How to Deal with Large Dataset, Class Imbalance and Binary Output in SVM based Response Model

Hyunjung Shin and Sungzoon Cho
Department of Industrial Engineering, College of Engineering, Seoul National University
San 56-1, Shillim-Dong, Kwanak-Gu, 151-744, Seoul, Korea
Email: {hjshin72, zoon}@snu.ac.kr

Abstract

Support Vector Machine (SVM) employs Structural Risk Minimization (SRM) principle to generalize better than conventional machine learning methods employing the traditional Empirical Risk Minimization (ERM) principle. When applying SVM to response modeling in direct marketing, however, one has to deal with the practical difficulties: large training data, class imbalance and binary SVM output. This paper proposes ways to alleviate or solve the addressed difficulties through informative sampling, use of different costs for different classes, and use of distance to decision boundary. This paper also provides various evaluation measures for response models in terms of accuracies, lift chart analysis and computational efficiency.

I. Introduction

Direct marketing is concerned with identifying likely buyers of certain products or services and promoting them to the potential buyers through various channels. A response model predicts a probability that a customer is going to respond to a promotion or offer. Using the model, one can identify a subset of customers who are more likely to respond than others. A more accurate response model will have more respondents and fewer non-respondents in the subset. By doing so, one can significantly reduce the overall marketing cost without sacrificing opportunities.

Various statistical and machine learning methods have been proposed for response modeling [4], [6], [10], [18]. Most recent is Support Vector Machine (SVM) that has been spotlighted in the machine learning community thanks to its theoretical soundness and practical performance. First, it is quite satisfying from a theoretical point of view. SVM can be characterized by three statements [29]. SVM attempts to position a decision boundary so that the *margin* between the two classes is maximized. The major parameters of SVM are taken from the training patterns. Non-linear SVM is based on the use of kernels to deal with high dimensional feature space without directly working in it. Conventional neural networks tend to overfit the training dataset, resulting in poor generalization since parameter selection is based on Empirical Risk Minimization (ERM) principle which minimizes the error on the training set. On the contrary, the SVM formulation embodies the Structural Risk Minimization (SRM) principle which minimizes the error on the training set with the lowest *capacity*. The difference allows SVM to generalize better, which is the goal in statistical learning. Theoretically, SVM includes a large class of neural networks (including radial basis functions networks), yet it is simple enough to be analyzed mathematically. Second, SVM achieved great success in practical applications as diverse as face detection and recognition, handwritten character and digit recognition, text detection and categorization, etc. *Byun et al.* gave a comprehensive up-to-date survey on SVM applications [2].

However, there are some difficulties one would face when SVM is attempted to be applied to response modeling. First, SVM training can become computationally intractable. Generally, retailers keep huge amounts of customer data. Moreover, a new customer's record will be added on top of it on and on. Unfortunately, most standard SVM QP solvers have time complexity of O(M^3) where M is the number of training patterns: MINOS, CPLEX, LOQO and MATLAB QP routines. And the solvers using decomposition methods have time complexity of $I\cdot$ O($Mq+q^3$)

where I is the number of iterations and q is the size of the working set: Chunking, SMO, SVM$^{\text{light}}$ and SOR [11], [20]. Needless to say, I increases as M increases. Second, response modeling is likely to have a severe class imbalance problem since the customers' response rates are typically very low. Most of customers belong to the non-respondents' group (class 1), while only a few customers belong to the respondents' group (class 2). Under such a circumstance, most classifiers do not behave well, and neither does SVM. Third, one has to find a way to estimate scores or likelihoods from SVM. Given a limited amount of marketing expenses, a marketer wants to maximize the return or total revenue. Thus, one would like to know who is more likely to purchase than others. Response models compute each customer's likelihood or propensity to respond to a particular offer of a product or a service. These likelihood values or scores are then used to sort the customers in a descending order. Now, the marketer simply applies a cut-off value based on the marketing expenses and only those customers whose scores are larger than the value are identified. However, an SVM classifier returns a binary output, not a continuous output which can be interpreted as a score.

In this paper, we address the obstacles mentioned above. For the intractability problem of SVM training, we present a pattern selection algorithm that reduces the training set without accuracy loss. The algorithm selects only the patterns near the decision boundary based on neighborhood properties. Its performance was previously validated for various problems in [26]. To alleviate the class imbalance problem, we propose to assign for different classes different misclassification costs which is inversely proportional to the size of the corresponding class dataset. Finally, we propose to use the distance from a pattern to the decision hyperplane in the feature space for scores. In addition, we provide various measures for evaluating the response models in both accuracy and profit.

The remaining part of this paper is organized as follows. Section II presents related works. Section III addresses the obstacles in applying SVM to response modeling. The section proposes ways to reduce the training set, to handle the class imbalance problem, and to obtain the customer scores from an SVM classifier. Section IV provides the experimental results on a direct marketing dataset. The section includes the data set description, experimental design, and performance measurements. We conclude this paper with some future works in section V.

II. Related Work

Although SVM is applied to a wide variety of application domains, there have been only a couple of SVM application reports in response modeling. *Cheung et al.* used SVM for content-based recommender systems [3]. Web retailers implement a content-based system to provide recommendations to a customer. The system automatically matches his/her interests with product-contents through web-pages, newsgroup messages, and new items. It is definitely a form of direct marketing that has emerged by virtue of recent advances in the world wide web, e-business, and on-line companies. They compared Naive Bayes, C4.5 and 1-nearest neighbor rule with SVM. The SVM yielded the best results among them. More specific SVM application to response modeling was attempted by *Viaene et al.* [30]. They proposed a Least Square SVM (LS-SVM) based wrapper approach. *Wrapper* indicates an input variable selection procedure working together with a learning algorithm, and it is frequently compared with alternative procedure, *filter*, that performs variable selection independently from a learning algorithm. In their study, the input variable pool was composed of RFM and non-RFM variables from the customer dataset provided by a major Belgian mail-order company. Then, the wrapper approach was performed in a sequential backward fashion, guided by a best-first variable selection strategy. Their approach, a wrapper around the LS-SVM response model, could gain significant reduction of model complexity without degrading predictive performance.

Now, let us focus on the researches related to the difficulties we addressed in this paper. First, the most straightforward method to reduce a large training set is random sampling. In SVM, however, the patterns near the decision boundary are critical to learning. The training set reduced by random sampling may omit those, thus would lead to significantly poorer prediction results. Some SVM researchers thus have attempted to identify those training patterns near the decision boundaries. *Lyhyaoui et al.* implemented RBF classifiers which somewhat resemble SVMs, to clear the difference between both methods [16]. RBF classifiers were built on the patterns near the decision boundary. To find them, they proposed 1-nearest neighbor method in the opposite class after class-wise clustering. But this method makes an impractical assumption that the training set is clean. An approach focusing more on SVM was proposed by *Almeida et al.* who conducted *k*-means clustering on the entire training set [1]. All patterns were selected for heterogeneous clusters while only the centroids were selected for homogeneous clusters. The drawbacks of this research are that it is not clear how to determine the number of clusters, and that the clustering performance is generally unstable [15]. More recently, *Shin and Cho* proposed a neighborhood properties based pattern selection algorithm (NPPS) [26], which will be introduced in section III-A.

Second, regarding class imbalance, many researchers have recognized this problem and suggested several methods: enlarging the small class dataset by random sampling, reducing the large class dataset by random sampling, and ignoring the small class dataset and using only the large class dataset to build a one-class recognizer [12]. *Japkowicz* compared the three commonly used methods above on the degree of concept complexity using a standard neural network classifier. All the methods generally improved the performance of the learning algorithm. In particular, the first two methods were very effective especially as the concept complexity increases while the last one was relatively less accurate. *Ling et al.* addressed the specificity of the class imbalance problem which resides in marketing datasets [14]. They did not attempt to balance the imbalanced class ratio for better predictive accuracy. Instead, to circumvent the class imbalance problem, a marketing specific evaluation measure, *lift index*, was suggested. Lift index provides the customer's rank (score) by reflecting the confidence of classification result. They argued that even if all of the patterns are predicted as one class, as long as the learning algorithm produces suitable ranking of the patterns, the imbalanced class distribution in the training set would no longer be a problem. However, in their experiments all the best lift index were obtained when the sizes of the classes were equal. Thus, they recommended to reduce the large class dataset so that its size becomes equal to that of the small class. Alternatively, different misclassification costs can be incorporated into classes, which avoids direct artificial manipulation on the training set [13].

Third, getting scores from a logistic regression model or a neural network model with sigmoidal output function is well known. The output gives a value of probability belonging to the class, that is ranged from 0 to 1. Thus the output value is used as a score for sorting the customers. *Ling et al.* made use of the ada-boost algorithm [7], an ensemble approach, to get the customers' scores. Basically, ada-boost maintains a sampling probability distribution on the training set, and modifies the probability distribution after each classifier is built. The probability of patterns with an incorrect prediction by the previous classifier is increased. So these patterns will be sampled more likely in the next round of boosting, to be learnt correctly. A pattern's probability to be incorrectly predicted allowed a corresponding rank [14]. Sometimes, scores could be directly estimated by regression model having continuous target value, i.e., the dollars spent or the amount of orders. To do that, however, one needs to diagnose the problems the target variable has and conduct suitable remedies to cure them. *Malthouse* [18] built a regression model to estimate the dollars spent on DMEF4 [31]. There was a large number of extreme values and the distribution was highly skewed. The extreme values could have a large influence on estimate values under least squares. And the variance of target variable most likely increased with its mean (heteroscedasticity). Thus, he performed log transformation to alleviate skewness and heterocedasticity, and used *winsorization* to exclude some extreme values of target. The predicted value of the dollars spent was used as a score in lift chart analysis. The lift result by means of regression problem based score will be briefly compared with that by means of classification problem based score in section IV-D. Generally speaking, regression problem requires more information from input variables than classification problem does. In other words, binary classification is the simplest subproblem of regression. Producing good scores from marketing regression model is difficult at the present time. In addition, since SVM theory stemmed

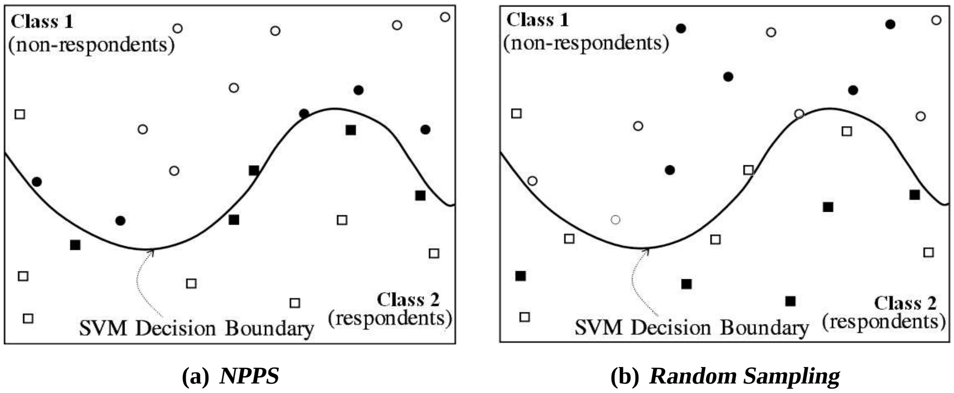


Fig. 1. NPPS and random sampling select different subsets: outlined circles and squares are the patterns belonging to class 1 (non-respondents' group) and class 2 (respondents' group), respectively. Black solid circles and squares are the selected patterns.

from classification context [23], it is natural to get scores from an SVM classifier.

III. Support Vector Machines for Response Modeling

A. Large Training Dataset

We propose to use a neighborhood property based pattern selection algorithm (NPPS) [25], [26]. The idea is to select only those patterns located around decision boundary since they are the ones that contain most information. Contrary to a usually employed "random sampling," this approach can be viewed as "informative or intelligent sampling." Fig. 1 conceptually shows the difference between NPPS and random sampling in selecting a subset of the training data. NPPS selects the patterns in the region around the decision boundary, while random sampling selects those from the whole input space. Obviously, the decision boundary is not known until a classifier is built. Thus, the algorithm utilizes neighborhood properties to infer the proximity of a pattern to the decision boundary. *The first neighborhood property* is that "*a pattern located near the decision boundary tends to have more heterogeneous neighbors in their class membership*." Thus, the proximity of pattern $\vec{x}$'s to the decision boundary was estimated by "*Neighbors_Entropy ($\vec{x}, k$)*", which is defined as the entropy of the pattern $\vec{x}$'s *k*-nearest neighbors' class labels,

$$\text{Neighbors\_Entropy } (\vec{x}, k) = \sum_{j=1}^{J} P_j \cdot log_J \frac{1}{P_j},$$

where P_j is defined as k_j/k where k_j is the number of neighbors belonging to class j among the k nearest neighbors of $\vec{x}$, J classification problem. A pattern with a positive Neighbors_Entropy($\vec{x}, k$) value is assumed to be close to the decision boundary, thus selected for training. Among the patterns having a positive value of Neighbors_Entropy ($\vec{x}, k$), however, overlapping or noisy patterns are also present. These patterns have to be identified and removed as much as possible since they are more likely to be misclassified. *The second neighborhood property* thus dictates that "*an overlap or a noisy pattern tends to belong to a different class from its neighbors*." If a pattern's own label is different from the majority label of its neighbors, it is likely to be incorrectly labeled. The measure "*Neighbors_Match ($\vec{x}, k$)*" is defined as the ratio of $\vec{x}$'s neighbors whose label matches that of $\vec{x}$,

$$\text{Neighbors\_Match } (\vec{x}, k) = \frac{|\{\vec{x}'|label(\vec{x}') = label(\vec{x}),\ \vec{x}' \in kNN(\vec{x})\}|}{k},$$

where $kNN(\vec{x})$ is the set of k nearest neighbors of $\vec{x}$. The patterns with a small Neighbors_Match($\vec{x}, k$) value is likely to be the ones incorrectly labeled. Only the patterns satisfying the two conditions, Neighbors_Entropy ($\vec{x}, k$) > 0 and Neighbors_Match ($\vec{x}, k$) $\geq \beta \cdot \frac{1}{J}$ ($0 < \beta \leq 1$), are selected. However, the NPPS evaluating kNNs for

```
NPPS ($\mathbf{D}$, $k$) $\{$

    [0] Initialize $\mathbf{D}_\mathbf{e}^0$ with randomly chosen patterns from $\mathbf{D}$.
        Constants $k$ (the number of neighbors) and $J$ (the number of classes) are given.
        Initialize $i$ and various sets as follows:
        $i \leftarrow 0, \quad \mathbf{S}_\mathbf{o}^0 \leftarrow \emptyset, \quad \mathbf{S}_\mathbf{x}^0 \leftarrow \emptyset, \quad \mathbf{S}^0 \leftarrow \emptyset.$

    while $\mathbf{D}_\mathbf{e}^i \neq \emptyset$ do
        [1] Choose $\vec{x}$ satisfying [Expanding Criteria].
                $\mathbf{D}_\mathbf{o}^i \leftarrow \{\vec{x} \mid Neighbors\_Entropy\,(\vec{x}, k) > 0,\ \vec{x} \in \mathbf{D}_\mathbf{e}^i\}.$
                $\mathbf{D}_\mathbf{x}^i \leftarrow \mathbf{D}_\mathbf{e}^i - \mathbf{D}_\mathbf{o}^i.$

        [2] Select $\vec{x}$ satisfying [Selecting Criteria].
                $\mathbf{D}_\mathbf{s}^i \leftarrow \{\vec{x} \mid Neighbors\_Match\,(\vec{x}, k) \geq \beta/J,\ \vec{x} \in \mathbf{D}_\mathbf{o}^i\}.$

        [3] Update the pattern sets: the expanded, the non-expanded, and the selected.
                $\mathbf{S}_\mathbf{o}^{i+1} \leftarrow \mathbf{S}_\mathbf{o}^i \cup \mathbf{D}_\mathbf{o}^i$ , $\mathbf{S}_\mathbf{x}^{i+1} \leftarrow \mathbf{S}_\mathbf{x}^i \cup \mathbf{D}_\mathbf{x}^i$ , $\mathbf{S}^{i+1} \leftarrow \mathbf{S}^i \cup \mathbf{D}_\mathbf{s}^i.$

        [4] Compute the next evaluation set $\mathbf{D}_\mathbf{e}^{i+1}$.
                $\mathbf{D}_\mathbf{e}^{i+1} \leftarrow \bigcup_{\vec{x} \in \mathbf{D}_\mathbf{o}^i} k\mathbf{NN}(\vec{x}) - (\mathbf{S}_\mathbf{o}^{i+1} \cup \mathbf{S}_\mathbf{x}^{i+1}).$
        [5] $i \leftarrow i + 1.$
    end
    return $\mathbf{S}^i$
$\}$
```

Fig. 2. NPPS

M patterns have time complexity of O(M^2), so the pattern selection process itself can be time-consuming. To accelerate the pattern selection procedure, let us consider *the third neighborhood property*, "*the neighbors of a pattern located near the decision boundary tend to be located near the decision boundary as well*." Assuming the property, one may compute only the neighbors' label entropy for the patterns near the decision boundary instead of all the training patterns. Only the neighbors of a pattern satisfying Neighbors_Entropy $(\vec{x}, k) > 0$, are evaluated in the next step. This lazy evaluation reduce the time complexity from O(M^2) to O(vM), where v is the number of patterns in the overlap region. In most practical problems, $v < M$ holds. We provided the time complexity analysis for the fast NPPS in [27], and a systematic procedure for determining the value of *k* in [28]. The algorithm is shown in Fig. 2.

B. Class Imbalance

Usually there are many more non-respondents than respondents in training datasets. Thus, sub-sampling of non-respondent class data is the most widely used method to balance the datasets. However, random sampling allows "important" patterns near the decision boundary to be missed. Those patterns are likely to become support vectors. Loss of those patterns could result in a poor generalization performance of SVM. Thus, instead, we propose to employ different misclassification costs to different class errors in the objective function, which is naturally allowed in SVM. This approach is not only safer, but also more principled.

Consider the most general SVM formulation allowing both non-separable and nonlinear cases, given M patterns $(\vec{x}_i, y_i), i = 1, \cdots, M$ where $\vec{x}_i \in \Re^d$ and $y_i \in \{-1,\ 1\}$. Let us assume that patterns with $y_i = -1$ belong to non-respondents' group (class 1) while those with $y_i = 1$ belong to respondents' group (class 2). SVM training involves solving the following quadratic programming problem which yields the largest margin ($\frac{2}{\|w\|}$) between classes,

$$\begin{aligned} \min \quad & \Theta\left(\vec{w}, \xi\right) = \frac{1}{2}||\vec{w}||^2 + C\sum_{i}^{M}\xi_i, \\ \text{s. t.} \quad & y_i\left(\vec{w}\cdot\Phi(\vec{x}_i) + b\right) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad i = 1, \ldots, M, \end{aligned} \tag{1}$$

where $\vec{w} \in \Re^d$, $b \in \Re$. The ξ's are nonnegative slack variables for a non-separable case, which play a role of allowing a certain level of misclassification. The $\Phi(\cdot)$ is a mapping function for a nonlinear case that projects patterns from the input space into a feature space. The C is the original cost term which is equally applied to the misclassified patterns.

Now, let m_1 and m_2 denote the size of class 1 and class 2 data sets, respectively, with $m_1 \gg m_2$ and $M = m_1 + m_2$. One way to alleviate data imbalance problem is to assign to a large class a smaller cost while assign to a small class a larger cost, which assures that a small class is not “neglected.” In response modeling, there are many more non-respondents than respondents, thus the size of non-respondents is m_1 while that of respondents is m_2. One way to accomplish it is to define and assign C_1 and C_2 to each class as below

$$\min \quad \Theta\left(\vec{w}, \xi\right) = \frac{1}{2}||\vec{w}||^2 + C_1\sum_{i\in non-respondents}\xi_i + C_2\sum_{i\in respondents}\xi_i, \tag{2}$$

where C_1 and C_2 are defined respectively as

$$\begin{aligned} C_1 &= \frac{m_2}{M}\cdot C, \\ C_2 &= \frac{m_1}{M}\cdot C. \end{aligned} \tag{3}$$

In order to emphasize small respondent dataset, a larger cost C_2 was assigned to its error term.

C. Getting Scores from an SVM Classifier

The objective of response modeling is to compute the likelihood or propensity of each customer to respond to a particular offer so that the mailing response or profit is maximized. Lift chart is commonly used for this purpose, which sorts the customers by the descending order of their estimated value (score), and then the customers in the first several deciles are finally decided to be mailed. Although an SVM classifier returns a binary output (-1 or 1) from the decision function

$$f(\vec{x}) = sign\left(\vec{w}\cdot\Phi(\vec{x}) + b\right) = sign\left(\sum_{i\in SVs} y_i\alpha_i\Phi(\vec{x}_i)\cdot\Phi(\vec{x}) + b\right) = sign\left(\sum_{i\in SVs} y_i\alpha_i K(\vec{x}_i, \vec{x}) + b\right), \tag{4}$$

one can still estimate a score based on the *distance between a pattern and the decision boundary*. In other words, we assume that a pattern located further from the decision boundary has a higher probability of belonging to that class. The decision boundary hyperplane $\bar{f}(\vec{x})$ in a feature space Φ is represented as

$$\bar{f}(\vec{x}) = \sum_{i\in SV} y_i\alpha_i\Phi(\vec{x}_i)\cdot\Phi(\vec{x}) + b = 0 \tag{5}$$

from Eq. (4). It should be noted that the decision boundary is a hyperplane in the feature space Φ even though it is a nonlinear hyper-surface in the input space. In the feature space, hence, the distance from a pattern $\Phi(\vec{x})$ to the decision boundary hyperplane $\bar{f}(\vec{x})$ can be calculated by

$$\text{dist}(\Phi(\vec{x}), \bar{f}(\vec{x})) = \frac{|\bar{f}(\vec{x})|}{|\sum\limits_{i\in SV} y_i\alpha_i\Phi(\vec{x}_i)|^2}. \tag{6}$$

The exact value of the distance is possible to obtain from Eq.(6) by using *kernel trick* [29] even though the actual mapping function $\Phi(\cdot)$ is not known, the feature space Φ could be an infinite dimensional space, and furthermore, multiple mapping functions, $\Phi(\cdot)$s, could exist. A kernel function $K(\vec{x}, \vec{x}')$ replaces $\Phi(\vec{x}) \cdot \Phi(\vec{x}')$ particularly during denominator calculation in Eq.(6). However, one does not need to know the exact value of the distance, since only a relative score or rank is all that is required in lift chart analysis. The denominator in Eq. (6) is common for all patterns, thus the signed function value in the numerator, $\bar{f}(\vec{x})$, can be used in computing ranks. The larger the value of $\bar{f}(\vec{x})$, the lower the rank of that particular customer's likelihood becomes.

IV. Experiments

This section provides the empirical results of SVM based response modeling with the proposed approach. In particular, the performance evaluation measures pertinent to response modeling are also proposed and measured.

A. Dataset

In machine learning literature, so-called standard and public datasets are used. But, in response modeling, or in direct marketing fort that matter, such datasets do not seem to exist. Many papers use a unique dataset which is not available for other researchers. The only exception seems to be datasets from the Direct Marketing Educational Foundation (DMEF) [31]. The DMEF makes marketing datasets available to researchers. Dataset DMEF4, was used in various researches [8], [18], [19]. It is concerned with an up-scale gift business that mails general and specialized catalogs to its customer base several times each year. The problem is to estimate how much each customer will spend during the test period, 09/1992–12/1992, based on the training period, 12/1971–06/1992. There are 101,532 patterns in the dataset, each of which represents the purchase history information of a customer. Each customer is described by 91 input variables. A subset of 17 input variables, some original and some derived, were employed just as in [18] (see table I). The dataset has two target variables, TARGDOL (target mailing dollars) and TARGORD (target mailing orders). The former indicates the purchase dollar amount during the test period, and the latter indicates the number of orders during the test period. The TARGDOL or the TARGORD could be directly estimated by building a regression model. *Malthouse* built a regression model to estimate the value of TARGDOL. But due to the problems of regression (section II), we formulated the problem into a classification one. A new target variable, RESPONSE, was defined as follows: 1 if TARGDOL (TARGORD) $>$ 0, 0 otherwise. *Ha et al.* used the same derivation to fit a neural network classifier [8]. Thus, all the customers were categorized into either a non-respondent (class 1) or a respondent (class 2). The response rate is 9.4%, which means the class distribution of the dataset is highly imbalanced.

B. SVM Models

To verify the effectiveness of NPPS described in section III-A, we considered seven SVMs trained with randomly selected patterns. They are denoted as R*-SVM where '*' indicates the ratio of random samples drawn without replacement. S-SVM denotes the SVM trained with the patterns selected by NPPS (see table II). Each model was trained and evaluated using five-fold cross-validation. The number of neighbors (k) of NPPS, was set to 4 according to guidelines suggested in [28]. All the SVM models in table II use the same hyper-parameter values to equalize their effects. The RBF kernel, $exp\left(-||\vec{x} - \vec{x}'||^2/2\sigma^2\right)$, was used with parameter σ set to 0.5, and the misclassification tolerance parameter C in Eq. (1) set to 10. These parameter settings were determined through a trial-error approach over the combination of C and σ, ($\{0.1, 1, 10, 100, 1000\} \times \{0.25, 0.5, 1, 2, 3\}$), using ten fold cross-validation performance. The class imbalance problem addressed in section III-B appeared in all the eight datasets. The sets selected by random sampling showed the common class ratio of $m_1 : m_2 = 90.6\% : 9.4\%$. That is also the same ratio as the original training set since we conducted a stratified random sampling by the target variable. The training set reduced by NPPS, however, showed a different class ratio, $m_1 : m_2 = 65.5\% : 34.5\% \; (= 5810 : 3061)$ on

TABLE I

INPUT VARIABLES

Variable	Formula	Description
Original Variables		
purseas		number of seasons with a purchase
falord		life-to-date (LTD) fall orders
ordtyr		number of orders this year
puryear		number of years with a purchase
sprord		LTD spring orders
Derived Variables		
recency		order days since 10/1992
tran38	1/recency	
tran51	$0 \leq$ recency < 90	
tran52	$90 \leq$ recency < 180	five dummy variables (tran51–55) having the value 1, if the condition is satisfied, otherwise the value 0
tran53	$180 \leq$ recency < 270	
tran54	$270 \leq$ recency < 366	
tran55	$366 \leq$ recency < 730	
comb2	$\sum_{i=1}^{14}$ prodgrp i	number of product groups purchased from this year
tran25	1 / (1+lorditm)	inverse of latest-season items
tran42	$\log(1 + ordtyr \times falord)$	interaction between the number of orders
tran44	$\sqrt{ordhist \times sprord}$	interaction between LTD orders and LTD spring orders
tran46	$\sqrt{comb2}$	

average. Even though NPPS improved the ratio of the smaller class from 9.4% up to 34.5%, the imbalance problem still remained. Thus, the different misclassification costs, C_1 and C_2 were set on every dataset as they were defined in Eq. (3). C_1 and C_2 of R*-SVM were 0.94 $(= 0.094 \times 10)$ and $9.06 (= 0.906 \times 10)$, respectively. On the other hands, those of S-SVM were $3.45 (= 0.345 \times 10)$ and $6.55 (= 0.655 \times 10)$.

C. Performance Measurements

The performances of the eight SVM response models were compared in terms of three criteria: accuracies, lift chart and computational efficiency.

1) Accuracies: The accuracy of a classifier can be described by a confusion matrix (see table III). Let m_{ij} denote the number of patterns which were classified as class j but whose actual class label is class i. A most widely used accuracy measurement is an Average Correct-classification Rate (ACR) which is defined as

$$\text{Average Correct-classification Rate (ACR)} \quad = \frac{TN+TP}{M} = \frac{m_{11}+m_{22}}{M}.$$

But, the average correct-classification rate can be misleading in an imbalanced dataset where the heavily-represented class is given more weight. Receiver Operating Characteristic (ROC) analysis is usually performed as well [21], which measures the classifier's accuracy over the whole range of thresholds in terms of Specificity (Sp) and Sensitivity (Se) [22]. They are defined as

$$\begin{aligned} \text{Specificity (Sp)} \quad &= \frac{TN}{TN+FP} = \frac{m_{11}}{m_{11}+m_{12}} = \frac{m_{11}}{m_1}, \\ \text{Sensitivity (Se)} \quad &= \frac{TP}{FN+TP} = \frac{m_{22}}{m_{21}+m_{22}} = \frac{m_{22}}{m_2}. \end{aligned}$$

TABLE II

SVM MODELS:

The number of patterns selected from NPPS slightly varies with the given set of each fold, thus it is represented as an average over the five reduced training sets.

Model	**No. of Training Data**	**Training Data**
R05-SVM	4060	5% random samples
R10-SVM	8121	10% random samples
R20-SVM	16244	20% random samples
R40-SVM	32490	40% random samples
R60-SVM	48734	60% random samples
R80-SVM	64980	80% random samples
R100-SVM	81226	100% random samples
S-SVM	avg. 8871	the patterns selected by NPPS

TABLE III

CONFUSION MATRIX:

FP, FN, TP and TN means false positive, false negative, true positive, and true negative in due order where TP and TN are the correct classification.

		Classified		
		class 1 (non-respondent)	class 2 (respondent)	
Actual	class 1 (non-respondent)	m_{11} (TN)	m_{12} (FP)	m_1
	class 2 (respondent)	m_{21} (FN)	m_{22} (TP)	m_2

Since we fixed the classification threshold at 0 in the SVM decision function Eq. (4), however only one pair of Sp and Se per model was available. Thus, here the ROC plot has the eight pairs of (1-Sp, Se) scattered for their comparison. Another accuracy measure, Balanced Correct-classification Rate (BCR), was defined so as to incorporate Sp and Se into one term. BCR enforces balance in the correct classification rate between two classes. It is defined as

$$\text{Balanced Correct-classification Rate (BCR)} \quad = \text{Sp} \cdot \text{Se} = (\tfrac{m_{11}}{m_1}) \cdot (\tfrac{m_{22}}{m_2}).$$

2) Lift Chart Analysis of Response Rate and Profit: Once the test patterns were sorted in a descending order according to $\bar{f}(\vec{x})$, two kinds of lift charts were investigated. One is for *response rate*, and the other for *profit*. From the business point of view, the ultimate goal of direct mailing is to maximize the profit rather than the response rate itself [17]. Thus we evaluated the eight competing SVM models from a profit aspect as well. For profit lift chart analysis, another target variable of DMEF4 dataset, *TARGDOL (target mailing dollar)*, was associated with the rank of $\bar{f}(\vec{x})$, which indicates the purchase dollar amount during the test period. Two measurements were used in evaluating lift charts. One is the average response rate or profit in the top decile, "Top-Decile". This measures how well two model identifies a small number of highly likely respondents. The other is "Weighted-Decile" defined as

$$\text{Weighted-Decile} = \frac{\{1.0 \times d_1 + 0.9 \times d_2 + 0.8 \times d_3 \ldots + 0.1 \times d_{10}\}}{1.0 + 0.9 + 0.8 + \ldots + 0.1},$$

where $d_i,\ (i = 1, \ldots\ 10)$ is a cumulative average response rate or profit till i^{th} decile in the lift table. This measures how well the model identifies a larger number of likely respondents in a larger rollout. A similar evaluation by two measurements has been adopted in data mining competitions [14].

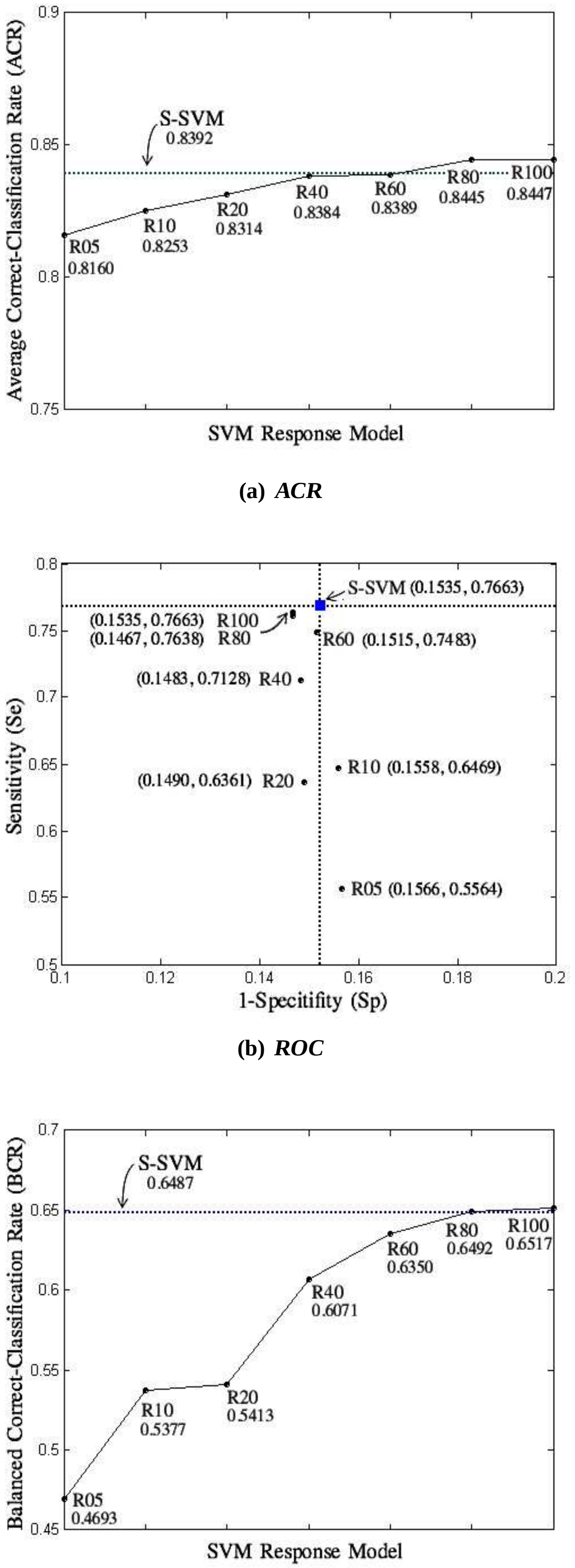
3) Computational Efficiency: The evaluation was done in several measures: the number of training patterns, training time, and the number of support vectors, and recall time. The number of patterns directly influences the time complexity. The training time of SVM increases in proportion to the cube of the number of training patterns (in case of standard QP solver). The recall time increases linearly to the number of support vectors. Training time is of important concern to a direct marketer who is in charge of SVM modeling with a huge amount of data, while recall time is critical when the model is deployed to work in a real-time application such as fraud detection. Although recall time is not a primary issue in response modeling, we measured it for potential use to another application.

D. Results

We now give the experimental results of the eight SVM response models in the order of accuracies, lift chart analysis, and computational efficiency.

Fig. 3 shows how the eight SVM response models performed in terms of ACR, ROC, and BCR. First, Fig. 3(a) indicates a mean ACR over five-fold cross-validation of each SVM model. For the sake of convenience, R*-SVM is briefly denoted as ‘R*’ in the figure. Sampling more patterns results in higher ACR, but the increasing rate is not very high. From R05 to R100, only about 3.52% (=$\{0.8447-0.8160\}/0.8160\times100\%$) of accuracy was gained from 1,900% (= $\{100-5\}/5\times100\%$) data increase. The S-SVM achieved ACR in the range of those from R60–R80. However, we could not make good evaluation of the model comparison using ACR because of class imbalance. In Fig. 3(b), the eight pairs of (1-Sp, Se) were plotted in ROC chart. A point located upper left corresponds to a better performance. The ACR is effectively broken down into two classwise accuracies, Sp for non-respondents (class 1) and Se for respondents (class 2). The *Sp*s of the eight SVM models are similar, while the *Se*s show a significant differences. It should be noted that it is Se, accuracy for respondents’group, that is of greater importance to direct marketers, since their primary goal is to identify the respondents, not the non-respondents. S-SVM achieved a best Se, better than that of even R100-SVM. Fig. 3(c) shows the BCRs of the eight SVM response models. BCR clearly distinguished the accuracies of the eight SVM models. Sampling more data results in a larger BCR also. The BCR of S-SVM is almost same as that of R100-SVM.

Fig. 4 illustrates the lift chart of the cumulative average response rate. The base average response rate of DMEF4 dataset was 9.4%, which is represented as a solid horizon at the bottom of the chart. Two observations can be made. First, all the SVM response models did better than the base response rate. Second, more training patterns lead to a better lift of the response rate. R100-SVM showed the best performance while the R05-SVM showed the worst. Models trained with more patterns showed a steeper lift in the first several deciles. The lift curve of S-SVM was almost identical to that of R80-SVM. It is illuminating to compare the curve shape of S-SVM with that of R10-SVM represented as a dash-dot line. Although the two models had almost the same number of training patterns, they were significantly different in the lift performance. Fig. 5 shows the results of the lift measures described in section IV-C.2: Top-Decile and Weighted-Decile. From the top 10 percentile of customers, R100-SVM obtained 51.45% response rate (see Fig. 5(a)). The Top-Decile response rate of S-SVM was 48.65%, which is almost equal to that of R80-SVM, 48.79%. Fig. 5(b) shows the results of Weighted-Decile response rates. R100-SVM still did best, and S-SVM and R80-SVM came second. But the gap between the first and the second was not so big as in the Top-Decile response rate. Now, Fig. 6 and Fig. 7 describe the lift chart results in terms of the profit. The average purchase dollar amount of DMEF4 was $48 when averaged over the respondents’ group, but $4.5 when averaged over all customers. The horizon line in the lift chart of Fig. 6 represents the $4.5 base average profit. All the models did better than the base average profit and an SVM with more training patterns produced a higher profit in the first several deciles. But in terms of the profit lift, S-SVM showed performance comparable to that of



(a) ***ACR***

(b) ***ROC***

(c) ***BCR***

Fig. 3. Accuracies: accuracy of R*-SVM is depicted as a solid circle while that of S-SVM is represented as a dotted reference line.

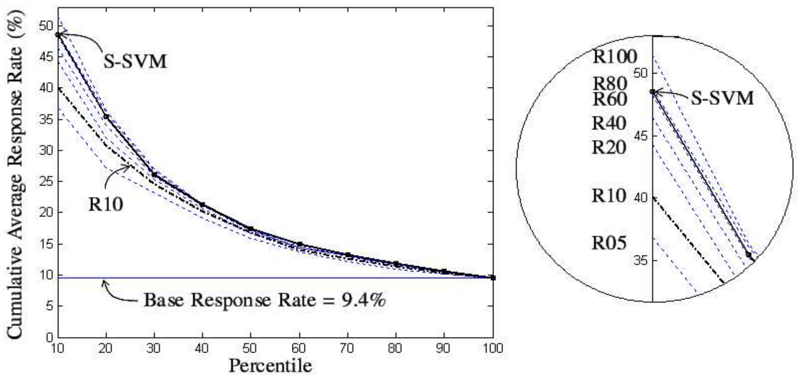
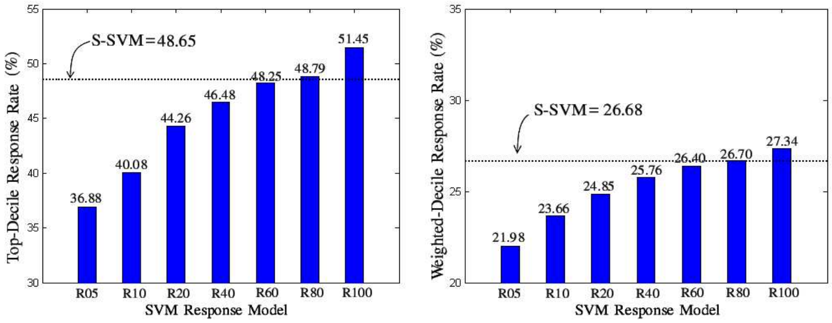


Fig. 4. Lift chart of cumulative average response rate: R*-SVMs are depicted dotted lines but among them R10-SVM is represented as a dash-dot line. S-SVM is represented as a solid-dot line.



(a) ***Top-Decile response rate*** **(b)** ***Weighted-Decile response rate***

Fig. 5. Top-Decile response rate and Weighted-Decile response rate: R*-SVM is depicted as a bar while S-SVM is represented as a dotted reference line.

R100-SVM. It is also remarkable that the profit lifts of R100-SVM or S-SVM outperformed those of *Malthouse* who got the scores by fitting the problem as a regression one [18]. For the cumulative average profit (dollars) of the second decile, *Malthouse*' regression model recorded \$12–\$15 while the SVM classification model recorded \$17–\$18. Fig. 7 illustrates the Top-Decile profit and the Weighted-Decile profit. The Top-Decile profit and the Weighted-Decile profit of R100-SVM were \$23.78 and \$12.99, respectively, and those of R80-SVM were \$22.25 and \$12.56. S-SVM were \$23.53 in the Top-Decile profit and \$12.77 in the Weighted-Decile profit, which were slightly less than those of R100-SVM but more than those of R80-SVM.

Finally, table IV shows the results of computational efficiency measures in rows: the number of training patterns, training time, the number of support vectors, its proportion to training patterns, and recall time. We used OSU SVM Classifier Matlab Toolbox, which is a hybrid algorithm of SMO and SVM^{light}, and is known as one of the fastest solvers [32]. Training time increased proportionally to the number of training patterns with the peak of 4820 (sec) for R100-SVM. On the other hand, S-SVM took only 68 (sec). The total time of S-SVM was 129 (sec), when the NPPS running time, 61 (sec), was included. Note that SVM training is usually performed several times to find a set of optimal parameters, but the pattern selection is performed only once. In the third row, the number of support vectors is represented. At most, half of the random sampling training patterns were support vectors while 74% of the NPPS selected training patterns were support vectors. The result confirms that the NPPS' selection of training patterns was more efficient. Recall time was proportional to the number of support vectors as shown in the last row. Overall, the computational efficiency of S-SVM was comparable to that of R10-SVM or R20-SVM.

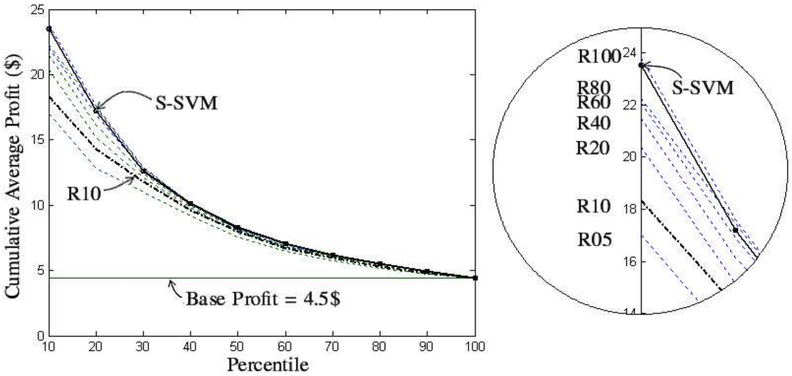
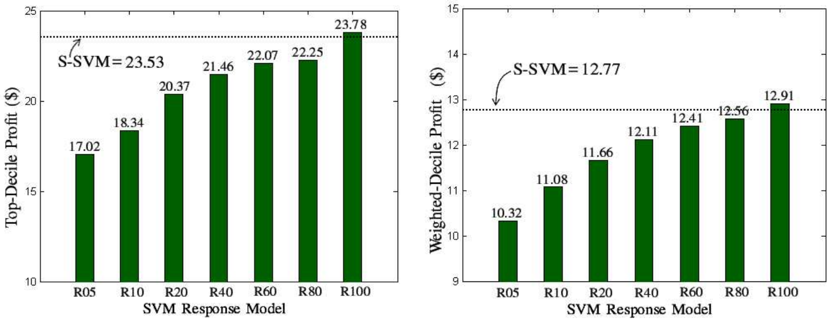


Fig. 6. Lift chart of cumulative average profit: R*-SVMs are depicted dotted lines but among them R10-SVM is represented as a dash-dot line. S-SVM is represented as a solid-dot line.



(a) ***Top-Decile profit*** **(b)** ***Weighted-Decile profit***

Fig. 7. Top-Decile profit and Weighted-Decile profit: R*-SVM is depicted as a bar while S-SVM is represented as a dotted reference line.

V. Conclusions and Discussions

In this paper, we introduced SVM, a powerful classification model, and practical difficulties when applying it to response modeling in direct marketing: large training data, class imbalance and binary SVM output. We then proposed how to alleviate and solve those difficulties: informative sampling, different costs for different classes, and use of distance to decision boundary. In the experiments, we showed that the proposed solutions worked quite well. In particular, several models were trained and evaluated in terms of accuracies, lift chart analysis and computational efficiency. The SVM trained with the patterns selected by proposed NPPS (S-SVM) were compared with the ones trained with random samples (R*-SVMs where '*' indicates the sampling percentage). Fig. 8 summarizes the results in terms of various measures. The horizontal bars in the figure shows the performance of S-SVM relative to those

TABLE IV

Computational Efficiency of SVM response models

	R05	R10	R20	R40	R60	R80	R100	S
No. of Training Patterns	4,060	8,121	16,244	32,490	48,734	64,980	81,226	8,871
Training Time (sec)	14	57	149	652	1,622	2,907	4,820	68
No. of Support Vectors (%)	1,975 (49%)	4,194 (52%)	7,463 (46%)	14,967 (46%)	22,193 (46%)	28,968 (45%)	35,529 (43%)	6,624 (74%)
Recall Time (sec)	17	31	56	112	166	237	381	45

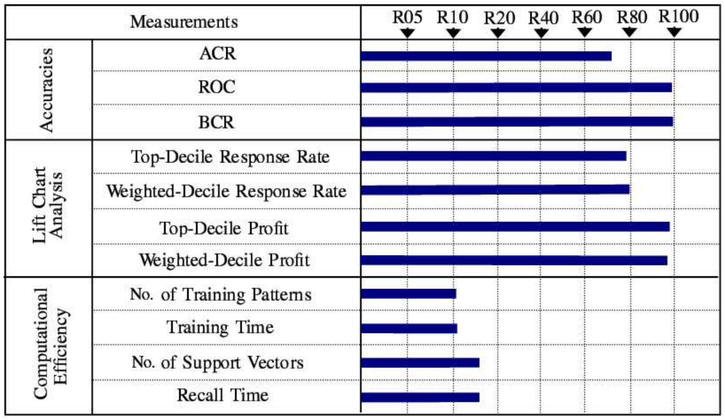


Fig. 8. How well S-SVM performed relative to R*-SVMs

of R*-SVMs in various measures. S-SVM achieved the accuracies and uplifts comparable to those of R80-SVM and R100-SVM with a computational cost comparable to those of R10-SVM and R20-SVM.

Here, we would like to address some future research works. First, in lift chart analysis, we used two measures, Top-Decile and Weighted-Decile. The former is for specifying a small number of customers in the top decile, while the latter is for covering a larger number of customers in all deciles. If the mailing depth is optimized through a break-even analysis between revenue and cost, then more accurate and practical evaluation measure needs to be created. Second, the proposed pattern selection algorithm, NPPS, can also be utilized to reduce the lengthy training time of neural network classifiers. But it is necessary to add extra correct patterns to the selected pattern set in order to enhance the overlap region near the decision boundary [5], [9]. The rationale is that "overlap patterns" located on the "wrong" side of the decision boundary cause the MLP training to take a longer time. Since the derivatives of the back-propagated errors are evaluated at those patterns, the derivatives are very small if they are grouped in a narrow region on either side of the decision boundary. By means of adding extra correct patterns, however, the network training converged faster. Third, the current version of NPPS works for classification problems only, thus is not applicable to regression problems. In a regression problem, the patterns located away from others, such as outliers, are less important to learning. Thus, a straightforward idea would be to use the mean (μ) and variance (Σ) of k nearest neighbors' outputs. A pattern having a small value of Σ can be replaced by μ of its neighbors and itself, then these $k+1$ patterns can be replaced by one pattern or their centroid. On the contrary, a pattern having a large value of Σ can be totally eliminated, and its neighbors will be used for the next pattern searching. A similar research was conducted in [24] based on ensemble neural network, but more extended study based on *k* nearest neighbors is still under consideration. Regression NPPS will also be helpful for direct marketing problems with large datasets.

References

[1] Almeida, M. B., Braga, A. and Braga J. P., "SVM-KM: Speeding SVMs Learning with A Priori Cluster Selection and *k*-means," *Proc. of the 6th Brazilian Symposium on Neural Networks*, pp. 162–167, 2000.

[2] Byun, H. and Lee, S., "Applications of Support Vector Machines for Pattern Recognition: A Survey," *International Workshop on Pattern Recognition with Support Vector Machines (SVM2002), Lecture Notes in Computer Science (LNCS 2388)*, Niagara Falls, Canada, pp. 213–236, 2002.

[3] Cheung, K.-W., Kwok, J. K., Law, M. H. and Tsui, K.-C., "Mining Customer Product Rating for Personalized Marketing," *Decision Support Systems*, vol. 35, pp. 231–243, 2003.

[4] Chiu, C., "A Case–Based Customer Classification Approach for Direct Marketing," *Expert Systems with Applications*, vol. 22, pp. 163–168, 2002.

[5] Choi, S. H. and Rockett, P., "The Training of Neural Classifiers with Condensed Dataset," *IEEE Transactions on Systems, Man, and Cybernetics- PART B: Cybernetics*, vol. 32, no. 2, pp. 202–207, 2002.

[6] Coenen, F., Swinnen, G., Vanhoof, K. and Wets, G., “The Improvement of Response Modeling: Combining Rule–Induction and Case–Based Reasoning,” *Expert Systems with Applications*, vol. 18, pp. 307–313, 2000.

[7] Freund, Y. and Schapire, R., “Experiments with a New Boosting Algorithm” *Proc. of the Thirteenth International Conference on Machine Learning*, pp. 148–156, 1996.

[8] Ha, K., Cho, S., and MacLachlan, D., “Response Models Based on Bagging Neural Networks,” submitted.

[9] Hara, K. and Nakayama, K., “A Training Method with Small Computation for Classification,” *Proc. of the IEEE-INNS-ENNS International Joint Conference*, vol. 3, pp. 543–548, 2000.

[10] Haughton, D. and Oulabi, S. “Direct Marketing Modeling with CART and CHAID,” *Journal of Direct Marketing*, vol. 11, no. 4, pp. 42–52, 1997.

[11] Hearst, M. A., Schölkopf, B., Dumais, S., Osuna, E., and Platt, J., “Trends and Controversies - Support Vector Machines,” *IEEE Intelligent Systems*, vol. 13, pp. 18–28, 1997.

[12] Japkowicz, N., “Learning from Imbalanced Data Sets: A Comparison of Various Strategies,” *In AAAI Workshop on Learning from Imbalanced Data Sets*, Menlo Park, CA, AAAI Press, 2000.

[13] Lee, K. K., Gunn, S. R., Harris, C. J, and Reed, P. A. S., “Classification of Imbalanced Data with Transparent Kernels,” *Proc. of INNS-IEEE International Joint Conference on Neural Networks*, pp. 2410–2415, 2001.

[14] Ling, C. X. and Li, C., “Data Mining for Direct Marketing: Problems and Solutions,” *Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pp. 73–79, 1998.

[15] Liu C. L., and Nakagawa M., “Evaluation of Prototype Learning Algorithms for Nearest-Neighbor Classifier in Application to Handwritten Character Recognition,” *Pattern Recognition*, vol. 34, pp. 601–615, 2001.

[16] Lyhyaoui, A., Martinez, M., Mora, I., Vazquez, M., Sancho, J. and Figueiras-Vaidal, A. R., “Sample Selection Via Clustering to Construct Support Vector-Like Classifiers,” *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1474–1481, 1999.

[17] Malthouse, E. C., “Ridge Regression and Direct Marketing Scoring Models,” *Journal of Interactive Marketing*, vol. 13, no. 4, pp. 10–23, 1999.

[18] Malthouse, E. C., “Assessing the Performance of Direct Marketing Models,” *Journal of Interactive Marketing*, vol. 15, no. 1, pp. 49–62, 2001.

[19] Malthouse, E. C., “Performance–Based Variable Selection for Scoring Models,” *Journal of Interactive Marketing*, vol. 16, no. 4, pp. 10–23, 2002.

[20] Platt, J. C. “Fast Training of Support Vector Machines Using Sequential Minimal Optimization,” *Advances in Kernel Methods: Support Vector Machines*, MIT press, Cambridge, MA, pp. 185–208, 1999.

[21] Provost, F. and Fawcett, T., “Analysis and visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions,” *Proc. of 3rd International Conference on Knowledge Discovery and Data Mining,* AAAI press, pp. 43–48, 1997.

[22] SAS Institute Inc., *Enterprise Mining Premier*, 1998.

[23] Schölkopf, B., Burges, C. J. C, and Smola, A. J., *Advances in Kernel Methods: Support Vector Learning*, MIT press, Cambridge, MA, 1999.

[24] Shin, H. J. and Cho, S., “Pattern Selection Using the Bias and Variance of Ensemble,” *Journal of the Korean Institute of Industrial Engineers*, vol. 28, No. 1, pp. 112–127, 2001.

[25] Shin, H. J. and Cho, S., “Pattern Selection For Support Vector Classifiers,” *The 3rd International Conference on Intelligent Data Engineering and Automated Learning (IDEAL), Lecture Notes in Computer Science (LNCS 2412)*, Manchester, UK, pp. 469–474, 2002.

[26] Shin, H. J. and Cho, S., “Fast Pattern Selection for Support Vector Classifiers,” *Proc. of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Lecture Notes in Artificial Intelligence (LNAI 2637)*, Seoul, Korea, pp.376–387, 2003.

[27] Shin, H. J. and Cho, S., “Fast Pattern Selection Algorithm for Support Vector Classifiers: *Time Complexity Analysis*,” *The 4th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL), Lecture Notes in Computer Science (LNCS 2690)*, Hong Kong, China, pp. 1008–1015, 2003.

[28] Shin, H. J. and Cho, S. Z., “How Many Neighbors To Consider in Pattern Pre-selection for Support Vector Classifiers?,” *Proc. of the International Joint Conference on Neural Networks (IJCNN)*, Portland, U.S.A., pp. 565–570, 2003.

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

[30] Viaene, S., Baesens, B., Van Gestel, T., Suykens, J. A. K., Van den Poel, D., Vanthienen, J., De Moor, B. and Dedene, G., “Knowledge Discovery in a Direct Marketing Case using Least Squares Support Vector Machines,” *International Journal of Intelligent Systems*, vol. 16, pp. 1023–1036, 2001.

[31] http://www.the-dma.org/dmef/dmefdset.shtml

[32] http://www.kernel-machines.org/