THE UNIVERSITY OF TEXAS AT SAN ANTONIO, COLLEGE OF BUSINESS

Working Paper SERIES

Date September 19, 2012

WP # 0038MSS-61-2012

A Hierarchical Multiple Kernel Support Vector Machine for Customer Churn Prediction Using Longitudinal Behavioral Data

Zhen-Yu Chen, Zhi-Ping Fan

Department of Management Science and Engineering, School of Business Administration,

Northeastern University, Shenyang 110819, China

Minghe Sun

Department of Management Science and Statistics, College of Business,

The University of Texas at San Antonio, San Antonio, TX 78249-0632, USA

COLLEGE OF BUSINESS KNOWLEDGE FOR A NEW WORLD

ONE UTSA CIRCLE
SAN ANTONIO, TEXAS 78249-0631
210 458-4317 | BUSINESS.UTSA.EDU

A Hierarchical Multiple Kernel Support Vector Machine for Customer Churn Prediction Using Longitudinal Behavioral Data

Zhen-Yu Chen[a,1], Zhi-Ping Fan[a], Minghe Sun[b]

[a]*Department of Management Science and Engineering, School of Business Administration,*
Northeastern University, Shenyang 110819, China

[b]*Department of Management Science and Statistics, College of Business,*
The University of Texas at San Antonio, San Antonio, TX 78249-0632, USA

Abstract:

The availability of abundant data posts a challenge to integrate static customer data and longitudinal behavioral data to improve performance in customer churn prediction. Usually, longitudinal behavioral data are transformed into static data before being included in a prediction model. In this study, a framework with ensemble techniques is presented for customer churn prediction directly using longitudinal behavioral data. A novel approach called the hierarchical multiple kernel support vector machine (H-MK-SVM) is formulated. A three phase training algorithm for the H-MK-SVM is developed, implemented and tested. The H-MK-SVM constructs a classification function by estimating the coefficients of both static and longitudinal behavioral variables in the training process without transformation of the longitudinal behavioral data. The training process of the H-MK-SVM is also a feature selection and time subsequence selection process because the sparse non-zero coefficients correspond to the variables selected. Computational experiments using three real-world databases were conducted. Computational results using multiple criteria measuring performance show that the H-MK-SVM directly using longitudinal behavioral data performs better than currently available classifiers.

Keywords: Data mining; Customer relationship management; Customer churn prediction; Support vector machine; Multiple kernel learning

JEL Classification: C32, C38, C61

[1] Corresponding author. Tel.: +86 24 83871630; Fax: +86 24 23891569.

E-mail address: zychen@mail.neu.edu.cn (Z.-Y. Chen); zpfan@mail.neu.edu.cn (Z.-P. Fan); minghe.sun@utsa.edu (M. Sun).

1. Introduction

In markets with intensive competition, customer relationship management (CRM) is an important business strategy. Business firms use CRM to build long term and profitable relationships with specific customers (Coussement and Van den Poel, 2008b; Ngai *et al*., 2009). An important task of CRM is customer retention. Customer churn is a marketing related term meaning that customers leave or reduce the amount of purchase from the firm. Customer churn prediction aims at identifying the customers who are prone to switch at least some of their purchases from the firm to competitors (Buckinx and Van den Poel, 2005; Coussement and Van den Poel, 2008b). Usually, new customer acquisition results in higher costs and probably lower profits than customer retention (Buckinx and Van den Poel, 2005; Coussement and Van den Poel, 2008a; Zorn *et al*., 2010). Therefore, many business firms use customer churn prediction to identify customers who are likely to churn. Measures can be taken to assist them in improving intervention strategies to convince these customers to stay and to prevent the loss of businesses (Zorn *et al*., 2010).

Longitudinal behavioral data are widely available in databases of business firms. How to use the longitudinal behavioral data to improve customer churn prediction is a challenge to researchers. In some methods, longitudinal behavioral data are transformed into static data through aggregation or rectangularization. In this study, frameworks for customer churn prediction are developed. In the framework using ensemble techniques, a novel data mining technique called hierarchical multiple kernel support vector machine (H-MK-SVM) is proposed to model both static and longitudinal behavioral data. A three phase algorithm is developed and implemented to train the H-MK-SVM. The H-MK-SVM constructs a classification function by estimating the coefficients of both static and longitudinal behavioral variables in the training process without transformation of the longitudinal behavioral data. Because of the sparse nature of the coefficients, the H-MK-SVM supports adaptive feature selection and time subsequence selection. Furthermore, the H-MK-SVM can benefit customer churn prediction performance in both contractual and non-contractual settings.

This paper is organized as follows. The next section reviews previous work. Section 3 describes the frameworks for customer churn prediction using longitudinal behavioral data. Section 4 presents the fundamentals of the support vector machine (SVM) and the multiple kernel SVM (MK-SVM). The model formulation and the three phase training algorithm of the H-MK-SVM are presented in Section 5. The computational experiments are described in Section 6 and the computational results are reported in Section 7.

Conclusions and further remarks are given in Section 8.

2. Previous Work

Recently, the topic of customer churn prediction has been discussed extensively in a number of domains such as telecommunications (Kisioglu and Topcu, 2010; Tsai and Lu, 2009; Verbeke *et al*., 2011), retail markets (Baesens *et al*., 2004; Buckinx and Van den Poel, 2005), subscription management (Burez and Van den Poel, 2007; Coussement and Van den Poel, 2008a), financial services (Glady *et al*., 2009) and electronic commerce (Yu *et al*. 2010). Many data mining techniques have been successfully applied in customer churn prediction. These techniques include artificial neural networks (ANN) (Tsai and Lu, 2009), decision trees (Qi *et al*, 2009), Bayesian networks (Baesens *et al*., 2004; Kisioglu and Topcu, 2010), logistic regression (Buckinx and Van den Poel, 2005; Burez and Van den Poel, 2007), AdaBoosting (Glady *et al*., 2009), random forest (Buckinx and Van den Poel, 2005; Burez and Van den Poel, 2007), the proportional hazard model (Van den Poel and Larivière, 2004) and SVMs. Lessmann and Voß (2008) gave a detailed review on this topic.

SVMs have strong theoretical foundations and the SVM approach is a state-of-the-art machine leaning method. SVMs have been widely used in many areas such as pattern recognition and data mining (Schölkopf and Smolla, 2002; Vapnik, 1995, 1998) and have achieved successes in customer churn prediction (Coussement and Van den Poel, 2008a; Lessmann and Voß, 2008, 2009; Verbeke *et al*., 2011).

Demographic and behavioral attributes have been widely used for customer churn prediction (Buckinx and Van den Poel, 2005). Customer demographic data are static, while longitudinal behavioral data are temporal. Customer demographic data can be directly obtained from the data warehouse of the business firm, while the longitudinal behavioral data of individual customers are usually separately stored in transactional databases (Cao, 2010; Cao and Yu, 2009; Chen *et al*., 2005). Three typical customer behavioral variables are recency, frequency and monetary variables. Recency is the time period since the customer's last purchase to the time of data collection; frequency is the number of purchases made by individual customers within a specified time period; and the monetary variable represents the amount of money a customer spent during a specified time period (Chen *et al*., 2005; Buckinx and Van den Poel, 2005).

The temporal nature of customer longitudinal behavioral data is usually neglected in customer churn prediction (Eichinger *et al*., 2007; Orsenigo and Vercellis, 2010; Prinzie and Van den Poel, 2006a). Usually, the longitudinal behavioral variables are transformed into static variables through aggregation or rectangularization before being included in the prediction model (Cao, 2010; Cao and Yu, 2009; Eichinger *et al*., 2007; Orsenigo and Vercellis, 2010; Prinzie and Van den Poel, 2006a). The transformation results in the loss of temporal development information with potential discriminative ability. For example, the changing values of recency, frequency, and monetary variables between different time periods may have better customer churn predictive ability than the values of these variables in a fixed time period or their averages in a series of time periods. Furthermore, traditional customer churn prediction models usually use static data with two dimensions. Relatively few models proposed in the literature capture temporal information in longitudinal behavioral data with three dimensions (Prinzie and Van den Poel, 2006a).

Customer longitudinal behavioral modeling has been studied in the fields of financial distress prediction (Sun *et al*., 2011) and customer acquisition analysis (Prinzie and Van den Poel, 2006b, 2007, 2009). However, relatively little research has focused on longitudinal behavioral data for customer churn prediction. Prinzie and Van den Poel (2006a) incorporated static and a one dimensional temporal variable into a customer churn prediction model. They used a sequence alignment approach to model the temporal variable, clustered customers on the sequential dimension, and incorporated the clustering information into a traditional classification model. This approach is limited to the use of temporal variables with one dimension. Eichinger *et al*. (2007) proposed a classification approach using sequence mining combined with a decision tree for modeling customer event sequence. Orsenigo and Vercellis (2010) proposed a two stage strategy for multivariate time series classification. In the first stage, a rectangularization strategy using a fixed cardinality warping path is proposed to transform multivariate time series into a rectangular table. In the second stage, a temporal discrete SVM is used for classification. They applied this approach to telecommunication customer churn prediction. Huang *et al*. (2010) reformulated the rectangular table by adopting each time element of the temporal vector as a predictor. Long historical behavioral time series may lead to the “dimension curse” of the prediction models in this approach. The proportional hazard model (Cox) was extended for customer churn prediction using longitudinal behavioral variables (Van den Poel and Larivière, 2004). The coefficients of static and longitudinal behavioral variables are estimated, and then a threshold of the hazard is set for predicting churners.

The above studies made great contributions to customer churn prediction using longitudinal behavioral data. However, they still have limitations. A transformation may result in the loss of potentially useful structural information embodied in the longitudinal behavioral data or increase the computational cost. Moreover, customer churn prediction involves predictors with multiplex data such as static data, temporal data, event sequential data, textural data, and so on (Coussement and Van den Poel, 2008b; Eichinger *et al.* , 2007; Orsenigo and Vercellis, 2010; Prinzie and Van den Poel, 2006a). Relatively little research has focused on simultaneously modeling multiplex data. In addition, the availability of expanded customer longitudinal behavioral and demographic data raises a new and important question about which variables to use and which time subsequence in the longitudinal behavioral variables to consider (Dekimpe and Hanssens, 2000).

The H-MK-SVM approach proposed in this study is the first attempt of using static and longitudinal behavioral data in a direct manner. The computational results show that this approach outperforms other methods.

3. Frameworks for Customer Churn Prediction

In this section, three frameworks for customer churn prediction using longitudinal behavioral data are discussed. The longitudinal behavioral attributes are used in different ways in these frameworks.

Within the CRM context, demographic and transactional data recorded in the data warehouses of business firms have been widely used for customer churn prediction. Customer demographic and transactional data are organized in terms of entity relationship in relational databases (Cao, 2010; Eichinger *et al*., 2007). Each customer is treated as an observation and n is used to represent the number of observations in the demographic dataset.

Demographic data can be directly used as static attributes after simple data preprocessing such as feature selection and data cleaning for missing values. In a dataset with m_1 static variables, the static attributes of a customer i is usually represented by the input vector $\mathbf{s}_i = \{s_{ij} \mid j = 1,\cdots,m_1\}$.

Transactional data are transformed into longitudinal behavioral data, i.e., customer-centered multivariate time series of fixed length. The number of longitudinal behavioral attributes is represented by M and the number of time points in the longitudinal behavioral variables is represented by T . The longitudinal behavioral data are represented by a three-dimensional matrix $\{\mathbf{b}_i \mid i = 1,\cdots,n\}$. Each $\mathbf{b}_i = \{b_{i\tilde{j}t} \mid \tilde{j} = 1,\cdots,M;\ t = 1,\cdots,T\}$ is a rectangular matrix. The class label $y_i \in \{+1,-1\}$ indicates the status of customer i , i.e., $y_i = 1$ if customer i has churned and $y_i = -1$ otherwise.

3.1 The standard framework for customer churn prediction

In customer churn prediction, an aggregation strategy such as weighted averages is usually used to transform longitudinal behavioral data into static data, called transformed static data in this study (Eichinger *et al.*, 2007; Orsenigo and Vercellis, 2010; Prinzie and Van den Poel, 2006a). The transformed static data are represented by a matrix $\{\mathbf{ts}_i \mid i=1,\cdots,n\}$. The input vector $\mathbf{ts}_i = \{ts_{ij} \mid j=1,\cdots,m_2\}$ represents the transformed static attributes of customer i, where m_2 is the number of the transformed static attributes.

Customer churn prediction with only static variables is a standard binary classification problem. The dataset with only static variables is represented by a two-dimensional matrix $\{\mathbf{x}_i,\ i \in \{1,\cdots,n\}\}$, where $\mathbf{x}_i = [\mathbf{s}_i,\ \mathbf{ts}_i]_m$ is the input vector of observation i with $m = m_1 + m_2$. The standard framework for customer churn prediction is illustrated in Fig. 1a.

3.2 A framework with feature construction techniques

A variety of binary classification methods can be used for customer churn prediction with only static variables (Lessmann and Voß, 2008). Time series classification methods can be used for customer churn prediction with longitudinal behavioral variables (Orsenigo and Vercellis, 2010). Feature construction techniques usually used for time series classification derive a rectangular matrix representation of the time series first and then apply an existing classifier. The proportional hazard model (Cox) with longitudinal behavioral variables (Van den Poel and Larivière, 2004) and the time windows techniques (Huang *et al.*, 2010) are examples of feature construction techniques.

A framework with feature construction techniques is proposed to use these existing classifiers for customer churn prediction using longitudinal behavioral data as illustrated in Fig. 1b. In this framework, the longitudinal behavioral data $\{\mathbf{b}_i \mid i=1,\cdots,n\}$ are transformed into a standard rectangular matrix $\{\mathbf{tb}_i \mid i=1,\cdots,n\}$, where $\mathbf{tb}_i = [\mathbf{b}_{i,1}\ \mathbf{b}_{i,2}\ \cdots\ \mathbf{b}_{i,M}]_{1\times(M\times T)}$ and each $\mathbf{b}_{i,\tilde{j}} = \{b_{i\tilde{j}t} \mid t=1,\cdots,T\}$ is a vector. Each time point of the longitudinal behavioral variable $b_{i\tilde{j}t}$ is used as an attribute. For example, data with $M=3$ longitudinal behavioral variables and $T=12$ time points are treated as data with $M\times T=36$ (3×12) attributes when the entire time series is used. Hence, the number of features in this framework is larger than those in the standard framework and in the framework with ensemble techniques discussed in the following. In this framework, the input vector of observation i is $\tilde{\mathbf{x}}_i = [\mathbf{s}_i,\ \mathbf{tb}_i]_{1\times(m_1+M\times T)}$.

3.3 A framework with ensemble techniques

The proposed framework with ensemble techniques for customer churn prediction with the

H-MK-SVM using both static and longitudinal behavioral data is illustrated in Fig. 1c. In this framework, longitudinal behavioral data are directly used as input of the time series classifier without any aggregation as commonly used in the standard framework or rectangularization as commonly used by time series classifiers. At the same time, static data are used as input of a standard classifier. An ensemble classifier then combines the results of the time series classifier and the standard classifier. In this framework, the input vector $\mathbf{x}_i$ for customer i is a composite vector containing two parts, the static variables $\mathbf{s}_i$ and the longitudinal behavioral variables $\mathbf{b}_i$, i.e., $\mathbf{x}_i = (\mathbf{s}_i, \mathbf{b}_i)$. Because existing classifiers cannot be directly used in this framework, a novel method called the H-MK-SVM is developed and used for this purpose.

Figure 1 approximately here

4. Support Vector Machines and Multiple Kernel Support Vector Machines

The SVM and the MK-SVM, as the theoretical foundations of the H-MK-SVM, are briefly discussed in the following. The use of both static data and transformed longitudinal behavioral data in the SVM and the MK-SVM is also discussed.

4.1 Support vector machines

The basics of SVMs are introduced in this section. Vapnik (1995, 1998) and Schölkopf and Smolla (2002) gave more details. In a binary classification problem, a training dataset $G = \{(\mathbf{x}_{1,} y_1), \cdots, (\mathbf{x}_{n,} y_n)\}$ is available. A SVM constructs an optimal hyperplane in a high dimensional feature space

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \cdot \phi(\mathbf{x}) + b \qquad (1)$$

where $\phi(\mathbf{x}) : \Re^m \mapsto \Re^{m'}$, with m' □ m, is a nonlinear function which maps an input vector $\mathbf{x} \in \Re^m$ from the input space $\Re^m$ to $\phi(\mathbf{x}) \in \Re^{m'}$ in a higher dimensional feature space $\Re^{m'}$.

In the hyperplane (1), $\mathbf{w}$ and b are the coefficients estimated using the available training dataset. According to structural risk minimization principles (Vapnik, 1995, 1998), $\mathbf{w}$ and b are estimated by solving the following quadratic program

$$\min \quad J(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \qquad (2)$$

$$\text{s.t.} \quad y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \qquad i = 1, \cdots, n \qquad (3)$$

$$\xi_i \geq 0 \qquad i = 1, \cdots, n \qquad (4)$$

where C in (2) is a user defined regularization parameter.

The Lagrangian of the quadratic program in (2)-(4) is

$$L(\mathbf{w},\boldsymbol{\xi},b;\boldsymbol{\alpha}) = J(\mathbf{w},\boldsymbol{\xi},b) - \sum_{i=1}^{n} \alpha_i \left\{ y_i(\mathbf{w}^T \phi(\boldsymbol{\xi}_i) + b) + \xi_i - 1 \right\} - \sum_{i=1}^{n} \mu_i \xi_i \,. \tag{5}$$

Taking partial derivatives of $L(\mathbf{w},\boldsymbol{\xi},b;\boldsymbol{\alpha})$ with respect to the primal variables and substituting the results into $L(\mathbf{w},\boldsymbol{\xi},b;\boldsymbol{\alpha})$ in (5) lead to the dual formulation of the SVM

$$\max \quad L(\mathbf{w},\boldsymbol{\xi},b;\boldsymbol{\alpha}) = \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) \right\} \tag{6}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0 \tag{7}$$

$$0 \le \alpha_i \le C \qquad\qquad i = 1, \cdots, n \,. \tag{8}$$

In (6)-(8), α_i is the Lagrange multiplier of observation i, and $k(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function which defines an inner product $\langle \phi(\mathbf{x}), \phi(\mathbf{x}) \rangle$. When convenient, $\boldsymbol{\alpha}$ is used to denote the vector of all Lagrange multipliers. A commonly used kernel function is the Gaussian, also called the radial basis, function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{\sigma^2} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2 \right), \tag{9}$$

where $1/\sigma^2$ is the parameter of the kernel. The relationship between the vectors $\mathbf{w}$ and $\boldsymbol{\alpha}$ is

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{x}_i) \,. \tag{10}$$

The dual in (6)-(8) is also a quadratic program and is usually easier to solve than the primal in (2) -(4). The training process of the SVM is the solution process of the dual in (6)-(8). After the SVM is trained, the values of $\boldsymbol{\alpha}$ are determined. The resulting classification function in the Lagrange multipliers is

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i \, y_i k(\mathbf{x}_i, \mathbf{x}) + b \right). \tag{11}$$

The SVM for customer churn prediction is depicted graphically in Fig. 2. In Fig. 2a, the SVM is used for the standard framework of customer churn prediction in Fig. 1a. The static attributes $\mathbf{s}_i$ and transformed static attributes $\mathbf{ts}_i$ are used as input of the SVM in this framework. In Fig. 2b, the SVM is used for the framework with feature construction techniques as shown in Fig. 1b. The static attributes $\mathbf{s}_i$ and the transformed longitudinal behavioral attributes $\mathbf{tb}_i$ are used as input of the SVM in this framework.

Figure 2 approximately here

4.2 Multiple kernel support vector machines

The generalization performance of the SVM is sensitive to the kernel function, the kernel parameter

$1/\sigma^2$ (if the Gaussian function is used) and the regularization parameter C. Hence, selecting a good kernel function is important in an application.

Currently, multiple kernel learning is a popular topic in kernel methods. A MK-SVM uses a combination of some basic kernels to adaptively approximate an optimal kernel (Chen *et al.*, 2007, 2011; Sonnenburg *et al.*, 2006; Gönen and Alpaydin, 2011). The existing algorithms of the MK-SVM can be classified as one-step methods and two-step methods (Gönen and Alpaydin, 2011). One-step methods include several standard reformulations of the MK-SVM such as the quadratically constrained quadratic programming, semidefinite programming, and second-order cone programming (Bach *et al.*, 2004; Lanckrient *et al.*, 2004a). Because of the computational complexity of the one-step methods, two-step methods have been proposed for fast implementation purpose (Chapelle *et al.*, 2002; Chen *et al.*, 2007, 2011; Gunn and Kandola, 2002; Keerthi *et al.*, 2007; Rakotomamonjy *et al.*, 2008; Sonnenburg *et al.*, 2006). Chen *et al.* (2007) used a two phase algorithm while Chapelle *et al.* (2002) and Keerthi *et al.* (2007) used a gradient decent algorithm to optimize the kernel parameters.

The two-step method of the MK-SVM presented here was proposed by Chen *et al.* (2007). It introduces a single feature kernel in the MK-SVM for simultaneous feature selection and pattern classification. When m basic kernel functions are used, a linear combination of the basic kernels is

$$k(\mathbf{x}_i,\mathbf{x}_j)=\sum_{d=1}^{m}\beta_d k_d(\mathbf{x}_i,\mathbf{x}_j), \tag{12}$$

where $\beta_d \ge 0$ is the weight of the basic kernel $k_d(\mathbf{x}_i,\mathbf{x}_j)$. When convenient, $\boldsymbol{\beta}$ is used to denote the vector of all weights of the basic kernels. When a single feature basic kernel is introduced in the MK-SVM, the linear combination is of the form

$$k(\mathbf{x}_i,\mathbf{x}_j)=\sum_{d=1}^{m}\beta_d k_d(x_{i,d},x_{j,d}), \qquad \beta_d \ge 0 \tag{13}$$

where $x_{i,d}$ denotes the dth component of the input vector $\mathbf{x}_i$. As in (12), β_d in (13) is the coefficient representing the weight on the single feature basic kernel $k_d(x_{i,d},x_{j,d})$.

When the multiple kernel function (13) is used in (6)-(8), the formulation of the MK-SVM becomes

$$\max \qquad L(\boldsymbol{\alpha},\boldsymbol{\beta})=\left\{\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\,\alpha_j y_i y_j\sum_{d=1}^{m}\beta_d k_d(x_{i,d},x_d)\right\} \tag{14}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i\alpha_i=0 \tag{15}$$

$$0 \le \alpha_i \le C \qquad i = 1,\cdots,n \tag{16}$$

$$\beta_d \ge 0 \qquad d = 1,\cdots,m\,. \tag{17}$$

Chen *et al.* (2007) proposed a two phase iterative scheme which decomposes the quadratic program in (14)-(17) into two sub-problems, a quadratic program and a linear program. The two sub-problems are solved iteratively by standard solution methods. The two phase iterative scheme is summarized as follows.

In phase 1, the values of $\boldsymbol{\beta}$ are fixed at initial values in the first iteration or at the values obtained in phase 2 of the previous iteration in subsequent iterations, and the values of the Lagrange multipliers $\boldsymbol{\alpha}$ are obtained by solving the following quadratic program

$$\max_{\boldsymbol{\alpha}} \quad \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \sum_{d=1}^{m} \beta_d k_d (x_{i,d}, x_d) \right\} \tag{18}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0 \tag{19}$$

$$0 \le \alpha_i \le C \qquad i = 1,\cdots,n\,. \tag{20}$$

In phase 2, the values of $\boldsymbol{\alpha}$ are fixed at the values obtained in phase 1 and a shrinkage strategy is used to select features by solving a linear program. In the linear program, a L_1-norm soft margin error function is minimized to obtain the values of $\boldsymbol{\beta}$. Any attribute d corresponding to $\beta_d = 0$ is discarded and all others are kept as selected features. The linear program is

$$\min \quad J(\boldsymbol{\beta}, \boldsymbol{\xi}) = \sum_{d=1}^{m} \beta_d + \lambda \sum_{i=1}^{n} \xi_i \tag{21}$$

$$\text{s.t.} \quad y_i \left(\sum_{d=1}^{m} \beta_d \sum_{j=1}^{n} \alpha_j y_j k_d (x_{i,d}, x_{j,d}) + b \right) \ge 1 - \xi_i \qquad i = 1,\cdots,n \tag{22}$$

$$\xi_i \ge 0 \qquad i = 1,\cdots,n \tag{23}$$

$$\beta_d \ge 0 \qquad d = 1,\cdots,m\,. \tag{24}$$

In (21), λ is a regularization parameter that controls the sparseness of $\boldsymbol{\beta}$. The L_1-norm soft margin error function (21) and the constraints (22)-(24) are derived from the primal problem (2)-(4) when the multiple kernel (13) and $\mathbf{w}$ in (10) are used. The dual of the linear program in (21)-(24) is solved in the implementation because the dual is easier to solve.

The classification function obtained with the MK-SVM is

$$f(\mathbf{x})=\mathrm{sgn}\left(\sum_{i=1}^{n}\alpha_{\mathrm{i}}y_i\sum_{d=1}^{m}\beta_d k_d(x_{i,d},x_d)+b\right), \tag{25}$$

where the Lagrange multipliers $\boldsymbol{\alpha}$ and feature coefficients $\boldsymbol{\beta}$ are learned by solving the quadratic and linear programs in the two phase iterative scheme.

The classification function can also be written as

$$f(\mathbf{x})=\mathrm{sgn}\left\{\sum_{d=1}^{m}\beta_d f_d(\mathbf{x})+b\right\}=\mathrm{sgn}\left\{\sum_{d=1}^{m}\beta_d\sum_{i=1}^{n}\alpha_{\mathrm{i}}y_i k_d(x_{i,d},x_d)+b\right\}. \tag{26}$$

Each $f_d(\mathbf{x})=\sum_{i=1}^{n}\alpha_{\mathrm{i}}y_i k_d(x_{i,d},x_d)$ is a sub-function of feature d. The classification function $f(\mathbf{x})$ is a linear combination of the sub-functions $f_d(\mathbf{x})$ plus a bias b.

The MK-SVM for customer churn prediction is depicted graphically in Fig. 3. In Fig. 3a, the static attributes $\mathbf{s}_i$ and transformed static attributes $\mathbf{ts}_i$ are used as input of the MK-SVM for the standard framework of customer churn prediction in Fig. 1a. In Fig. 3b, the static attributes $\mathbf{s}_i$ and the transformed longitudinal behavioral attributes $\mathbf{tb}_i$ are used as input of the MK-SVM for the framework with feature construction techniques in Fig. 1b.

Figure 3 approximately here

5. The Hierarchical Multiple Kernel Support Vector Machine and the Training Algorithm

The H-MK-SVM is formulated and the three phase training algorithm is developed in this section. The decomposition of the resulting classification function is also discussed.

5.1 Model Formulation

In the H-MK-SVM, two types of multiple kernels are used for the static and the longitudinal behavioral attributes, respectively. For the static attributes, the single feature kernel (27) is used (Chen *et al.*, 2007)

$$k(\mathbf{s}_i,\mathbf{s}_{\tilde{i}})=\sum_{j=1}^{m_1}\beta_j k_j(s_{i,j},s_{\tilde{i},j}), \qquad \beta_j\geq 0. \tag{27}$$

In (27), $k_j(s_{i,j},s_{\tilde{i},j})$ is the basic kernel mapping dimension j of the static attribute $\mathbf{s}_i$ and β_j is the coefficient of $k_j(s_{i,j},s_{\tilde{i},j})$. For the longitudinal behavioral attributes, the single feature kernel (28) is used

$$k(\mathbf{b}_i,\mathbf{b}_{\tilde{i}})=\sum_{\tilde{j}=1}^{M}\tilde{\beta}_{\tilde{j}}\sum_{t=1}^{T}\gamma_t k_{\tilde{j},t}(b_{i,\tilde{j},t},b_{\tilde{i},\tilde{j},t}). \tag{28}$$

In (28), $\tilde{\beta}_{\tilde{j}}$ is the coefficient of dimension $\tilde{j}$ of the temporal attribute input vector $\mathbf{b}_i$, $k_{\tilde{j},t}(b_{i,\tilde{j},t}, b_{\tilde{i},\tilde{j},t})$ is the basic kernel mapping time point t of the time series $\mathbf{b}_{i,\tilde{j}}$ and γ_t is the weight of $k_{\tilde{j},t}(b_{i,\tilde{j},t}, b_{\tilde{i},\tilde{j},t})$. When convenient, $\tilde{\boldsymbol{\beta}}$ is used to denote the vector of all the coefficients and $\boldsymbol{\gamma}$ is used to denote the vector of all the weights of the basic kernels in (28).

When multiple kernels (27) and (28) are used, the formulation of the H-MK-SVM becomes

$$\max_{\boldsymbol{\beta},\tilde{\boldsymbol{\beta}},\boldsymbol{\gamma}} \max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,\tilde{i}=1}^{n} \alpha_i \alpha_{\tilde{i}} y_i y_{\tilde{i}} \left(\sum_{j=1}^{m_1} \beta_j k_j(s_{i,j}, s_{\tilde{i},j}) + \sum_{\tilde{j}=1}^{M} \tilde{\beta}_{\tilde{j}} \sum_{t=1}^{T} \gamma_t k_{\tilde{j},t}(b_{i,\tilde{j},t}, b_{\tilde{i},\tilde{j},t}) \right) \tag{29}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0 \tag{30}$$

$$0 \le \alpha_i \le C \qquad i = 1,\cdots,n \tag{31}$$

$$\beta_j \ge 0 \qquad j = 1,\cdots,m_1 \tag{32}$$

$$\tilde{\beta}_{\tilde{j}} \ge 0 \qquad \tilde{j} = 1,\cdots,M \tag{33}$$

$$\gamma_t \ge 0 \qquad t = 1,\cdots,T\,. \tag{34}$$

5.2 The three phase training algorithm

The H-MK-SVM is trained through a three phase training algorithm. The H-MK-SVM model in (29) -(34) is decomposed into three sub-problems, a quadratic program and two linear programs that can all be solved with standard solution algorithms.

In phase 1, $\boldsymbol{\beta}$, $\tilde{\boldsymbol{\beta}}$ and $\boldsymbol{\gamma}$ are fixed and the values of the Lagrange multipliers $\boldsymbol{\alpha}$ are updated by solving a quadratic program that is similar to a standard SVM. The quadratic program is the dual of the primal problem (2)-(4) but using the multiple kernels (27) and (28),

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,\tilde{i}=1}^{n} \alpha_i \alpha_{\tilde{i}} y_i y_{\tilde{i}} \left(\sum_{j=1}^{m_1} \beta_j k(s_{i,j}, s_{\tilde{i},j}) + \sum_{\tilde{j}=1}^{M} \tilde{\beta}_{\tilde{j}} \sum_{t=1}^{T} \gamma_t\, k(b_{i,\tilde{j},t}, b_{\tilde{i},\tilde{j},t}) \right) \tag{35}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0 \tag{36}$$

$$0 \le \alpha_i \le C \qquad i = 1,\cdots,n \tag{37}$$

$$\gamma_t \ge 0 \qquad t = 1,\cdots,T\,. \tag{38}$$

In phase 2, the values of $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ are fixed and the values of $\boldsymbol{\beta}$ and $\tilde{\boldsymbol{\beta}}$ are updated by minimizing a L_1-norm soft margin error function in the following linear program

$$\min_{\boldsymbol{\beta},\tilde{\boldsymbol{\beta}},\boldsymbol{\xi}} \quad \sum_{j=1}^{m_1}\beta_j + \sum_{\tilde{j}=1}^{M}\tilde{\beta}_{\tilde{j}} + \lambda\sum_{i=1}^{n}\xi_i \tag{39}$$

$$\text{s.t.} \quad y_i\left\{\sum_{\tilde{i}=1}^{n}\alpha_{\tilde{i}}\, y_{\tilde{i}}\left(\sum_{j=1}^{m_1}\beta_j k(s_{i,j},s_{\tilde{i},j}) + \sum_{\tilde{j}=1}^{M}\tilde{\beta}_{\tilde{j}}\sum_{t=1}^{T}\gamma_t\, k(b_{i,\tilde{j},t},b_{\tilde{i},\tilde{j},t})\right)+\tilde{b}\right\} \geq 1-\xi \tag{40}$$

$$\xi_i \geq 0 \qquad i=1,\cdots,n \tag{41}$$

$$\beta_j \geq 0 \qquad j=1,\cdots,m_1 \tag{42}$$

$$\tilde{\beta}_{\tilde{j}} \geq 0 \qquad \tilde{j}=1,\cdots,M\ . \tag{43}$$

The L_1-norm soft margin error function (39) and the constraints (40)-(43) are derived from the primal problem (2)-(4) when multiple kernels (27) and (28) are used. Demiriz *et al*. (2002) proved that minimizing the L_1-norm soft margin error function directly optimizes a generalization error bound. Moreover, this method can enforce sparseness of the coefficients $\boldsymbol{\beta}$ and $\tilde{\boldsymbol{\beta}}$. In the implementation, the dual of this linear program is actually solved because the dual is easier to solve than the primal.

In phase 3, the values of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\tilde{\boldsymbol{\beta}}$ are fixed and the values of $\boldsymbol{\gamma}$ are updated by minimizing the L_1-norm soft margin error function in the following linear program. The linear program is also derived from the primal problem (2)-(4) using multiple kernels (27) and (28). It enforces sparseness of the coefficients $\boldsymbol{\gamma}$. The linear program is in (44)-(47) in the following but its dual is actually solved

$$\min_{\boldsymbol{\gamma},\boldsymbol{\xi}} \quad \sum_{t=1}^{T}\gamma_t + \tilde{\lambda}\sum_{i=1}^{n}\xi_i \tag{44}$$

$$\text{s.t.} \quad y_i\left\{\sum_{\tilde{i}=1}^{n}\alpha_{\tilde{i}}\, y_{\tilde{i}}\left(\sum_{j=1}^{m_1}\beta_j k(s_{i,j},s_{\tilde{i},j}) + \sum_{\tilde{j}=1}^{M}\tilde{\beta}_{\tilde{j}}\sum_{t=1}^{T}\gamma_t\, k(b_{i,\tilde{j},t},b_{\tilde{i},\tilde{j},t})\right)+\tilde{b}\right\} \geq 1-\xi_i \tag{45}$$

$$\xi_i \geq 0 \qquad i=1,\cdots,n \tag{46}$$

$$\gamma_t \geq 0 \qquad t=1,\cdots,T\ . \tag{47}$$

The Three Phase Training Algorithm

Input: Static data $\mathbf{s}$ and longitudinal behavioral data $\mathbf{b}$.

Initialization: Initialize the coefficients $\boldsymbol{\beta}^{(0)} = \mathbf{I}_{1\times m_1}$, $\tilde{\boldsymbol{\beta}}^{(0)} = \mathbf{I}_{1\times M}$, $\boldsymbol{\gamma}^{(0)} = \mathbf{I}_{1\times T}$.

While not converging

```
  For $\tilde{t}=1$ to MaxIteration
    Phase 1: Find the Lagrange coefficients $\boldsymbol{\alpha}^{(\tilde{t})}$ using fixed $\boldsymbol{\beta}^{(\tilde{t}-1)}$, $\tilde{\boldsymbol{\beta}}^{(\tilde{t}-1)}$ and $\boldsymbol{\gamma}^{(\tilde{t}-1)}$ by solving the quadratic program in (35)-(38).
    Phase 2: Find the static and longitudinal behavioral attribute coefficients $\boldsymbol{\beta}^{(\tilde{t})}$ and $\tilde{\boldsymbol{\beta}}^{(\tilde{t})}$ using the Lagrange multipliers $\boldsymbol{\alpha}^{(\tilde{t})}$ obtained in Phase 1 and fixed $\boldsymbol{\gamma}^{(\tilde{t}-1)}$ by solving the dual of the linear program in (39)-(43).
    Phase 3: Find the time series coefficients $\boldsymbol{\gamma}^{(\tilde{t})}$ using the Lagrange multipliers $\boldsymbol{\alpha}^{(\tilde{t})}$ and the static and longitudinal behavioral attribute coefficients $\boldsymbol{\beta}^{(\tilde{t})}$ and $\tilde{\boldsymbol{\beta}}^{(\tilde{t})}$ found in Phases 1 and 2 by solving the dual of the linear program in (44)-(47).
  End For
End While
Output
```

The H-MK-SVM model in (29)-(34) has three blocks of variables, i.e., the parameters $\boldsymbol{\alpha}$, $\{\boldsymbol{\beta}, \tilde{\boldsymbol{\beta}}\}$ and $\boldsymbol{\gamma}$. These parameters are learned through the three phase training algorithm. Because the parameters obtained are sparse, the training process is also a feature selection process. In each phase of the three phase training algorithm, a convex program is optimized with respect to one block while the other blocks are fixed. Besides the quadratic program in phase 1 which is a standard SVM, the other two convex programs directly optimize a generalization error bound (Demiriz *et al.*, 2002). The three phase training algorithm iterates among these three phases to obtain a final solution by iteratively optimizing the same primal problem, and thus can be viewed as an application of the block coordinate decent method. The convergence property of the block coordinate decent method for convex differentiable minimization with bounded constraints can be found in Tseng (2001) and the references therein. The convergence of the algorithm is guaranteed in a few iterations. Practically, the solution obtained in a one iteration update is usually sufficient. Similar phenomena are observed with other iterative algorithms such as sparse kernel (Gunn and Kandola, 2002), structured SVM (Lee *et al.*, 2006) and component selection and smoothing operator (CSSO) (Lin and Zhang, 2006). This one iteration update approach substantially reduces the computational time needed.

Efficient linear and quadratic programming solution techniques can be used in the three phase training algorithm to speed up the training of the H-MK-SVM. For example, the parallel and distributed algorithms of the SVM (Woodsend and Gondzio, 2009) can be used in phase 1 to greatly reduce the computational complexity of the H-MK-SVM.

The output of the three phase training algorithm is the classification function

$$f(\mathbf{s},\mathbf{b})=\operatorname{sgn}\left\{\sum_{i=1}^{n}\alpha_i\, y_i\left(\sum_{j=1}^{m_1}\beta_j k(s_{i,j},s_j)+\sum_{\tilde{j}=1}^{M}\tilde{\beta}_{\tilde{j}}\sum_{t=1}^{T}\gamma_t\, k(b_{i,\tilde{j},t},b_{\tilde{j},t})\right)+\tilde{b}\right\}. \quad (48)$$

The bias $\tilde{b}$ in (48) is computed as follows

$$\tilde{b}=y_{\tilde{i}}-\sum_{i=1}^{n}\alpha_i\, y_i\left(\sum_{j=1}^{m_1}\beta_j k(s_{i,j},s_{\tilde{i},j})+\sum_{\tilde{j}=1}^{M}\tilde{\beta}_{\tilde{j}}\sum_{t=1}^{T}\gamma_t\, k(b_{i,\tilde{j},t},b_{\tilde{i},\tilde{j},t})\right)\text{ for }\alpha_{\tilde{i}}\in(0,C). \quad (49)$$

5.3 The decomposition of the classification function

The classification function (48) can be decomposed into three parts as shown in (50), i.e., the output of the static data $f_1(\mathbf{s})$, the output of the temporal data $f_2(\mathbf{b})$, and a bias $\tilde{b}$ in (49). Both $f_1(\mathbf{s})$ and $f_2(\mathbf{b})$ can be viewed as sub-classifiers.

$$f(\mathbf{s},\mathbf{b})=\operatorname{sgn}(f_1(\mathbf{s})+f_2(\mathbf{b})+\tilde{b})=\operatorname{sgn}\left\{\sum_{i=1}^{n}\alpha_i\, y_i\left(\sum_{j=1}^{m}\beta_j k(s_{i,j},s_j)+\sum_{\tilde{j}=1}^{M}\tilde{\beta}_{\tilde{j}}\sum_{t=1}^{T}\gamma_t\, k(b_{i,\tilde{j},t},b_{\tilde{j},t})\right)+\tilde{b}\right\} \quad (50)$$

The H-MK-SVM uses hierarchical multiple kernels and is trained hierarchically. There are three levels in the hierarchy. On the first level, the Lagrange multipliers $\boldsymbol{\alpha}$ are learned to obtain the support vectors. On the second level, the coefficients $\boldsymbol{\beta}$ and $\tilde{\boldsymbol{\beta}}$ are learned for the selection of the static and longitudinal behavioral attributes, respectively. On the third level, the coefficients $\boldsymbol{\gamma}$ are learned to select time subsequence in the selected longitudinal behavioral attributes. Hence the model is called the hierarchical MK-SVM (H-MK-SVM).

Schematically, the H-MK-SVM for customer churn prediction using longitudinal behavioral data is shown in Fig. 4. In the H-MK-SVM, the two types of multiple kernels are used in the sub-classifiers to model static and longitudinal behavioral data respectively. The H-MK-SVM can be viewed as an ensemble classifier to combine the results of the sub-classifiers.

Figures 4 approximately here

6. Computational Experiments

The H-MK-SVM is a better and more useful approach for customer churn prediction using longitudinal behavioral data only if it outperforms other existing approaches. Hence, there are a few questions about the performance of the three frameworks and the corresponding methods on customer churn prediction using longitudinal behavioral data. One question is whether the framework with ensemble techniques outperforms the other two with transformed static data or with rectangularized longitudinal behavioral data. Another question is how the H-MK-SVM, the MK-SVM, the SVM and other more traditional methods compare using multiple criteria measuring performance. One more question is whether the H-MK-SVM and other methods are equally effective on balanced and imbalanced data. Using three real-world databases, these questions are answered by means of computational experiments. In the following, the datasets used and the design of the computational experiments are described. Data preprocessing and parameter tuning are also discussed.

6.1 The databases

Three datasets extracted from three real-world databases, Foodmart 2000, AdventureWorksDW, and Telecom, are used in the computational experiments. Table 1 summarizes the characteristics of these datasets including the number of observations N, the number of static attributes m_1, the number of longitudinal behavioral attributes M, the length, i.e., the number of time points, of each longitudinal behavioral attribute in the original datasets T', the length of each longitudinal behavioral attribute in the training, validation and testing sets T, and the percentage of churners in the datasets p.

Table 1 approximately here

The Foodmart dataset is extracted from the retail database Foodmart 2000 in Microsoft SQL Server 2000 (Chen *et al.*, 2005; Tsai and Shieh, 2009). The database includes the customer and transactional datasets of a chain store. The customer dataset records 10281 customers with different types of membership cards. Two transactional datasets in the database are used for this study. The Sales_Fact_1997 dataset records 86,837 transactions in 1997 and the Sales_Fact_1998 dataset records 182,283 transactions in 1998. A purchase record in the transactional datasets includes transaction date, sales amount, whether or not promotion was involved, and so on. The $M = 3$ longitudinal behavioral variables, the amount spent on all product categories, the number of purchases, and the number of products purchased per month by each customer, are computed from these two datasets. The length of each longitudinal behavioral variable is $T' = 24$ months. The $m_1 = 11$ static attributes derived from the customer dataset include age, gender, marital status, annual income, total number of children, number of children at home, education, membership card, occupation, homeowner, and the number of automobiles owned.

The Adventure dataset is extracted from the database AdventureWorksDW[2] in Microsoft SQL

[2] available at: http://msftdbprodsamples.codeplex.com/releases/view/4004.

Server 2005. The datasets used in the experiments include the Reseller and the ResellerSales datasets. The Reseller dataset records 701 resellers, and the ResellerSales dataset records 60855 transactions about sales amount, order quantity per purchase, and so on. The experiments use $M = 3$ longitudinal behavioral variables with $T' = 36$ time points and $m_1 = 7$ static variables.

The Telecom database is from a cell phone service company provided by the Center for CRM at Duke University[3]. The database contains $M = 3$ longitudinal behavioral variables, i.e., plan chosen, total minutes consumed each month, and whether or not a promotion was involved, for $T' = 20$ months.

6.2 Experimental design

Fig. 5 shows the process of customer churn prediction using longitudinal behavioral data. The blocks of the process are described in the following.

Figure 5 approximately here

Data preprocessing: the demographic data can be directly obtained from the data warehouse of the businesses, while the longitudinal behavioral data need to be extracted from transactional datasets. Data preprocessing methods including normalization and sampling are applied to the static and longitudinal behavioral data (Crone *et al.*, 2006).

Model training and validation: A holdout validation approach is used in the computational experiments. Each dataset is randomly partitioned into a training set, a validation set and a testing set with $N/10$ ($N/3$), $9N/20$ ($N/3$) and $9N/20$ ($N/3$) observations respectively for the Foodmart and Telecom (Adventure) datasets. The three phase, two phase, and the interior-point algorithms are used to train the H-MK-SVM, the MK-SVM and the SVM, respectively. These algorithms are implemented in the Matlab 7.4 development environment. In the training process, the computational results are evaluated on the validation set to find the optimal kernel parameters and the regularization parameters.

Model testing: The parameters and the sparse coefficients learned from model training and validation are used in the classification function to evaluate each observation in the testing set. Criteria used to measure performance include the percentage of correctly classified observations (PCC), the percentage of correctly classified observations in the positive class (Sensitivity), the percentage of correctly classified observations in the negative class (Specificity), the area under the receiver operating characteristic curve (AUC), the top 10% lift (Lift), the maximum profit (MP) and the H-measure (H). The LSSVMlab toolbox

[3] available at: http://www.fuqua.duke.edu/centers/ccrm/datasets/download.html.

was used for the computation of the AUC. Verbeke *et al.* (2011) provided a detailed description of the computation of the MP criterion. The settings of the parameters used to compute the MP criterion are specified in Verbeke *et al.* (2011). Although the settings of these parameters are different for different applications, the fixed settings do not affect the performance comparison between the H-MK-SVM and the other classifiers. The results reported in the next section are the ones on the testing sets.

6.3 Data preprocessing

In the transactional datasets (Sales_Fact_1997 and Sales_Fact_1998, for example), each record represents a transaction of one product (service) category for one customer. Analysis Services of Microsoft SQL Server 2005 was used to transform the transactional datasets into the longitudinal behavioral datasets. For example, the amount of spending (volume), the number of purchases (frequency), and the number of products purchased (variety) are the three longitudinal behavior variables that represent the customer behavior varying over time in the Foodmart and the Adventure datasets.

Observations with missing values in either the static or the longitudinal behavioral variables were deleted. Observations with no transactions in the time periods of the training sets were also deleted. After data preprocessing, the Foodmart dataset contains 8842 observations and the Adventure dataset contains 633 observations. All the 3399 observations in the Telecom datasets are retained.

The Telecom dataset is in a contractual setting and the Foodmart and the Adventure datasets are not. The labels of the observations in the Telecom dataset are known because of the contractual nature of the telecommunications business (Zorn *et al.*, 2010). The customer lifetime value (CLV) is used as the criterion to distinguish churners from non-churners in the Foodmart and the Adventure datasets. Details about the computation of the CLV are provided by Benoit and Van den Poel (2009). The time periods used in the testing set are different from those in the training and validation sets to reflect the temporal nature of the data. In the Foodmart (Adventure) dataset, the customer longitudinal behavioral data from January 1997 to December 1997 (from July 2002 to June 2004) are used to train and validate the models and those from July 1997 to June 1998 (from January 2003 to December 2004) are used to test the models. A customer or reseller is labeled as a churner if the CLV in the last six months of the corresponding period is zero.

In customer churn prediction, the number of churners is usually much smaller than the number of non-churners in the available datasets. Hence, the datasets are usually imbalanced. For example, only 24.33% of the customers are churners in the Adventure dataset. One of the most commonly used techniques for dealing with imbalanced data is sampling, such as oversampling and undersampling (Burez and Van den

Poel, 2009; Verbeke *et al.*, 2011). Undersampling is used in this experiment. The sampling ratio θ is defined as the number of non-churners over the number of churners in a training set. The training set is balanced if $\theta = 1$. In the computational experiments, both balanced and imbalanced training sets are used. Balanced training sets from all three datasets are used. Imbalanced training sets with $\theta = 2$, $\theta = 5$ and $\theta = 15$ from the Adventure dataset are used. The training and the validation sets of the Adventure dataset are pooled together first. The $N/3$ observations in the imbalanced training sets are obtained by sampling the pooled set. The rest of the observations in the pooled set are used as the validation set. All observations in the testing set are kept intact and used to test the models.

6.4 Parameter tuning

The grid search approach is used when selecting regularization parameters and kernel parameters for the H-MK-SVM, the MK-SVM, and the SVM. These parameters control the error margin tradeoff and play a crucial role in the performance and the sparseness of the models (Chen *et al.*, 2007, 2011).

In the experiments, the Gaussian kernel function (9) is used in the H-MK-SVM, the MK-SVM, and the SVM. For kernel methods, different kernel functions or a kernel function with different parameters correspond to different mappings and thus can be viewed as capturing different information in the data (Lanckriet *et al.*, 2004b). In the multiple kernel function (13), the identical kernel parameter $1/\sigma^2$ is used in each single kernel $k_d(x_{i,d}, x_{j,d})$ for both feature selection and classification when learning the sparse coefficients β_d. In the H-MK-SVM, the static and longitudinal behavioral data have different domains. The static data are expressed as rectangular matrices and the longitudinal behavioral data are expressed as three-dimensional matrices in which each observation corresponds to a multivariate time series. Therefore, a kernel parameter $1/\sigma_1^2$ is used for the multiple kernel function (27) for the static data and a different kernel parameter $1/\sigma_2^2$ is used for the multiple kernel function (28) for the longitudinal behavioral data.

Using the grid search approach in the three phase training algorithm for the H-MK-SVM, exponentially growing values for C, $1/\sigma_1^2$ and $1/\sigma_2^2$, i.e., $C = 2^{-10}, \ldots, 2^0, \ldots, 2^{10}$ and $1/\sigma_1^2$ ($1/\sigma_2^2$)$=10\times 2, 10\times 2^2, \ldots, 10\times 2^{10}$, are tried in phase 1; exponentially growing values for λ, i.e., $\lambda = 2^{-10}, \ldots, 2^0, \ldots, 2^{10}$, are tried in phase 2; and exponentially growing values for $\tilde{\lambda}$, i.e., $\tilde{\lambda} = 2^{-10}, \ldots, 2^0, \ldots, 2^{10}$, are tried in phase 3. The final values of these parameters used in the model are the ones with the best performance. Since no more than three parameters are tuned at a time, the computational time needed to find the final values by this tuning strategy is acceptable.

7. Computational Results

Computational results are reported in this section and the results answer the questions posted in the previous section. These results focus on customer churn prediction performance of the H-MK-SVM, the SVM and the MK-SVM. The SVM and the MK-SVM are used as benchmarks for comparisons because they are among the most effective methods and because they are the basis on which the H-MK-SVM is developed. Some other classifiers are also used as benchmarks for comparisons. All computations were performed on a laptop computer with an Intel Core i3 processor with a 2.40 GHz clock speed and 2GB of RAM.

The input data $\mathbf{x}_i = [\mathbf{s}_i, \mathbf{ts}_i]_{1\times m}$ and $\tilde{\mathbf{x}}_i = [\mathbf{s}_i, \mathbf{tb}_i]_{1\times(m_1+M\times T)}$ are both used in these methods, i.e., the frameworks in Fig. 1a and Fig. 1b, except in the H-MK-SVM. The results are reported in separate columns in the following tables. In these tables, D represents the dimension of the input data. For the standard framework in Fig. 1a, the input for observation i is $\mathbf{x}_i = [\mathbf{s}_i, \mathbf{ts}_i]_{1\times m}$ and thus $D = m$ (14, 10, and 3 for the three datasets, respectively). For the framework in Fig. 1b, the input for observation i is $\tilde{\mathbf{x}}_i = [\mathbf{s}_i, \mathbf{tb}_i]_{1\times(m_1+M\times T)}$, and thus $D = m_1 + M \times T$ (47, 79, and 60 for the three datasets, respectively). The best result for each measure and for each dataset is highlighted in these tables.

7.1 Performance of the H-MK-SVM, the MK-SVM and the SVM on balanced data

As previously stated, longitudinal behavioral data can be used in three ways for customer churn prediction. The SVM, the MK-SVM and many other traditional methods can be used within the standard framework and the framework with feature construction techniques, and the H-MK-SVM can be used within the framework with ensemble techniques. Therefore, the comparisons among the SVM, the MK-SVM and the H-MK-SVM are also the comparisons among these three ways of using longitudinal behavioral data. Results using balanced data for the three datasets are reported in Table 2. In addition to the performance criteria discussed earlier, the computational time (Time) in seconds is also reported in this table.

Tables 2 and 3 approximately here

As shown in Table 2, the AUC ranges between 76.31% and 98.70%; the Lift ranges between 4.16 and 9.15; the MP ranges between 2.39 and 22.50; and the H-measure ranges between 76.54% and 97.87% for the H-MK-SVM on these three datasets. The H-MK-SVM obtained the highest PCC, Sensitivity, AUC, Lift, MP and H-measure on these three datasets. In customer churn prediction, the misclassification of a churner may result in the loss of the customer but the misclassification of a non-churner may result in some extra marketing cost. Because the former is more costly than the latter, Sensitivity is a more important measure than Specificity. While Sensitivity and Specificity measure how accurate the methods can identify the observations in a single class, the AUC measures how well the methods discriminate the two classes. The H-MK-SVM does not obtain the highest Specificity, but it obtains the highest AUC. The computational time taken to train the H-MK-SVM is from 2 to 4 times of that taken to train the MK-SVM or the SVM.

The AUC, the Lift and the H-measure of the MK-SVM and the SVM are a bit higher using D=14 attributes than using D=47 attributes on the Foodmart dataset and a bit lower using D=3 attributes than using D=60 attributes on the Telecom dataset. Therefore, the performance of the MK-SVM and the SVM using rectangularized longitudinal behavioral data, i.e., the framework in Fig. 1b, is not obviously superior to that using the transformed static data, i.e., the framework in Fig. 1a.

The number of selected static attributes (m''), the number of selected longitudinal behavioral attributes (M''), and the number of selected time points in all the longitudinal behavioral attributes ($\tilde{T}$), along with m_1, M and $M \times T$, are listed in Table 3 for the H-MK-SVM. For example, $m'' = 2$ static attributes are selected from a total of $m_1 = 11$; $M'' = 2$ longitudinal behavioral attributes are selected from a total of $M = 3$; and $\tilde{T} = 7$ time points are selected from a total of $M \times T = 36$ for the Foodmart dataset. The most discriminative predictors in the Foodmart dataset are the two longitudinal behavioral variables, volume and frequency, and two static variables, annual income and occupation.

7.2 Performance comparison of different classifiers on balanced data

More computations are conducted to further compare the performance of the H-MK-SVM, the MK-SVM, the SVM and other methods. The eight other methods used in the experiments are least squares SVM (LS-SVM), feed-forward artificial neural network (FANN), radial basis function neural network (RBFNN), decision tree (DT), random forest (RF), AdaBoosting (Boosting), logistic regression (LR) and the proportional hazard model (Cox). The randomforest and GML AdaBoost Matlab toolboxes were used to implement the RF and the Boosting methods, respectively. Matlab 7.4 and the corresponding Matlab toolboxes are used to implement the other methods.

The AUC and the Lift of these methods are presented in Table 4. The H-MK-SVM obtained the highest AUC on the Adventure and Telecom datasets and the highest Lift on the Foodmart and the Adventure datasets, LR obtained the highest AUC on the Foodmart dataset, while DT, RF and Boosting obtained the highest Lift on the Telecom dataset. These results show that the use of the longitudinal behavioral data in the H-MK-SVM without transformation improves the performance of customer churn prediction.

Table 4 approximately here

Results in Table 4 also show that the performances of the other eight methods using rectangularized longitudinal behavioral attributes, i.e., the framework in Fig. 1b, are not obviously superior to those using aggregated attributes, i.e., the framework in Fig. 1a. Specially, the AUC and Lift of LR and Cox using rectangularized longitudinal behavioral attributes are far lower than those using aggregated attributes on the Foodmart and Adventure datasets.

7.3 Performance on imbalanced data

The undersampling approach produces small subsets of data for model construction when the dataset is small. Classification techniques may not perform well on high dimensional small datasets because the classification functions may be overfit and, therefore, have poor generalization properties. Therefore, imbalanced data may be used directly for model construction without sampling to obtain a balanced training set. A classifier with good performance on both balanced and imbalanced data is robust and is more preferred. The Adventure dataset is used to test the performance of the classification methods on imbalanced data because it has relatively a small number of observations and a high dimension.

The performances of the H-MK-SVM, the MK-SVM and the SVM, as well as the other eight methods, using the AUC and the Lift as criteria on the Adventure dataset with θ=1, 2, 5 and 15 are reported in Table 5. The H-MK-SVM obtained the highest AUC and Lift with θ=1, 2 and 15. The AUC of the H-MK-SVM, the MK-SVM, the SVM and the LS-SVM are all larger than 90% for θ=1, 2, 5 and 15. The AUCs of these four methods deceased by no more than 5%, while those of the FANN, the RBFNN, the DT and Boosting deceased by more than 10% as the sampling ratio θ increased. Hence, the H-MK-SVM, the MK-SVM, the SVM and the LS-SVM are robust with imbalanced data while the H-MK-SVM performs the best. Cox with D=10 also obtained an AUC larger than 90% for θ=1, 2, 5 and 15.

Table 5 approximately here

8. Conclusions

In this study, the frameworks for customer churn prediction using longitudinal behavioral data are developed, the H-MK-SVM model is formulated, and a three phase training algorithm is developed. The H-MK-SVM uses multiple kernels to construct a classification function with static and longitudinal behavioral data as input. The training process of the H-MK-SVM is also a feature selection process because the sparse non-zero coefficients correspond to the selected variables.

Computational experiments are conducted on three real-world databases. The experimental results show that the H-MK-SVM exhibits superior performance on both balanced and imbalanced data as compared to the MK-SVM, the SVM and eight other existing methods. The use of longitudinal behavioral data in the H-MK-SVM without transformation improves the performance of customer churn prediction. Furthermore, good prediction results are obtained in both contractual and non-contractual settings.

Large datasets are usually used for customer churn prediction in practice. The SVM, the MK-SVM and the H-MK-SVM are computationally more intensive than other more traditional methods. Therefore, more efficient optimization techniques need to be developed to reduce the computation time needed to train the H-MK-SVM. Collaborative pattern mining (Zhu *et al*., 2011) is an emergent framework for large-scale computation in a distributed environment. The development of a collaborative optimization method for coefficient estimation and parameter tuning will be a direction for future work. Selecting useful attributes from a large number of prediction variables in the framework with feature construction techniques will also be a direction for further research.

A good customer churn prediction model should identify potential churners not only as accurately as possible but also as early as possible. When longitudinal behavioral data are available, a dynamic model can be constructed to update customer information and to provide prediction results frequently. A new criterion can be developed to measure the performance of the model using the time period from the time of correct prediction to that of churn. Hence, dynamic customer churn prediction is another research direction.

Acknowledgements:

The authors greatly appreciate the two anonymous reviewers for their constructive suggestions which made the manuscript much stronger. This work was partly supported by the National Natural Science Foundation of China for Creative Research Groups (71021061), the National Natural Science Foundation of China (90924016, 71101023) and the China Postdoctoral Science Foundation (20110491502).

References：

Bach, F. R., Lanckrient, G. R. G., Jordan, M.I., 2004. Multiple kernel learning, conic duality and the SMO algorithm. In: G. Russell, S. Dale (Eds.), Proceedings of the Twenty First International Conference on Machine Learning, pp. 41-48.

Baesens, B., Verstraeten, G., Van den Poel, D., Egmont-Petersen, M., Van Kenhove, P., Vanthienen, J., 2004. Bayesian network classifiers for identifying the slope of the customer lifecycle of long-life customers. *European Journal of Operational Research*, **156**, 508–523.

Benoit, D. F., Van den Poel, D., 2009. Benefits of quantile regression for the analysis of customer lifetime value in a contractual setting: An application in financial services. *Expert Systems with Applications*, **36**, 10475–10484.

Buckinx, W., Van den Poel, D., 2005. Customer base analysis: Partial defection of behaviourally loyal clients in a non-contractual FMCG retail setting. *European Journal of Operational Research*, **164**, 252–268.

Burez, J., Van den Poel, D., 2007. CRM at a pay-TV company: Using analytical models to reduce customer attrition by targeted marketing for subscription services. *Expert Systems with Applications*, **32**, 277–288.

Burez, J., Van den Poel, D., 2009. Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, **36**, 4626–4636.

Cao, L., 2010. In-depth behavior understanding and use: The behavior informatics approach. *Information Sciences*, **180**, 3067–3085.

Cao, L., Yu, P. S., 2009. Behavior informatics: an informatics perspective for behavior studies. *IEEE Intelligent Informatics Bulletin*, **10**, 6–11.

Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S., 2002. Choosing multiple parameters for support vector machines. *Machine Learning*, **46**, 131–159.

Chen, M. C., Chiu, A. L., Chang, H. H., 2005. Mining changes in customer behavior in retail marketing. *Expert Systems with Applications*, **28**, 773–781.

Chen, Z. Y., Li, J. P., Wei, L. W., 2007. A multiple kernel support vector machine scheme for feature selection and rule extraction from gene expression data of cancer tissue. *Artificial Intelligence in Medicine*, **41**, 161–175.

Chen, Z. Y., Li, J. P., Wei, L. W., Xu, W. X., Shi, Y., 2011. Multiple kernel support vector machine based multiple tasks oriented data mining system for gene expression data analysis. *Expert System with*

Applications, **38**, 12151–12159.

Coussement, K., Van den Poel, D., 2008a. Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications*, **34**, 313–327.

Coussement, K., Van den Poel, D., 2008b. Integrating the voice of customers through call center emails into a decision support system for churn prediction. *Information & Management*, **45**, 164–174.

Crone, S. F., Lessmann, S., Stahlbock, R., 2006. The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing. *European Journal of Operational Research*, **173**, 781–800.

Dekimpe, M. G., Hanssens, D. M., 2000. Time-series models in marketing: Past, present and future. *International Journal of Research in Marketing*, **17**, 183–193.

Demiriz, A., Bennett, K. P., Shawe-Taylor, J., 2002. Linear programming boosting via column generation. *Machine Learning*, **46**, 225–254.

Eichinger, F., Nauck, D. D., Klawonn, F., 2006. Sequence mining for customer behaviour predictions in telecommunications. In: Proceedings of the Workshop on Practical Data Mining: Applications, Experiences and Challenges, Berlin, Germany.

Glady, N., Baesens, B., Croux, C., 2009. Modeling churn using customer lifetime value. *European Journal of Operational Research*, **197**, 402–411.

Gönen, M., Alpaydın, E., 2011. Multiple Kernel Learning Algorithms. *Journal of Machine Learning Research*, **12**, 2211–2268.

Gunn, S. R., Kandola, J. S., 2002. Structural modeling with sparse kernels. *Machine Learning*, **48**, 137–163.

Huang, B. Q., Kechadi, T.-M., Buckley, B., Kiernan, G., Keogh, E., Rashid, T., 2010. A new feature set with new window techniques for customer churn prediction in land-line telecommunications. *Expert Systems with Applications*, **37**, 3657–3665.

Keerthi, S. S., Sindhwani, V., Chapelle, O., 2007. An efficient method for gradient-based adaptation of hyperparameters in SVM Models. In: Schölkopf, B., Platt, J. C., Hoffman, T. (Eds.), *Advances in Neural Information Processing Systems* **19**. Cambridge: MIT Press, pp. 217–224.

Kisioglu, P., Topcu, Y. I., 2010. Applying Bayesian belief network approach to customer churn analysis: A case study on the telecom industry of Turkey. *Expert Systems with Applications*, **38**, 7151–7157.

Lanckrient, G. R. G., Cristianini, N., Bartlett, P., El Ghaoui, L., Jordan, M. I., 2004a. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, **5**, 27–72.

Lanckriet, G. R. G., De Bie, T., Cristianini, N., Jordan, M. I., Noble, W. S., 2004b. A statistical framework for genomic data fusion. *Bioinformatics*, **20**, 2626–2635.

Lee, Y., Kim, Y., Lee, S., Koo, J.Y., 2006. Structured multicategory support vector machines with analysis of variance decomposition. *Biometrika* , **93**, 555–571.

Lessmann, S., Voß, S., 2008. Supervised classification for decision support in customer relationship management. In: Bortfeldt, A., Homberger, J., Kopfer, H., Pankratz, G., Stangmeier, R. (Eds.), *Intelligent Decision Support*, Gabler, Wiesbaden, pp. 231–253.

Lessmann, S., Voß, S., 2009. A reference model for customer-centric data mining with support vector machines. *European Journal of Operational Research*, **199**, 520–530.

Lin, Y., Zhang, H. H., 2006. Component selection and smoothing in multivariate nonparametric regression. *The annals of statistics*, **34**, 2272–2297.

Ngai, E. W. T., Xiu, L., Chau, D. C. K., 2009. Application of data mining techniques in customer relationship management: A literature review and classification. *Expert system with applications*, **36**, 2592–2602.

Orsenigo, C., Vercellis, C., 2010. Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification. *Pattern Recognition*, **43**, 3787–3794.

Prinzie, A., Van den Poel, D., 2006a. Incorporating sequential information into traditional classification models by using an element/position-sensitive SAM. *Decision Support Systems*, **42**, 508–526.

Prinzie, A., Van den Poel, D., 2006b. Investigating purchasing sequence patterns for financial services using Markov, MTD and MTDg models. *European Journal of Operational Research*, *170*, 710–734.

Prinzie, A., Van den Poel, D., 2007. Predicting home-appliance acquisition sequences: Markov/Markov for discrimination and survival analysis for modelling sequential information in NPTB models. *Decision Support Systems*, **44**, 28–45.

Prinzie, A., Van den Poel, D., 2009. Modeling complex longitudinal consumer behavior with dynamic Bayesian networks: an acquisition pattern analysis application. *Journal of Intelligent Information System*, doi: 10.1007/s10844-009-0106-7.

Qi, J., Zhang, L., Liu, Y., Li, L., Zhou, Y., Shen, Y., Liang L., Li, H., 2009. ADTreesLogit model for customer churn prediction. *Annuls of Operations Research*, **168**, 247–265.

Rakotomamonjy, A., Bach, F. R., Canu, S., Grandvalet, Y., 2008. SimpleMKL. *Journal of Machine Learning Research*, **9**, 2491–2521.

Schölkopf, B., Smolla, A., 2002. *Learning with kernels–Support Vector Machines, Regularization, Optimization and Beyond*. MIT press, Cambridge, MA.

Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B., 2006. Large scale multiple kernel learning. *Journal of Machine Learning Research*, **1**, 1–18.

Sun, J., He, K. Y., Li, H., 2011. SFFS-PC-NN optimized by genetic algorithm for dynamic prediction of financial distress with longitudinal data streams. *Knowledge-Based Systems*, **24**, 1013–1023.

Tsai, C. F., Lu, Y. H., 2009. Customer churn prediction by hybrid neural networks. *Expert Systems with Applications*, **36**, 12547–12553.

Tsai, C. Y., Shieh, Y. C., 2009. A change detection method for sequential patterns. *Decision Support Systems*, **46**, 501–511.

Tseng, P., 2001. Convergence of a block coordinate decent method for nondifferentiable minimization. *Journal of optimization theory and applications*, **109**, 475–494.

Vapnik, V. N., 1995. *The nature of statistic learning theory*. Springer, New York.

Vapnik, V. N., 1998. *Statistic learning theory*. Wiley, New York.

Verbeke, W., Martens, D., Mues, C., Baesens, B., 2011. Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, **38**, 2354–2364.

Verbeke, W., Dejaeger, K., Martens, D., Hur, J., Baesens, B., 2011. New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research*, **218**, 211–229.

Van den Poel, D., Larivière, B., 2004. Customer attrition analysis for financial services using proportional hazard models. *European Journal of Operational Research*, **157**, 196–217.

Woodsend, K., Gondzio, J., 2009. Hybrid MPI/OpenMP parallel linear support vector machine training. *Journal of Machine Learning Research*, **10**, 1937–1953.

Yu, X., Guo, S. , Guo, J., Huang, X., 2010. An extended support vector machine forecasting framework for customer churn in e-commerce. *Expert Systems with Applications*, **38**, 1425–1430.

Zhu, X., Li, B., Wu, X., He, D., Zhang, C., 2011. CLAP: Collaborative pattern mining for distributed information systems. *Decision Support Systems*, doi:10.1016/j.dss.2011.05.002.

Zorn, S., Jarvis, W., Bellman, S., 2010. Attitudinal perspectives for predicting churn. *Journal of Research in Interactive Marketing*, 4, 157–169.

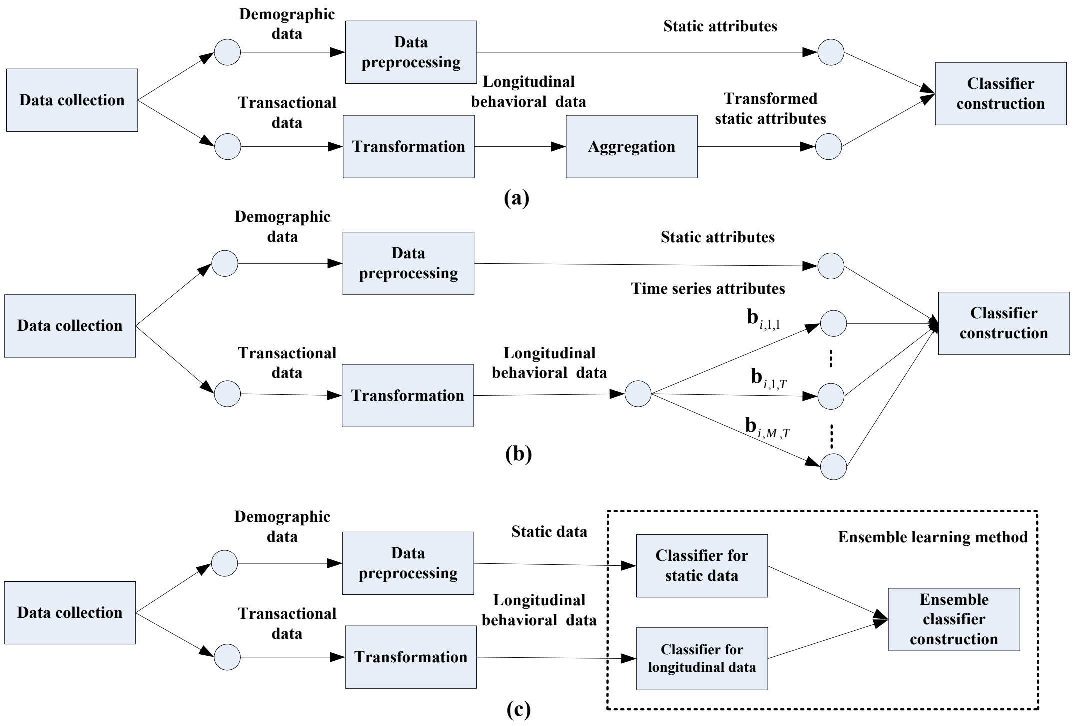

Figure 1. Frameworks for customer churn prediction: (a) the standard framework; (b) a framework with feature construction techniques; (c) a framework with ensemble techniques.

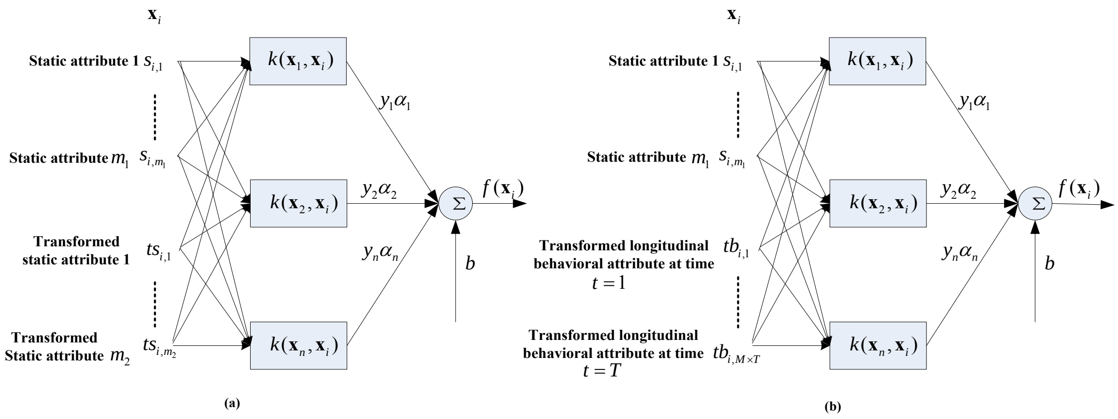
$\mathbf{x}_i$
Static attribute 1 $s_{i,1}$
Static attribute m_1 s_{i,m_1}
Transformed static attribute 1 $ts_{i,1}$
Transformed Static attribute m_2 ts_{i,m_2}
$k(\mathbf{x}_1, \mathbf{x}_i)$
$k(\mathbf{x}_2, \mathbf{x}_i)$
$k(\mathbf{x}_n, \mathbf{x}_i)$
$y_1\alpha_1$
$y_2\alpha_2$
$y_n\alpha_n$
Σ
$f(\mathbf{x}_i)$
b
(a)
$\mathbf{x}_i$
Static attribute 1 $s_{i,1}$
Static attribute m_1 s_{i,m_1}
Transformed longitudinal behavioral attribute at time $t=1$ $tb_{i,1}$
Transformed longitudinal behavioral attribute at time $t=T$ $tb_{i,M\times T}$
$k(\mathbf{x}_1, \mathbf{x}_i)$
$k(\mathbf{x}_2, \mathbf{x}_i)$
$k(\mathbf{x}_n, \mathbf{x}_i)$
$y_1\alpha_1$
$y_2\alpha_2$
$y_n\alpha_n$
Σ
$f(\mathbf{x}_i)$
b
(b)

Figure 2. SVM for customer churn prediction: (a) the standard framework; (b) the framework with feature construction techniques.

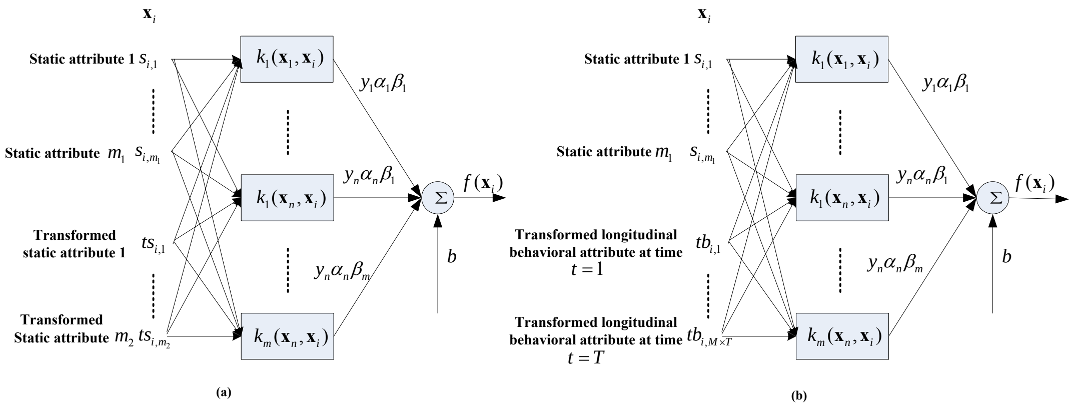
$\mathbf{x}_i$
Static attribute 1 $s_{i,1}$
Static attribute m_1 s_{i,m_1}
Transformed static attribute 1 $ts_{i,1}$
Transformed Static attribute m_2 ts_{i,m_2}
$k_1(\mathbf{x}_1,\mathbf{x}_i)$
$k_1(\mathbf{x}_n,\mathbf{x}_i)$
$k_m(\mathbf{x}_n,\mathbf{x}_i)$
$y_1\alpha_1\beta_1$
$y_n\alpha_n\beta_1$
$y_n\alpha_n\beta_m$
Σ
$f(\mathbf{x}_i)$
b
(a)
$\mathbf{x}_i$
Static attribute 1 $s_{i,1}$
Static attribute m_1 s_{i,m_1}
Transformed longitudinal behavioral attribute at time $t=1$ $tb_{i,1}$
Transformed longitudinal behavioral attribute at time $t=T$ $tb_{i,M\times T}$
$k_1(\mathbf{x}_1,\mathbf{x}_i)$
$k_1(\mathbf{x}_n,\mathbf{x}_i)$
$k_m(\mathbf{x}_n,\mathbf{x}_i)$
$y_1\alpha_1\beta_1$
$y_n\alpha_n\beta_1$
$y_n\alpha_n\beta_m$
Σ
$f(\mathbf{x}_i)$
b
(b)

Figure 3. MK-SVM for customer churn prediction: (a) the standard framework; (b) the framework with feature construction techniques.

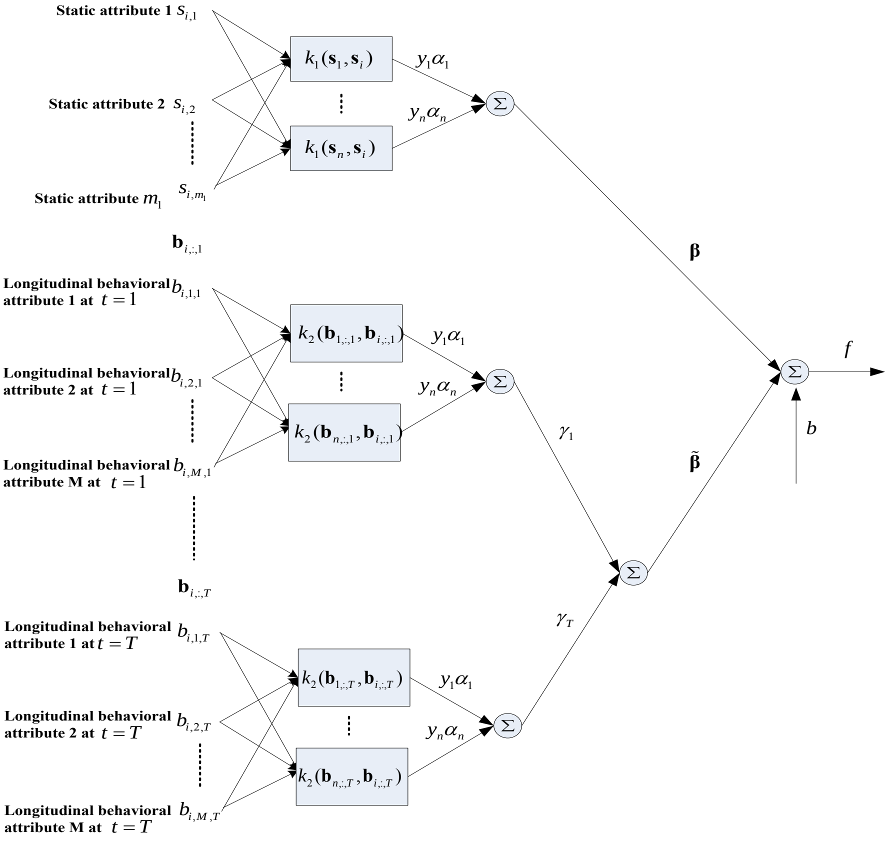
Static attribute 1 $s_{i,1}$
Static attribute 2 $s_{i,2}$
Static attribute m_1 s_{i,m_1}
$k_1(\mathbf{s}_1,\mathbf{s}_i)$
$k_1(\mathbf{s}_n,\mathbf{s}_i)$
$y_1\alpha_1$
$y_n\alpha_n$
Σ
$\boldsymbol{\beta}$
$\mathbf{b}_{i,:,1}$
Longitudinal behavioral attribute 1 at $t=1$ $b_{i,1,1}$
Longitudinal behavioral attribute 2 at $t=1$ $b_{i,2,1}$
Longitudinal behavioral attribute M at $t=1$ $b_{i,M,1}$
$k_2(\mathbf{b}_{1,:,1},\mathbf{b}_{i,:,1})$
$k_2(\mathbf{b}_{n,:,1},\mathbf{b}_{i,:,1})$
$y_1\alpha_1$
$y_n\alpha_n$
Σ
γ_1
f
Σ
b
$\tilde{\boldsymbol{\beta}}$
Σ
$\mathbf{b}_{i,:,T}$
γ_T
Longitudinal behavioral attribute 1 at $t=T$ $b_{i,1,T}$
Longitudinal behavioral attribute 2 at $t=T$ $b_{i,2,T}$
Longitudinal behavioral attribute M at $t=T$ $b_{i,M,T}$
$k_2(\mathbf{b}_{1,:,T},\mathbf{b}_{i,:,T})$
$k_2(\mathbf{b}_{n,:,T},\mathbf{b}_{i,:,T})$
$y_1\alpha_1$
$y_n\alpha_n$
Σ

Figure 4. H-MK-SVM for customer churn prediction using longitudinal behavioral data

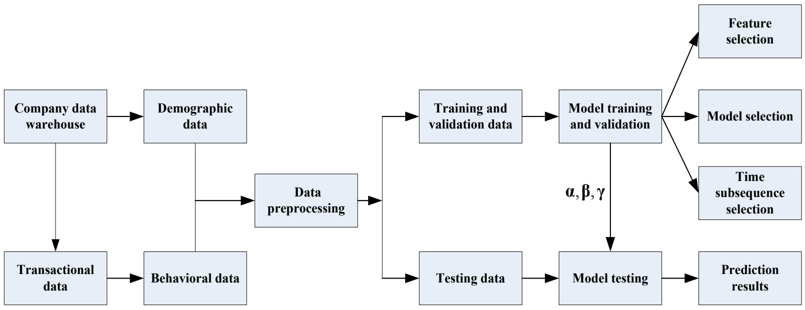

Figure 5. The process of the H-MK-SVM for customer churn prediction using longitudinal behavioral data

Table 1. Database characteristics

Dataset	N	m_1	M	T'	T	p
Foodmart	8842	11	3	24	12	20.90
Adventure	633	7	3	36	24	24.33
Telecom	3399	0	3	20	20	47.71

Table 2. Results of the H-MK-SVM, the MK-SVM and the SVM on the Foodmart, Adventure and Telecom datasets

	Foodmart					Adventure					Telecom				
Criterion	H-MK-	MK-SVM		SVM		H-MK-S	MK-SVM		SVM		H-MK-	MK-SVM		SVM	
	SVM	$D = 14$	$D = 47$	$D = 14$	$D = 47$	VM	$D = 14$	$D = 47$	$D = 14$	$D = 47$	SVM	$D = 14$	$D = 47$	$D = 14$	$D = 47$
PCC	**70.67**	61.00	65.81	62.81	65.00	**96.67**	**96.67**	**96.67**	95.83	**96.67**	**97.92**	83.43	**97.92**	82.75	97.50
Sensitivity	**95.63**	88.00	71.50	92.63	72.25	**100.00**	95.00	93.33	93.33	95.00	**95.83**	80.33	**95.83**	80.50	**95.83**
Specificity	45.75	34.00	**60.12**	32.81	57.75	93.33	98.33	**100.0**	98.33	93.33	**100.00**	86.50	**100.0**	85.00	99.17
AUC	**76.31**	70.65	69.32	72.02	68.40	**99.19**	98.83	97.75	99.03	98.25	**98.70**	90.60	98.18	90.70	98.24
Lift	**9.15**	8.42	6.84	8.88	6.94	**8.22**	**8.22**	**8.22**	**8.22**	**8.22**	**4.16**	3.37	4.12	3.37	4.12
MP	**2.39**	1.32	–0.99	1.97	–0.88	**5.30**	**5.30**	**5.30**	**5.30**	**5.30**	**22.50**	14.70	**22.50**	14.70	22.22
H	**76.54**	69.29	67.65	71.30	67.37	**96.77**	96.60	96.55	95.72	94.21	**97.87**	82.89	**97.87**	82.35	97.46
Time	240.16	64.05	122.14	**56.01**	112.88	3.69	2.81	2.96	**1.20**	3.11	136.61	35.38	35.92	**31.94**	33.06

Table 3. Results for feature selection and time subsequence selection

Dataset	m_1	m''	M	M''	$M \times T$	$\tilde{T}$
Foodmart	11	2	3	2	36	6
Adventure	7	3	3	2	72	7
Telecom	0	0	3	3	60	9

Table 4. Results of different classifiers using the AUC and the Lift criteria on balanced data

Methods	AUC						Lift					
	Foodmart		Adventure		Telecom		Foodmart		Adventure		Telecom	
H-MK-SVM	76.31		**99.19**		**98.70**		**9.15**		**8.22**		4.16	
D	14	47	10	79	3	60	14	47	10	79	3	60
MK-SVM	70.65	69.32	98.83	97.75	90.60	98.18	8.42	6.84	**8.22**	**8.22**	3.37	4.12
SVM	72.02	68.40	99.03	98.25	90.70	98.24	8.88	6.94	**8.22**	**8.22**	3.37	4.12
LS-SVM	83.64	76.58	97.92	97.58	68.70	97.82	8.67	7.72	**8.22**	**8.22**	4.19	4.02
FANN	84.37	69.88	98.06	99.06	83.70	97.92	8.61	5.69	**8.22**	**8.22**	4.09	4.05
RBFNN	53.50	50.76	93.64	90.36	78.34	81.20	3.73	1.67	6.86	7.52	3.81	4.02
DT	60.12	55.95	97.50	97.50	87.45	95.41	6.03	5.98	8.01	8.01	3.44	**4.19**
RF	59.00	57.69	95.83	95.00	83.42	97.92	7.22	6.75	6.86	6.86	3.40	**4.19**
Boosting	53.56	54.75	97.50	97.50	82.08	97.75	6.03	5.07	7.81	7.81	3.33	**4.19**
LR	**84.78**	73.18	98.78	68.83	82.27	82.27	8.76	6.89	**8.22**	0.00	3.81	3.81
Cox	83.17	71.71	96.86	74.44	82.23	97.61	8.43	5.98	**8.22**	6.85	3.84	4.02

Table 5. Results of different classifiers using the AUC and the Lift criteria on the Adventure dataset with θ=1, 2, 5 and 15

Methods	AUC								Lift							
	θ=1		θ=2		θ=5		θ=15		θ=1		θ=2		θ=5		θ=15	
H-MK-SVM	**99.19**		**97.28**		94.31		**96.33**		**8.22**		**8.22**		**8.22**		**8.22**	
D	10	79	10	79	10	79	10	79	10	79	10	79	10	79	10	79
MK-SVM	98.83	95.06	93.67	93.06	**98.83**	91.72	94.78	93.31	**8.22**	3.56	**8.22**	**8.22**	**8.22**	**8.22**	**8.22**	**8.22**
SVM	99.03	96.58	92.81	92.86	91.19	87.78	91.44	92.25	**8.22**	3.42	**8.22**	7.53	**8.22**	7.53	**8.22**	**8.22**
LS-SVM	97.92	97.58	94.94	92.83	90.72	90.42	91.25	92.72	**8.22**	**8.22**	**8.22**	**8.22**	**8.22**	**8.22**	**8.22**	**8.22**
FANN	98.06	99.06	93.36	94.86	88.56	88.56	85.33	79.61	**8.22**	**8.22**	**8.22**	**8.22**	**8.22**	7.53	**8.22**	**8.22**
RBFNN	93.64	90.36	89.36	71.17	84.22	60.83	68.72	75.44	6.86	7.52	7.19	3.08	6.17	6.89	7.53	6.85
DT	97.50	97.50	94.17	94.17	94.17	95.75	85.83	85.83	8.01	8.01	**8.22**	**8.22**	**8.22**	**8.22**	**8.22**	**8.22**
RF	95.83	95.00	92.50	89.17	77.50	87.50	90.00	67.50	6.86	6.86	7.53	7.88	6.89	6.89	**8.22**	**8.22**
Boosting	97.50	97.50	94.17	94.17	93.33	93.33	86.67	86.67	7.81	7.81	**8.22**	**8.22**	7.53	7.53	**8.22**	**8.22**
LR	98.78	68.83	90.82	74.65	93.11	82.67	83.90	86.99	**8.22**	0.00	7.53	0.00	7.53	7.53	**8.22**	6.17
Cox	96.86	74.44	94.08	72.03	92.39	77.86	95.42	84.78	**8.22**	6.85	**8.22**	4.80	**8.22**	4.11	**8.22**	7.53