vector machine[15]. In support vector machine, quadratic programming(QP) formulation is made where the dimension of kernel matrix (M×M) is equal to the number of training patterns(M). A standard QP solver has time complexity of order O(M3) : MINOS, CPLEX, LOQO and MATLAB QP routines. And the solvers using decomposition methods approximately have time complexity of T·O(Mq +q3) where T is the number of iterations and q is the size of the working set : Chunking, SMO, SVM light and SOR[6]. Needless to say, T increases as M increases. One way to circumvent this computational burden is to select some of training patterns in advance which contain most information given to learning. One of the merits of SVM theory distinguishable from other learning algorithms is that it is clear that which patterns are of importance to training. Those are called support vectors(SVs), distributed near the decision boundary, and fully and succinctly define the classification task at hand[4]. Furthermore, on the same training set, the SVMs trained with different kernel functions, i.e., RBF, polynomial and tanh, have selected almost identical subset as support vectors[13]. Therefore, it is worth finding such would-be support vectors prior to SVM training.

Here we propose neighborhood property based pattern selection algorithm(NPPS). The time complexity of NPPS is O(vM) where v is the number of patterns in the overlap region around decision boundary[14]. We utilized k nearest neighbors to look around the pattern's periphery. The first neighborhood property is that "a pattern located near the decision boundary tends to have more heterogeneous neighbors in their class-membership". The second neighborhood property dictates that "an overlap or a noisy pattern tends to belong to a different class from its neighbors". And the third neighborhood property is that "the neighbors of a pattern located near the decision boundary tend to be located near the decision boundary as well". The first one is used for identifying those patterns located near the decision boundary. The second one is used for removing the patterns located on the wrong side of the decision boundary. And the third one is used for skipping calculation of unnecessary distances between patterns, thus accelerating the pattern selection procedure. Figure 6 shows the proposed algorithm.

```
NPPS()
{
    Initialize D with randomly chosen patterns from D.
    $i \leftarrow 0,\ S_{\mathbf{o}}^{0} \leftarrow \varnothing,\ S_{\mathbf{x}}^{0} \leftarrow \varnothing,\ S^{0} \leftarrow \varnothing.$
    While D_e^0 not empty do     {
    /* Choose x satisfying [expanding criteria] */
    $D_{\mathbf{o}}^{i} \leftarrow \{x \mid Neighbors\_Entropy\,(x) > 0,\ x \in D_{\mathbf{e}}^{i}\}, D_{\mathbf{x}}^{i} \leftarrow D_{\mathbf{e}}^{i} - D_{\mathbf{o}}^{i}.$
    /* Select x satisfying [Selecting criteria] */
    $D_{\mathbf{s}}^{i} \leftarrow \{x \mid Neighbors\_Match\,(x) \geq \beta/J,\ x \in D_{\mathbf{o}}^{i}\}.$
    /* Update the pattern sets : Expanded, Non-expanded,
    Selected */
    $S_{\mathbf{o}}^{i+1} \leftarrow S_{\mathbf{o}}^{i} \cup D_{\mathbf{o}}^{i},\quad S_{\mathbf{x}}^{i+1} \leftarrow S_{\mathbf{x}}^{i} \cup D_{\mathbf{x}}^{i},\quad S^{i+1} \leftarrow S^{i} \cup D_{\mathbf{s}}^{i}.$
    /* Compute the next evaluation set */
        $D_{\mathbf{e}}^{i+1} \leftarrow \bigcup_{x \in D_{\mathbf{o}}^{i}} kNN(x) - (S_{\mathbf{o}}^{i+1} \cup S_{\mathbf{x}}^{i+1})$
      $i = i+1$   }
    return S^i
}
/* where Neighbors_entropy   $(\vec{x}, k) = \sum_{j=1}^{J} P_j(\vec{x}) \cdot \log_J \frac{1}{P_j(\vec{x})}$
Neighbors_match$(x, k) = \frac{|\{\vec{x}' \mid label\ (\vec{x}') = label\ (\vec{x}),\ \vec{x}' \in kNN\ (\vec{x})\}|}{k}$
```

Figure 6. Neighborhood Property based Pattern selection algorithm

Figure 7 visualizes one of the experimental results of artifical probles previously reported The decision boundaries in both figures look quite similar, thus, generalization performance is mimilar. The results show that NPPS reduced SVM training time up to almost two orders of magnitude with virtually no less of accuracy



(a) SVM result with *all* patterns

(b) SVM result with *selected* patterns

Figure 7. 4x4 CHECKERBOARD PRBOLEM[1)]

Table 2. Comparison between SVM trained with all data vs. sampled data.[2)]

Training Patterns		**SVs**		**Execution Time (sec)**		**Test Error (%)**	
ALL	**SUB SET**	**ALL**	**SUB SET**	**ALL**	**SUB SET**	**ALL**	**SUB SET**
81,226	8,871	35,529	6.624	4820	129	34.8	35.1

Table 2 clearly shows how the proposed pattern selection method works for DMEF4 dataset. Out of 81,226 patterns, slightly more than 10% or 8,871 were selected. However,

[1)] Decision boundary is depicted as a solid line and the margins are defined by the dotted lines in both sides of it. Support vectors are outlined. Figure (a) indicates a typical SVM result of all patterns while (b) stands for that of selected patterns by NPPS.

[2)] The column, 'SELECTED' of Execution Time includes SVM training time as well as NPPS running time.

about one fifth of support vectors were selected. The execution time was reduced from 4,820 seconds to 129 seconds, achieving almost 40 times decrease while resulting in almost identical error rate.

5. Conclusions

In this paper, we addressed three major data mining issues in response modeling. They include instability of complex nonlinear models, too many potential predictor variables to consider, and too many data in the database. For instability problem, we suggested to use bagging ensemble method. For variable selection problem, we proposed a genetic algorithm based wrapper approach. For pattern selection problem, we proposed a neighborhood property based method. All these proposed methods were empirically shown that they are effective. A real world data set from Direct Marketing Educational Foundation(DMEF) was employed. Future works include simplification or removal of phase two in wrapper approach and pattern selection for regression problem.

References

[1] Bentz, Y. and Merunkay, D., “Neural Networks and the Multinomial Logit for Brand Choice Modeling : a Hybrid Approach”, *Journal of Forecasting*, Vol.19(3), 2000, pp.177-200.

[2] Bounds, D. and Ross, D., “Forecasting Customer Response with Neural Network”, in : Fisler, E. and Beale, R. (eds.), *Handbook of Neural Computation*, Taylor & Francis,pp.1-7, 1997.

[3] Breiman, L., “Bagging predictors”, *Machine Learning*, Vol.24(2), 1996, pp.123-140.

[4] Cauwenberghs, G. and Poggio, T., *Incremental and Decremental Support Vector Machine Learning, Advances in Neural Information Processing Systems,* 13th edition, MIT Press, pp.409-415, 2001.

[5] Geman, S., Bienenstock, E. and Doursat, R., “Neural Networks and the Bias/Variance Dilemma”, *Neural Computaion*, Vol.4(1), 1992, pp.1-58.

[6] Hearst, M. A., Schölkopf, B., Dumais, S., Osuna, E. and Platt, J., “Trends and Controversies Support Vector Machines”, *IEEE Intelligent Systems,* Vol.3, 1997, pp.18-28.

[7] Ho, T. K., “The random subspace method for constructing decision forests”, *IEEE transactions on Pattern Analysis and Machine Intelligence*, Vol.20(8), 1998, pp.823-844.

[8] Moutinho, L., Curry, B., Davies, F. and Rita, P., *Neural Network in Marketing*, Routledge, 1994.

[9] Platt, J. C., “Fast Training of Support Vector Machines Using Sequential Minimal Optimization”, in : Schölkopf, B., Christopher, J. C. and Mika, B. S. (eds.), *Advances in Kernel Methods : Support Vector Machines*, MIT press, pp.185-208, 1999.

[10] Pontil, M. and Verri, A., “Properties of Support Vector Machines”, *Neural Computation,* Vol.10, 1998, pp.955-974.

[11] Schapire, R., “The Strength of Weak Learnability”, *Machine Learning*, Vol.5, 1990, pp.197-227.

[12] Sharkey, A. J., “On Combining Artificial Neural Nets”, *Connection Science*,Vol.8, 1996, pp.299-314.

[13] Schölkopf, B., Burges, C. and Vapnik, V., “Extracting supports data for given task”, *Proceedings of 1st International Conference on Knowledge Discovery and Data Mining*, AAAI, 1995, pp.252-257.

[14] Shin, H. J. and Cho, S., “Fast Pattern Selection Algorithm for Support Vector Classifiers : Time Complexity Analysis”, *Proceedings of the 3rd International Conference on Intelligent Data Engineering and Automated Learning*, AAAI, 2003, pp.1008-1015.

[15] Vapnik, V., *The Nature of Statistical Learning Theory*, 2nd edtion, Springer, 1999.

[16] Yang, J. and Honavar, V., “Feature Subset Selection using a Genetic Algorithm”, in : Liu, H. and Motoda, H. (eds.), *Feture Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, pp.117-136, 1998.

[17] Yao, X. and Liu, Y., “Making Use of Population Inform ation”, *IEEE transactions on Systems, Man, and Cybern etics*, Vol.28(3), 1998, pp.417-425.

[18] Yu, E. and Cho, S., “Keystroke Dynamics Identity Ver ification-Its Problems and Practical Solutions”, *Com puter and Security*, Vol.23(5), 2004, pp.428-440.

[19] Zahavi, J. and Levin, N., “Issues and Problems in Applying Neural Computing to Target Marketing”, *Journal of Direct Marketing*, Vol.11(4), 1997, pp.63-75.

Authors

조성준

(Cho, Sungzoon)

서울대학교 산업공학과에서 학사, 석사를 마치고 미국 University of Washington과 University of Maryland에서 컴퓨터사이언스로 각각 석사와 박사를 마쳤다. 포항공대 컴퓨터공학과 조교수를 역임하였고 현재 서울대학교 산업공학과에 부교수 겸 학과장으로 재직하고 있다. 관심분야는 데이터마이닝 알고리즘 및 데이터마이닝의 비즈니스 응용이다.

E-mail : zoon@snu.ac.kr

Tel : +82-2-880-6275

우은철

(Yu, Enzhe)

중국 하얼빈공대(HIT) 컴퓨터공학과를 졸업하고 강릉대학교 산업공학과에서 석사학위를 취득하였으며 서울대학교 산업공학과에서 데이터마이닝 전공으로 박사학위를 취득하였다. 2004년8월 삼성SDS 컨설팅본부에 입사하였으며 현재 삼성전자 무선사업부 정보전략업무를 담당하고 있다. 관심분야는 데이터마이닝, 생체인식, CRM, SCM, BPM, SOA, EA 등이다.

E-mail : ez.yu@samsung.com

신현정

(Shin, Hyunhung)

2005년 서울대학교 산업공학과에서 박사학위를 받았다. 전공분야는 데이터마이닝으로 이에 관한 제반 알고리즘을 연구, 개발한다. 특히, Support Vector Machines 및 Kernel Method, Neural Network, Graph Method에 이론적 전문성이 있다. 응용분야는 크게 Customer Relationship Management와 Bioinformatics를 겸하고 있다. 2004년부터는 독일의 Max-Planck-Institute 연구원으로 재직하고 있다.

E-mail : shin@tuebingen.mpg.de

Tel : +49-0-7071-601-824

하경남

(Ha, Kyongnam)

현재 미국 Texas A&M University에서 Industrial and Systems Engineering 박사과정 중에 있다. 서울대학교 산업공학과 학사학위를 받았으며, 동대학교 대학원 데이터마이닝 연구실에서 석사학위를 받았다. 관심분야는 스토케스틱 프로그래밍을 이용한 광고모델, 데이터마이닝 기법 기반의 응답모델, Combinatorial Problems 및 휴리스틱 알고리즘 개발 등이다.

E-mail : knamha@neo.tamu.edu

MacLachlan, L. Douglas

MacLachlan, L. Douglas is Professor, Department of Marketing and International Business, University of Washington Business School. He has a BA in Physics, MA in Statistics, MBA and Ph. D. in Marketing, all from the University of California, Berkeley. He teaches primarily marketing research, database marketing, multivariate data analysis and data mining. He has had extensive experience in consulting, marketing research and management development and has published many articles in various academic and business journals. He has been Visiting Scholar at INSEAD in Fontainebleau, France; and Visiting Professor at Catholic University of Leuven, Belgium and at Koç University in Istanbul, Turkey. As Associate Dean of the UW Business School, he helped establish the Global Executive MBA joint program between Yonsei University and UW.

E-mail : macl@u.washington.edu