THE UNIVERSITY OF TEXAS AT SAN ANTONIO, COLLEGE OF BUSINESS

Working Paper SERIES

Date April 25, 2014

WP # 0004MSS-MINGHESUN-2014

Ensemble Learning for Cross-Selling Using Multitype Multiway 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

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

Ensemble Learning for Cross-Selling Using Multitype Multiway Data

Zhen-Yu Chen[a,*], 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:

Cross-selling is an integral component of customer relationship management. Using relevant information to improve customer response rate is a challenging task in cross-selling. Incorporating multitype multiway customer behavioral, including related product, similar customer and historical promotion, data into cross-selling models is helpful in improving the classification performance. Customer behavioral data can be represented by multiple high-order tensors. Most existing supervised tensor learning methods cannot directly deal with heterogeneous and sparse multiway data in cross-selling. In this study, two novel ensemble learning methods, multiple kernel support tensor machine (MK-STM) and multiple support vector machine ensemble (M-SVM-E), are proposed for cross-selling using multitype multiway data. The MK-STM and the M-SVM-E can also perform feature selections from large sparse multitype multiway data. Based on these two methods, collaborative and non-collaborative ensemble learning frameworks are developed. In these frameworks, many existing classification and ensemble methods can be combined for classification using multitype multiway data. Computational experiments are conducted on two databases extracted from open access databases. The experimental results show that the MK-STM exhibits the best performance and has better performance than existing supervised tensor learning methods.

Keywords: Data mining; Customer relationship management; Direct marketing; Cross-selling; Ensemble learning; Multitype multiway data; Big data; Support tensor machine

JEL Classification: C32, C38, C51, C61

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

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

1. Introduction

Cross-selling has become an integral component of customer development in the life cycle of customer relationship management (CRM) (Ngai *et al*., 2009). Cross-selling refers to promotion activities aiming at selling products to customers who have already bought some other products from the same vendor (Knott *et al.*, 2002; Li *et al.*, 2011; Ngai *et al.*, 2009). Selling additional products to the same customers can help the firm increase the customer lifetime value, improve the relationship with customers and reduce the chance of churn (Prinzie and Van den Poel, 2006; Rust and Chung, 2006). The task of identifying specific customers for cross-selling recommendations is a two-class classification problem.

Cross-selling managers face a challenging task of improving the low customer response rate (Li *et al*., 2011). Unlike repeated purchases, there are not any historical purchase data about customers of the products (services) for recommendation in cross-selling. Customer demographic data are usually used in cross-selling modeling. Introducing customer behavioral data may help in improving classification performance and customer response rate. In addition to customer demographic data, three types of customer behavioral data are used in this study as listed in the following.

(1) Related product data. Customer purchase history of related products, *i.e*., products that are similar or complementary to the target product, can be used to predict the purchase likelihood of the target product. Related product data have been used in classification models for cross-selling (Kamakura *et al*., 2004; Prinzie and Van den Poel, 2006, 2007, 2011). Three typical purchase behavioral variables are recency, frequency and monetary (RFM) variables. Moreover, similarities between products have been used in item-based collaborative filtering methods to make recommendations (Adomavicius and Tuzhilin, 2005).

(2) Similar customer data. Purchase history of similar customers, *i.e*., customers with similar purchase behavior to the target customer, can be used to predict the purchase likelihood of the target customer. Similar customer data have not yet been used for cross-selling modeling. However, similarities between customers have been used in user-based collaborative filtering methods to make recommendations (Adomavicius and Tuzhilin, 2005).

(3) Historical promotion data. Past promotions may have long-term effects on customer purchase behavior (Li *et al*., 2011). A few studies have used historical promotion data as controlled variables in cross-selling models (Li *et al*., 2011).

All these three types of data are longitudinal behavioral data because each of them has a time aspect. Prinzie and Van den Poel (2006, 2007, 2011) described customer purchase behavior as unidimensional or multivariate sequences without considering the time aspect and used sequence analysis techniques to predict the next product to purchase. Unlike the sequential data, as special cases of longitudinal behavioral data (Chen *et al*., 2012), longitudinal behavioral data have fixed time-intervals, and thus may be used to predict both the likelihood and the timing of the next purchase of specific products for cross-selling.

The related product, similar customer and historical promotion data have higher dimensions than the customer demographic data. They are of multiway data and are represented by high-order tensors (Hoff, 2011). A tensor that generalizes the notions of vectors (first-order tensors) and matrices (second-order tensors) is a natural way to represent multiway data (Signoretto *et al*., 2011). Each dimension of a tensor is called a mode. The related product data can be represented by a fourth-order tensor with four modes: customer × product × RFM variables × time; the similar customer data can be represented by a fifth-order tensor with five modes: customer × similar customer × product × RFM variables × time; and the historical promotion data can be represented by a fourth-order tensor with four modes: customer × product × promotion × time. Because multiple types of multiway data are used, the term "multitype multiway data", as a particular type of "big data", is used to represent the input data of the ensemble learning models in this study.

Compared with multiway data in many other applications such as image and medical signal processing, multiway data in cross-selling have two distinct characteristics, *i.e*., heterogeneousness and sparseness. Business firms usually record and store large amount of heterogeneous customer data in their data warehouses (Chen *et al*., 2012). As mentioned above, the classification models in cross-selling involve three types of multiway data with different orders. To the best of our knowledge, no supervised tensor learning methods can be directly applied to multitype multiway data. Moreover, the uses of large amount of longitudinal behavioral, *i.e*., related product, similar customer and historical promotion, data in cross-selling provide both opportunities to improve the classification performance and challenges to deal with redundant data. Hence, it is important to develop sparse tensor learning methods to identify potential sparse structures of multitype multiway data. Most existing supervised tensor learning methods cannot filer multiway data and end up with using sparse representations.

In this study, two novel data mining methods, multiple kernel support tensor machine (MK-STM) and multiple support vector machine (SVM) ensemble (M-SVM-E), are proposed for cross-selling using multitype multiway data. Based on the MK-STM, a collaborative ensemble learning (CEL) framework is developed. In this framework, the base learners can be combined by integrative, parallel or sequential collaborative learning. Based on the M-SVM-E, a non-collaborative ensemble learning (NCEL) framework is developed. The advantage of this framework is that many existing classification and ensemble methods can be combined for classification using multitype multiway data. Unlike other supervised tensor learning methods, the ensemble learning methods including the MK-STM and M-SVM-E can directly deal with multitype multiway data. Furthermore, the MK-STM and M-SVM-E, as selective ensemble methods, can select a subset of features, *i.e.*, variables, with good discriminative abilities from a large number of variables in the sparse multitype multiway data.

This article is organized as follows. Section 2 discusses the relevant literature and outlines the contributions of this study. Section 3 describes the preliminaries of multilinear algebra, three typical supervised tensor learning methods, and tensor representation of the input data. The CEL framework and the MK-STM for cross-selling using multitype multiway data are developed in Section 4. The NCEL framework and the M-SVM-E for cross-selling using multitype multiway data are developed in Section 5. The computational experiments are described in Section 6. The computational results are reported in Section 7. Conclusions and directions for further research are given in Section 8.

2. Relevant Literature

This study is related to three fields of research in the literature, *i.e.*, cross-selling, supervised tensor classification and ensemble learning methods. These three fields will be discussed briefly and the contributions of this study relative to these fields will be outlined.

Unlike other elements of CRM and direct marketing such as customer segmentation, customer targeting and churn management, there are relatively few studies on cross-selling (Ngai *et al*., 2009; Prinzie and Van den Poel, 2006). Different, including statistical (Ansell *et al*., 2007; Kamakura *et al*., 2004; Li *et al*., 2011; Prinzie and Van den Poel, 2006, 2007), mathematical programming (Li *et al*.,

2011) and machine learning, methods (Ahn *et al*., 2011; Prinzie and Van den Poel, 2011) have been used to identify cross-selling opportunities. Li *et al*. (2011) applied a multivariate probit model to predict customer responses for cross-selling solicitations. They then proposed a stochastic dynamic programming model to take into account the temporal customer demand and the long-term effect of cross-selling promotions to reach decisions about cross-selling recommendations. Prinzie and Van den Poel (2006, 2007, 2011) considered customer purchase behavior as unidimensional or multivariate sequences, and respectively applied the mixture transition distribution model, the Markov chain and the Bayesian network to model the behavioral data and predict the next purchase of a customer. Kamakura *et al*. (2004) developed a multivariate split-hazard model to estimate the probability and timing of purchasing new products. Ansell *et al*. (2007) combined the customer lifestyle segmentation and the proportional hazard model to identify the cross-selling opportunities. Ahn *et al*. (2011) combined genetic algorithms with multiple classification methods for cross-selling in the mobile telecom market.

The second field of research related to this study is supervised tensor classification methods. Prior to applying tensor learning methods, the high-order tensors have to be vectorized or unfolded in advance before being used as input into traditional classification models. Compared with vector-represented learning methods, the tensor learning methods can preserve natural data structure and prevent information loss. In the last several years, supervised tensor learning have drawn wide attention in the fields of image, vision, video and medical signal processing (Hao *et al.*, 2013; Lu *et al.*, 2011; Signoretto *et al.*, 2011). The representative supervised tensor learning methods include the support tensor machine (STM) proposed by Tao *et al.* (2007), linear support high-order tensor machine (SHTM) proposed by Hao *et al.* (2013) and tensor kernel method (TK) proposed by Signoretto *et al.* (2011).

Ensemble learning, the third field of research related to this study, is a learning paradigm that combines multiple base learners to solve a problem (Dietterich, 2000). It is widely accepted that an ensemble of multiple classifiers often performs better than a single classifier (Zhou *et al.*, 2002). Therefore, ensemble learning has been an active area of study and has been successfully applied to semi-supervised, active, cost-sensitive and class-imbalanced learning (Zhou, 2012). Recently, much attention has been given to developing more efficient ensemble learning algorithms (Zhang and Zhou, 2011).

This study makes three major contributions. The first major contribution is the incorporation of multitype multiway data into classification models for cross-selling and tensor-based classification methods so as to improve classification performance and to improve customer response rate. In the past, customer demographic and aggregated behavioral data represented by matrices have usually been used as the input of standard cross-selling models. Despite the many uses of the tensor-based technologies across the literature of machine learning, attempt has not been made towards applying the tensor-based technologies to customer behavior modeling and cross-selling. The second major contribution of this study is the extension of the SHTM proposed by Hao *et al.* (2013) to the MK-STM by combining multiple kernel learning techniques and using the projection and hierarchical kernels. Moreover, the proposed MK-STM and M-SVM-E can directly deal with the multitype multiway data, while the SHTM and other supervised tensor learning methods cannot. The third major contribution is the development of two ensemble learning frameworks applying the existing and ensemble classification methods for supervised learning with multitype multiway data. As far as we know, there are no attempts in the literature to introduce ensemble learning methods into the field of supervised tensor learning.

3. Preliminaries

In this section, the basic definitions and concepts of multilinear algebra are given, three methods for supervised tensor learning including the STM, SHTM and TK are presented, and the tensor representation of the input data to the ensemble learning models are formally described. The multilinear algebra, SHTM and TK are the foundations for the proposed MK-STM. The STM, SHTM and TK are all used as competitive methods in the computational experiments.

3.1 Basic definitions

Following the conventional notations of multilinear algebra (Kolda and Bader, 2009; Lu *et al.*, 2011), vectors are denoted by boldface lowercase letters, *e.g.*, $\mathbf{a}$, matrices by boldface capital letters, *e.g.*, $\mathbf{A}$, and tensors by calligraphic letters, *e.g.*, $\mathcal{A}$. In the notation $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, N is the order, *i.e.*, the number of modes or ways, of $\mathcal{A}$ and I_p is the size, *i.e.*, the number of elements, of dimension p for $1 \le p \le N$. An element of $\mathcal{A}$ is denoted by $a_{i_1,i_2,\cdots,i_N}$ where $1 \le i_p \le I_p$ for $1 \le p \le N$.

The tensor product (outer product) of two tensors $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\mathcal{Y} \in \mathbb{R}^{I'_1 \times I'_2 \times \cdots \times I'_{N'}}$ is defined by

$$\mathcal{V} = (\mathcal{X} \circ \mathcal{Y}) \tag{1}$$

with elements $v_{i_1,i_2,\cdots,i_N,i'_1,i'_2,\cdots,i'_{N'}} = x_{i_1,i_2,\cdots,i_N} y_{i'_1,i'_2,\cdots,i'_{N'}}$. The inner product of two tensors $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined by

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1,i_2,\cdots,i_N} y_{i_1,i_2,\cdots,i_N} \ . \tag{2}$$

Unfolding is the process of reordering the elements of a tensor into a matrix (Kolda and Bader, 2009). The pth mode unfolding of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ ($p-$mode) is defined by

$$\mathcal{X}^{<p>} \in \mathbb{R}^{I_p \times (I_1 \times \cdots \times I_{p-1} \times I_{p+1} \times \cdots \times I_N)} , \tag{3}$$

where the column vectors of $\mathcal{X}^{<p>}$ are the $p-$mode vectors of $\mathcal{X}$ (Lu *et al.*, 2011). Note that there are different ways of ordering of the $p-$mode vectors in the literature, but the ordering does not affect the computational results as long as it is consistent (Kolda and Bader, 2009).

The $p-$mode product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J_p \times I_p}$, denoted by $\mathcal{H} = \mathcal{X} \times_p \mathbf{U}$, is a tensor in $\mathbb{R}^{I_1 \times I_2 \times \cdots I_{p-1} \times J_p \times I_{p+1} \times \cdots \times I_N}$ with elements

$$h_{i_1,i_2,\cdots,i_{p-1},j_p,i_{p+1},\cdots,i_N} = \sum_{i_p=1}^{I_p} x_{i_1,i_2,\cdots,i_N} u_{j_p,i_p} \ . \tag{4}$$

The Frobenius norm of a tensor $\mathcal{X} \in R^{I_1 \times I_2 \times \cdots \times I_N}$ is given by

$$\|\mathcal{X}\|_F = \langle \mathcal{X}, \mathcal{X} \rangle = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x^2_{i_1,i_2,\cdots,i_N}} \ . \tag{5}$$

The Frobenius norm of a tensor $\mathcal{X}$ measures the size of the tensor and its square is the energy of the tensor (Tao *et al.*, 2007). The distance between two tensors $\mathcal{X}$ and $\mathcal{Y}$ is denoted by $\|\mathcal{X} - \mathcal{Y}\|_F$.

If a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ can be written as

$$\mathcal{X} = \sum_{r=1}^{R} u_r^{<1>} \circ u_r^{<2>} \circ \cdots \circ u_r^{<N>} = \sum_{r=1}^{R} \prod_{p=1}^{N} \circ u_r^{<p>} \quad \left(u_r^{<p>} \in R^{I_p} \right), \tag{6}$$

then the result in (6) is called the rank-1 decomposition of $\mathcal{X}$ with length R. If $R=1$, the tensor $\mathcal{X}$ is called a rank-1 tensor.

3.2 Support tensor machine

The details of the STM can be found in Tao *et al.* (2007). A training dataset is represented by $G=\{(\mathcal{X}_{1,}y_1),\cdots,(\mathcal{X}_{n,}y_n)\}$ where $\mathcal{X}_i$ is the input, $y_i \in \{1,-1\}$ is the class label or the desired output of observation i and n is the number of observations. A linear classification function is constructed in the original input space

$$f(\mathcal{X}_{i_0}) = \operatorname{sgn}\left(\mathcal{X}_{i_0}\prod_{p=1}^{N}\times_n \mathbf{w}^{<p>} + b\right), \tag{7}$$

for any observation i_0 with an input $\mathcal{X}_{i_0}$ by training a STM, where $\mathbf{w}^{<p>}$ is the weight vector of the pth hyperplane and b is the bias.

Let $\mathcal{W} = \mathbf{w}^{<1>} \circ \mathbf{w}^{<2>} \circ \cdots \circ \mathbf{w}^{<N>}$ be a rank-1 tensor. The weight vector $\mathbf{w}^{<p>}$ and the bias b are obtained by solving the following quadratic programming (QP) model

$$\min_{\mathbf{w}^{<p>},b,\boldsymbol{\xi}} \quad \frac{1}{2}\left\|\prod_{p=1}^{N}\circ\mathbf{w}^{<p>}\right\|_F^2 + C\sum_{i=1}^{n}\xi_i \tag{8}$$

$$\text{s.t.} \quad y_i\left(\mathcal{X}_i\prod_{p=1}^{N}\times_p \mathbf{w}^{<p>} + b\right) \geq 1-\xi_i \qquad i=1,\cdots,n \tag{9}$$

$$\xi_i \geq 0 \qquad i=1,\cdots,n, \tag{10}$$

where C is the regularization parameter. In the model, ξ_i is an error term for observation i and $\boldsymbol{\xi}$ is the vector of all ξ_i for $i=1,\cdots,n$. The QP model in (8)-(10) is the primal formulation of the STM.

An alternating projection method (Tao *et al.*, 2007) was proposed to decompose the QP model (8) -(10) into P sub-problems, each of which is also a QP model,

$$\min_{\mathbf{w}^{<p>},b^{<p>},\xi^{<p>}} \quad \frac{1}{2}\left\|\mathbf{w}^{<p>}\right\|_F^2\left\|\prod_{l=1,l\neq p}^{N}\mathbf{w}^{(l)}\right\|_F^2 + C\sum_{i=1}^{n}\xi_i^{<p>} \tag{11}$$

$$\text{s.t.} \quad y_i\left((\mathbf{w}^{<p>})^T\left(\mathcal{X}_i\prod_{l=1,l\neq p}^{N}\times_l \mathbf{w}^{(l)}\right) + b^{<p>}\right) \geq 1-\xi_i^{<p>} \qquad i=1,\cdots,n \tag{12}$$

$$\xi_i^{<p>} \geq 0 \qquad i=1,\cdots,n. \tag{13}$$

In this QP model, $\mathbf{w}^{<p>}$ and $b^{<p>}$ are the weighting vector and the bias of the pth hyperplane, $\xi_i^{<p>}$ is the error term of observation i corresponding to the pth hyperplane, and $\boldsymbol{\xi}^{<p>}$ is the vector of $\xi_i^{<p>}$ for $i = 1, \cdots, n$. Each sub-problem in (11)-(13) is a standard SVM and is iteratively solved to obtain the final results of the original problem in (8)-(10).

3.3 Linear support high-order tensor machine

The details of the SHTM can be found in Hao *et al.* (2013). From the basic definitions of multilinear algebra, Hao *et al.* (2013) derived the following equations

$$\|\mathcal{W}\|_F^2 = \prod_{p=1}^{N} \left\|\mathbf{w}^{<p>}\right\|_F^2 \tag{14}$$

$$\langle \mathcal{W}, \mathcal{X}_i \rangle = (\mathbf{w}^{<p>})^T \left(\mathcal{X}_i \prod_{l=1, l \neq p}^{N} \times_l \mathbf{w}^{<l>} \right) = \mathcal{X}_i \prod_{l=1}^{N} \times_l \mathbf{w}^{<l>} . \tag{15}$$

Substituting the results in (14) and (15) into the STM model in (8)-(10) leads to the following QP model

$$\min_{\mathcal{W}, b, \boldsymbol{\xi}} \quad \frac{1}{2}\|\mathcal{W}\|_F^2 + C\sum_{i=1}^{n} \xi_i \tag{16}$$

$$\text{s.t.} \quad y_i \left(\langle \mathcal{W}, \mathcal{X}_i \rangle + b \right) \geq 1 - \xi_i \ , \qquad i = 1, \cdots, n \tag{17}$$

$$\xi_i \ \geq 0 \qquad i = 1, \cdots, n \cdot \tag{18}$$

The dual of the QP model in (16)-(18), also a QP model, is given as follows

$$\max_{\tilde{\boldsymbol{\alpha}}} \quad \sum_{i=1}^{n} \tilde{\alpha}_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{\alpha}_i \tilde{\alpha}_j y_i y_j (\mathcal{X}_i, \mathcal{X}_j) \tag{19}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i \tilde{\alpha}_i = 0 \, , \tag{20}$$

$$0 \leq \tilde{\alpha}_i \leq C \qquad i = 1, \cdots, n \, , \tag{21}$$

where $\tilde{\alpha}_i$ is the Lagrangian multiplier for observation i and $\tilde{\boldsymbol{\alpha}}$ is a vector with $\tilde{\alpha}_i$ for $i = 1, \cdots, n$ as components.

From the definitions of the rank-1 decomposition of a tensor in (6) and the inner product of $\mathcal{X}_i$ and $\mathcal{X}_j$ in (2), the QP model in (19)-(21) can be written as the following QP model

$$\max_{\tilde{\boldsymbol{\alpha}}} \quad \sum_{i=1}^{n} \tilde{\alpha}_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\hat{p}=1}^{R}\sum_{\hat{q}=1}^{R} \tilde{\alpha}_i \tilde{\alpha}_j y_i y_j \prod_{p=1}^{N} \mathbf{x}_{i\hat{p}}^{<p>} \mathbf{x}_{j\hat{q}}^{<p>} \tag{22}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i \tilde{\alpha}_i = 0 \, , \tag{23}$$

$$0 \le \tilde{\alpha}_i \le C \qquad i = 1, \cdots, n \, , \tag{24}$$

where $\mathbf{x}_{i\hat{p}}^{<p>}$ and $\mathbf{x}_{j\hat{q}}^{<p>}$ are the elements of the rank-1 decomposition of $\mathcal{X}_i$ and $\mathcal{X}_j$, respectively.

The QP model in (22)-(24) can be solved with standard QP solution algorithms or can be trained with standard SVM training procedures. The resulting classification function is

$$f(\mathcal{X}_{i_0}) = \operatorname{sgn}\left(\sum_{i=1}^{n} \sum_{\hat{p}=1}^{R} \sum_{\hat{q}=1}^{R} \tilde{\alpha}_i y_i \prod_{p=1}^{N} \mathbf{x}_{i\hat{p}}^{<p>} \mathbf{x}_{i_0\hat{q}}^{<p>} + b' \right), \tag{25}$$

for any observation i_0 with an input $\mathcal{X}_{i_0}$, where $\mathbf{x}_{i\hat{p}}^{<p>}$ and $\mathbf{x}_{i_0\hat{q}}^{<p>}$ are the elements of the rank-1 decomposition of $\mathcal{X}_i$ and $\mathcal{X}_{i_0}$, respectively, and b' is the bias.

3.4 Tensor kernels

Assume reproducing kernel Hilbert space $\left(\eta_1, \langle \cdot, \cdot \rangle_{\eta_1}\right)$, $\left(\eta_2, \langle \cdot, \cdot \rangle_{\eta_2}\right)$, $\cdots$, $\left(\eta_N, \langle \cdot, \cdot \rangle_{\eta_N}\right)$ of functions on an arbitrary set Ω . For any $p \in \mathbb{N}$, let $k^p = \langle \phi_p, \phi_p \rangle$ be a reproducing kernel of η_p with a nonlinear map ϕ_p .

Let ψ : $\eta_1 \times \eta_2 \times \cdots \times \eta_N \to \mathbb{R}$ be a bounded multilinear functional. For any $\mathcal{X} \in \Omega$, Signoretto *et al.*, (2011) showed that the multilinear function

$$\begin{aligned} \psi_{k_{\mathcal{X}}^1, k_{\mathcal{X}}^2, \cdots, k_{\mathcal{X}}^N}(f_1, f_2, \cdots, f_N) &= \left\langle k_{\mathcal{X}}^1, f_1 \right\rangle_{\eta_1} \left\langle k_{\mathcal{X}}^2, f_2 \right\rangle_{\eta_2} \cdots \left\langle k_{\mathcal{X}}^N, f_N \right\rangle_{\eta_N} \\ &= f_1(\mathcal{X}) f_2(\mathcal{X}) \cdots f_N(\mathcal{X}) \end{aligned} \tag{26}$$

belongs to the Hilber-Schmidt functions. For any $\mathcal{X} \in \Omega$ and $\mathcal{Y} \in \Omega$,

$$\left\langle \psi_{k_{\mathcal{X}}^1, k_{\mathcal{X}}^2, \cdots, k_{\mathcal{X}}^N}, \psi_{k_{\mathcal{Y}}^1, k_{\mathcal{Y}}^2, \cdots, k_{\mathcal{Y}}^N} \right\rangle = k^1(\mathcal{X}, \mathcal{Y}) k^2(\mathcal{X}, \mathcal{Y}) \cdots k^N(\mathcal{X}, \mathcal{Y}) \, . \tag{27}$$

According to (27), a kernel function can be stated as the product of some basic kernels

$$k(\mathcal{X}, \mathcal{Y}) = k^1(\mathcal{X}, \mathcal{Y}) k^2(\mathcal{X}, \mathcal{Y}) \cdots k^N(\mathcal{X}, \mathcal{Y}) \, , \tag{28}$$

where $k^p(\mathcal{X}, \mathcal{Y})$ for $p = 1, \cdots, N$ denotes the basic kernel of η_p . The basic kernel $k^p(\mathcal{X}, \mathcal{Y})$ can be a Gaussian, also called the RBF (Radial Basis Function), kernel

$$k^p(\mathcal{X}, \mathcal{Y}) = \exp\left(-\frac{1}{2\sigma^2} \left\| \mathcal{X}^{<p>} - \mathcal{Y}^{<p>} \right\|_F^2 \right), \tag{29}$$

where $1/\sigma^2$ is the kernel parameter and $\mathcal{X}^{<p>}$ and $\mathcal{Y}^{<p>}$ are the $p-$modes of $\mathcal{X}$ and $\mathcal{Y}$, respectively.

More generally, the basic kernel $k^p(\mathcal{X},\mathcal{Y})$ can be stated as

$$k^p(\mathcal{X},\mathcal{Y}) = \exp\left(-\frac{1}{2\sigma^2} d\left(\mathcal{X}^{<p>},\mathcal{Y}^{<p>}\right)^2\right), \tag{30}$$

where $d\left(\mathcal{X}^{<p>},\mathcal{Y}^{<p>}\right)$ denotes the distance between the $p-$modes of $\mathcal{X}$ and $\mathcal{Y}$. When a non-Euclidean distance is used, the function $d\left(\mathcal{X}^{<p>},\mathcal{Y}^{<p>}\right)$ denotes the chordal distance (projection Frobenius norm) on the Grassmannian manifolds (Signoretto *et al.*, 2011).

Let $\tilde{r}$ represent the rank of the $p-\text{mode}$ of a tensor $\mathcal{X}^{<p>}$, *i.e.*, $\tilde{r} = rank(\mathcal{X}^{<p>})$. Singular value decomposition is applied to the $p-\text{mode}$ of a tensor $\mathcal{X}^{<p>}$ as

$$\mathcal{X}^{<p>} = \left(\mathbf{U}_{\mathcal{X},1}^{<p>}\ \mathbf{U}_{\mathcal{X},2}^{<p>}\right)\begin{pmatrix} \mathbf{S}_{\mathcal{X},1}^{<p>} & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \mathbf{V}_{\mathcal{X},1}^{<p>} \\ \mathbf{V}_{\mathcal{X},2}^{<p>} \end{pmatrix}, \tag{31}$$

where $\mathcal{X}^{<p>}$ is a $n \times \tilde{m}$ matrix, $\mathbf{U}_{\mathcal{X},1}^{<p>}$ is a $n \times \tilde{r}$ matrix, $\mathbf{U}_{\mathcal{X},2}^{<p>}$ is a $n \times (n-\tilde{r})$ matrix, $\mathbf{V}_{\mathcal{X},1}^{<p>}$ is a $\tilde{r} \times \tilde{m}$ matrix, $\mathbf{V}_{\mathcal{X},2}^{<p>}$ is a $(\tilde{m}-\tilde{r}) \times \tilde{m}$ matrix and $\mathbf{S}_{\mathcal{X},1}^{<p>}$ is a $\tilde{r} \times \tilde{r}$ diagonal matrix.

The basic kernel $k^p(\mathcal{X},\mathcal{Y})$ adopting the projection Frobenius norm on the Grassmannian manifolds can be written as

$$k^p(\mathcal{X},\mathcal{Y}) = \exp\left(-\frac{1}{2\sigma^2}\left\|\mathbf{V}_{\mathcal{X},1}^{<p>}\mathbf{V}_{\mathcal{X},1}^{<p>\mathrm{T}} - \mathbf{V}_{\mathcal{Y},1}^{<p>}\mathbf{V}_{\mathcal{Y},1}^{<p>\mathrm{T}}\right\|_F^2\right). \tag{32}$$

This kernel uses the projection Frobenius norm to capture the topology of the input patterns (Signoretto *et al.*, 2011).

3.5 Tensor representation of the input data

As discussed in the Introduction, four types, *i.e.*, customer demographic, related product, similar customer and historical promotion, data are used as input to the ensemble learning models. The demographic data are represented by a matrix and the last three types of data are multiway data and are represented by tensors. The number of modes of the three tensors are represented by $N_2 = 4$, $N_3 = 5$ and $N_4 = 4$, respectively. The demographic data of a customer i is represented by the vector $\mathbf{x}_{1(i)} = \{x_{1(ij)} \mid j = 1,\cdots,m_1\}$ where m_1 denotes the number of demographic variables.

The related product data of a customer i is represented by the third-order tensor $\mathcal{X}_{2(i)} = \{x_{2(ij'kt)} \mid j' = 1,\cdots,m_2; k = 1,\cdots,m_3; t = 1,\cdots,T_1\}$. The three dimensions represent the related products, the RFM variables and the time points in the longitudinal purchase data. The numbers of the related products, RFM variables and time points are represented by m_2, m_3 and T_1, respectively.

The similar customer data of a customer i is represented by the fourth-order tensor $\mathcal{X}_{3(i)} = \{x_{2(i\tilde{i}\tilde{j}\tilde{k}\tilde{t})} \mid \tilde{i} = 1,\cdots,m_4; \tilde{j} = 1,\cdots,m_5; \tilde{k} = 1,\cdots,m_6; \tilde{t} = 1,\cdots,T_2\}$. The four dimensions represent the similar customers, the related products, the RFM variables and the time points. The numbers of similar customers, related products, RFM variables and time points are represented by m_4, m_5, m_6 and T_2, respectively.

The historical promotion data of a customer i is represented by the third-order tensor $\mathcal{X}_{4(i)} = \{x_{2(i\hat{j}\hat{k}\hat{t})} \mid \hat{j} = 1,\cdots,m_7; \hat{k} = 1,\cdots,m_8; \hat{t} = 1,\cdots,T_3\}$. The three dimensions represent the related products, the RFM variables and the time points. The numbers of the related products, RFM variables and time points are represented by m_7, m_8 and T_3, respectively.

Cross-selling aims at selling multiple associated products to customers with heterogeneous preferences. Hence, the solicitation decisions in cross-selling can be viewed as multiple binary classification problems. For a given product, the class label $y_i \in \{+1, -1\}$ in the dataset indicates the status of customer i, *i.e.*, $y_i = 1$ if customer i has purchased the product and $y_i = -1$ otherwise.

4. Collaborative Ensemble Learning and Multiple Kernel Support Tensor Machine

In this section, a CEL framework is developed. Based on the framework, a data mining model, the MK-STM, is formulated to integrate multitype multiway data. The model also performs feature selection from large sparse multitype multiway data. The kernels and the training method for the MK-STM are then presented.

4.1 The CEL framework

The MK-STM is trained through the CEL framework. The components of the CEL framework for cross-selling recommendations using multitype multiway data are summarized in the following.

(1) Data sources. Cross-selling using multitype multiway data involves diverse data from different sources. The input data include the demographic, related product, similar customer and historical promotion data which can be represented as a matrix, a fourth-order tensor, a fifth-order tensor and another fourth-order tensor, respectively.

(2) Unfolding of multiway data. An effective approach is needed to set up the kernel functions between two tensors. A straightforward way is to vectorize the multiway data, and then to use the standard kernels for the input vectors. However, unfolding, which transforms a tensor into matrices along each dimension, can preserve more structural information.

(3) Training the MK-STM with multiple tensor kernels. The projection and hierarchical kernels presented in Section 4.3 are used to model the diverse data including the demographic data and the three types of multiway data. The standard SVM training algorithms are then used to solve a QP problem to obtain the Lagrangian multipliers.

(4) Learning the weights of the basic kernels. When the projection and hierarchical kernels are used, a linear programming (LP) problem is solved to obtain the weights of the basic kernels. The final classification results are obtained using the Lagrangian multipliers and the weights of the basic kernels.

The MK-STM using multiple kernels to model diverse data is optimized to simultaneously obtain the final results. In this study, this method is called the CEL meaning that multiple base learners collaborate with each other to obtain the global results. With the development of massive data mining, collaborative learning has been the focus of study in recent years. For example, Zhu *et al.* (2011) developed a collaborative pattern mining framework for distributed frequent pattern mining.

The CEL framework is illustrated in Fig. 1. The framework consists of three main components: data sources, unfolding and collaborative learning. Collaborative learning is an integrated process of training and ensemble of the base learners. There are three, *i.e.*, integrative, parallel and sequential, ways for collaborative learning.

For integrative collaborative learning, the base learners are integrated into one model and an efficient algorithm is necessary to solve this model to obtain the weights of the base learners and the final results. The multiple kernel learning (MKL) algorithms such as the multiple kernel SVM (MK-

SVM) and the MK-STM discussed in the next subsection are the typical methods of integrative collaborative learning.

For parallel collaborative learning, the base learners are trained in a distributed way and they collaborate with each other by some communication strategy, *e.g.*, through a central processor, to obtain the final results. The collaborative MKL algorithm (Chen and Fan, 2012) is a typical method of parallel collaborative learning.

For sequential collaborative learning, the base learners are trained sequentially and the result of one base learner is used in the training of the next. For example, the boosting algorithms compute the weights of the observations in the training dataset according to the training errors of the previous base learner, and then use the weights to select observations from the training datasets and compute the weighted training errors of the current base learner.

Selective ensemble may be used as an ensemble strategy in the CEL framework. For example, the LP boosting method used to train the MK-STM in the second phase of the two-phase iterative strategy is selective ensemble selecting a few from a large number of basic kernels.

Fig. 1 approximately here

4.2 The MK-STM

In the last decade, SVM and MKL have been hot topics in the field of machine learning (Bach *et al.*, 2004; Cui and Curry, 2005). Specifically, MK-SVM is a state-of-the-art ensemble learning method which can combine multiple heterogeneous data and improve classification performance (Chen *et al.*, 2007). In this study, the MK-SVM is extended to the MK-STM for classification with multitype multiway data in cross-selling.

A MK-STM is used to construct a classification function. Consider a training dataset $G=\{(\mathbf{x}_{1(1)},\mathcal{X}_{2(1)},\mathcal{X}_{3(1)},\mathcal{X}_{4(1)},y_{(1)}),\cdots,(\mathbf{x}_{1(n)},\mathcal{X}_{2(n)},\mathcal{X}_{3(n)},\mathcal{X}_{4(n)},y_{(n)})\}$ with $\mathbf{x}_{1(i)}$, $\mathcal{X}_{2(i)}$, $\mathcal{X}_{3(i)}$ and $\mathcal{X}_{4(i)}$ as the input and $y_{(i)}\in\{1,-1\}$ as the class label or the desired output of observation i. A classification function of the following form is determined through learning

$$\begin{aligned} &f(\mathbf{x}_{1(i_0)},\mathcal{X}_{2(i_0)},\mathcal{X}_{3(i_0)},\mathcal{X}_{4(i_0)}) \\ &\quad=\operatorname{sgn}\left(\beta_1\left\langle\mathbf{w}_1,\mathbf{x}_{1(i_0)}\right\rangle+\beta_2\left\langle\mathcal{W}_2,\mathcal{X}_{2(i_0)}\right\rangle+\beta_3\left\langle\mathcal{W}_3,\mathcal{X}_{3(i_0)}\right\rangle+\beta_4\left\langle\mathcal{W}_4,\mathcal{X}_{4(i_0)}\right\rangle+\tilde{b}\right), \end{aligned} \tag{33}$$

for any observation i_0 with an input $(\mathbf{x}_{1(i_0)}, \mathcal{X}_{2(i_0)}, \mathcal{X}_{3(i_0)}, \mathcal{X}_{4(i_0)})$ by training a MK-STM, where $\mathbf{w}_1$, $\mathcal{W}_2$, $\mathcal{W}_3$ and $\mathcal{W}_4$ are the weights of the data components, β_1, β_2, β_3 and β_4 are the overall weights of the four types of input data, and $\tilde{b}$ is the bias. In the following, $\boldsymbol{\beta}$ is used to represent a vector with components $\beta_{\tilde{q}}$ for $\tilde{q} = 1, \cdots, 4$.

The weights $\mathbf{w}_1$, $\mathcal{W}_2$, $\mathcal{W}_3$ and $\mathcal{W}_4$ and the bias $\tilde{b}$ are obtained by solving the following QP model

$$\min_{\mathbf{w}_1, \mathcal{W}_2, \mathcal{W}_3, \mathcal{W}_4, \tilde{b}, \tilde{\boldsymbol{\xi}}, \boldsymbol{\beta}} \quad \frac{\beta_1}{2}\|\mathbf{w}_1\|_F^2 + \frac{\beta_2}{2}\|\mathcal{W}_2\|_F^2 + \frac{\beta_3}{2}\|\mathcal{W}_3\|_F^2 + \frac{\beta_4}{2}\|\mathcal{W}_4\|_F^2 + C\sum_{i=1}^{n}\tilde{\xi}_i \tag{34}$$

$$\text{s.t.} \quad y_i\left(\left\langle \mathbf{w}_1, \phi_1\left(\mathbf{x}_{1(i)}\right)\right\rangle + \left\langle \mathcal{W}_2, \phi_2\left(\mathcal{X}_{2(i)}\right)\right\rangle + \left\langle \mathcal{W}_3, \phi_3\left(\mathcal{X}_{3(i)}\right)\right\rangle + \left\langle \mathcal{W}_4, \phi_4\left(\mathcal{X}_{4(i)}\right)\right\rangle + \tilde{b}\right) \geq 1 - \tilde{\xi}_i \quad i = 1, \cdots, n \tag{35}$$

$$\tilde{\xi}_i \geq 0 \qquad i = 1, \cdots, n, \tag{36}$$

where C is the regularization parameter and ϕ_1, ϕ_2, ϕ_3 and ϕ_4 are the nonlinear maps. The model in (34)-(36) is the primal MK-STM model, while the weights $\mathbf{w}_1$, $\mathcal{W}_2$, $\mathcal{W}_3$ and $\mathcal{W}_4$, the bias $\tilde{b}$, the overall weights $\boldsymbol{\beta}$ and the error terms $\tilde{\boldsymbol{\xi}}$ are the primal variables in the QP model.

With the Lagrangian multipliers $\alpha_i \geq 0$ for the constraints in (35) and $\mu_i \geq 0$ for the constraints in (36), for $i = 1, \cdots, n$, the Lagrangian of the QP model in (34)-(36) is

$$\begin{aligned} L(\mathbf{w}_1, \mathcal{W}_2, \mathcal{W}_3, \mathcal{W}_4, \tilde{b}, \tilde{\boldsymbol{\xi}}, \boldsymbol{\alpha}) = {} & \frac{\beta_1}{2}\|\mathbf{w}_1\|_F^2 + \frac{\beta_2}{2}\|\mathcal{W}_2\|_F^2 + \frac{\beta_3}{2}\|\mathcal{W}_3\|_F^2 + \frac{\beta_4}{2}\|\mathcal{W}_4\|_F^2 + C\sum_{i=1}^{n}\tilde{\xi}_i - \sum_{i=1}^{n}\alpha_i \\ & \left\{ y_i\left(\left\langle \mathbf{w}_1, \phi_1\left(\mathbf{x}_{1(i)}\right)\right\rangle + \left\langle \mathcal{W}_2, \phi_2\left(\mathcal{X}_{2(i)}\right)\right\rangle + \left\langle \mathcal{W}_3, \phi_3\left(\mathcal{X}_{3(i)}\right)\right\rangle + \left\langle \mathcal{W}_4, \phi_4\left(\mathcal{X}_{4(i)}\right)\right\rangle + \tilde{b}\right) + \tilde{\xi}_i - 1 \right\} - \sum_{i=1}^{n}\mu_i\tilde{\xi}_i \end{aligned} . \tag{37}$$

The results in (38)-(43) in the following are obtained by taking the derivatives of the Lagrangian (37) with respect to the primal variables

$$\frac{\partial L}{\partial \mathbf{w}_1} = 0 \Rightarrow \mathbf{w}_1 = \sum_{i=1}^{n}\alpha_i y_i \phi_1(\mathbf{x}_{1(i)}) \tag{38}$$

$$\frac{\partial L}{\partial \mathcal{W}_2} = 0 \Rightarrow \mathcal{W}_2 = \sum_{i=1}^{n}\alpha_i y_i \phi_2(\mathcal{X}_{2(i)}) \tag{39}$$

$$\frac{\partial L}{\partial \mathcal{W}_3} = 0 \Rightarrow \mathcal{W}_3 = \sum_{i=1}^{n}\alpha_i y_i \phi_3(\mathcal{X}_{3(i)}) \tag{40}$$

$$\frac{\partial L}{\partial \mathcal{W}_4} = 0 \Rightarrow \mathcal{W}_4 = \sum_{i=1}^{n}\alpha_i y_i \phi_4(\mathcal{X}_{4(i)}) \tag{41}$$

$$\frac{\partial L}{\partial \tilde{b}} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i = 0 \tag{42}$$

$$\frac{\partial L}{\partial \tilde{\xi}_i} = 0 \Rightarrow \alpha_i + \mu_i = C \qquad\qquad i = 1, \cdots, n \tag{43}$$

Using the results in (38)-(43), the dual of the QP model in (34)-(36) is then given in (44)-(46) in the following

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\beta_1 k_1(\mathbf{x}_{1(i)}, \mathbf{x}_{1(j)}) + \left(\sum_{q=2}^{4} \beta_q k_q(\mathcal{X}_{q(i)}, \mathcal{X}_{q(j)}) \right) \right) \right\} \tag{44}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0 \tag{45}$$

$$0 \le \alpha_i \le C \qquad\qquad i = 1, \cdots, n\,, \tag{46}$$

where $\boldsymbol{\alpha}$ is the vector of Lagrangian multipliers with components α_i for $i = 1, \cdots, n$. The dual in (44)-(46) is a standard MKL problem. The input $(\mathbf{x}_{1(i)}, \mathcal{X}_{2(i)}, \mathcal{X}_{3(i)}, \mathcal{X}_{4(i)})$ of an observation i such that $0 < \alpha_i < C$ is called as a support tensor.

4.3 Kernels in the MK-STM

The non-tensor kernel $k_1(\mathbf{x}_{1(i)}, \mathbf{x}_{1(j)})$ in (44) can be the widely used Gaussian kernel

$$k_1(\mathbf{x}_{1(i)}, \mathbf{x}_{1(j)}) = \exp\left(-\frac{1}{2\sigma^2} \left\| \mathbf{x}_{1(i)} - \mathbf{x}_{1(j)} \right\|^2 \right). \tag{47}$$

According to (28), the tensor kernel $k_q(\mathcal{X}_{q(i)}, \mathcal{X}_{q(j)})$ in (44) can be written as the product of some basic kernels

$$k_q(\mathcal{X}_{q(i)}, \mathcal{X}_{q(j)}) = k_q^1(\mathcal{X}_{q(i)}^{<1>}, \mathcal{X}_{q(j)}^{<1>}) k_q^2(\mathcal{X}_{q(i)}^{<2>}, \mathcal{X}_{q(j)}^{<2>}) \cdots k_q^{N_q}(\mathcal{X}_{q(i)}^{<N_q>}, \mathcal{X}_{q(j)}^{<N_q>}) \tag{48}$$

where $k_q(\mathcal{X}_{q(i)}, \mathcal{X}_{q(j)})$ denotes the basic kernel on the tensor $\mathcal{X}_q$ and N_q, as mentioned in Section 3.5, denotes the order of $\mathcal{X}_{q(i)}$ for $q = 2, \cdots, 4$.

Compared with the multiplicative kernel in (48), the additive kernel has some advantages. The first advantage is its comprehensibility. The models with additive kernels are relatively easy to interpret (Christmann and Hable, 2012). As Chen *et al.* (2007) and Verbeke *et al.* (2011) pointed out, comprehensibility is an important metric to evaluate an intelligent model. The second advantage is its simplicity because additive kernels are linear combinations of basic kernels. Learning the weights of the sparse additive kernels in a LP model is computationally easy. The tensor kernel (48) can be written as the weighted sum of the basic kernels

$$k_q(\mathcal{X}_{q(i)}, \mathcal{X}_{q(j)}) = \sum_{p_q=1}^{N_q} \gamma_{p_q} k_q^{p_q}(\mathcal{X}_{q(i)}^{<p_q>}, \mathcal{X}_{q(j)}^{<p_q>}), \tag{49}$$

where γ_{p_q} is the weight of the basic kernel $k_q^{p_q}(\mathcal{X}_{q(i)}^{<p_q>}, \mathcal{X}_{q(j)}^{<p_q>})$, and N_q, as before, denotes the order of $\mathcal{X}_{q(i)}$. In the following, $\boldsymbol{\gamma}$ is used to represent a vector with components γ_{p_q} for $p_q = 1, \cdots, N_q$ and $q = 2, \cdots, 4$.

When the projection Frobenius norm is used in the basic kernel $k_q^{p_q}(\mathcal{X}_{q(i)}^{<p_q>}, \mathcal{X}_{q(j)}^{<p_q>})$, the kernel used in the QP model in (44)-(46) becomes

$$K_1(\boldsymbol{\beta}, \boldsymbol{\gamma}, \mathbf{x}_{1(i)}, \mathcal{X}_{2(i)}, \mathcal{X}_{3(i)}, \mathcal{X}_{4(i)}) = \\ \beta_1 k_1(\mathbf{x}_{1(i)}, \mathbf{x}_{1(j)}) + \left(\sum_{q=2}^{4} \beta_q \sum_{p_q=1}^{N_q} \gamma_{p_q} \exp\left(-\frac{1}{2\sigma^2} \left\| \mathbf{V}_{\mathcal{X}_{q(i),1}}^{<p_q>} \mathbf{V}_{\mathcal{X}_{q(i),1}}^{<p_q>\mathrm{T}} - \mathbf{V}_{\mathcal{X}_{q(j),1}}^{<p_q>} \mathbf{V}_{\mathcal{X}_{q(j),1}}^{<p_q>\mathrm{T}} \right\|_F^2 \right) \right). \tag{50}$$

The kernel with the projection Frobenius norm in (50) is called projection kernel.

When the Euclidean distance is used, the basic kernel $k_q^{p_q}(\mathcal{X}_{q(i)}^{<p_q>}, \mathcal{X}_{q(j)}^{<p_q>})$ can be written as a linear combination of multiple kernels

$$k_q^{p_q}(\mathcal{X}_{q(i)}^{<p_q>}, \mathcal{X}_{q(j)}^{<p_q>}) = \sum_{z_{p_q}=1}^{Z_{p_q}} \hat{\gamma}_{z_{p_q}} k_q^{p_q, z_{p_q}} \left(\mathcal{X}_{q(i, z_{p_q})}^{<p_q>}, \mathcal{X}_{q(j, z_{p_q})}^{<p_q>} \right), \tag{51}$$

where Z_{p_q} denotes the number of rows of the $p_q - \text{mode}$ matrix of the tensor $\mathcal{X}_q$ and $\hat{\gamma}_{z_{p_q}}$ is the weight of the basic kernel $k_q^{p_q, z_{p_q}} \left(\mathcal{X}_{q(i, z_{p_q})}^{<p_q>}, \mathcal{X}_{q(j, z_{p_q})}^{<p_q>} \right)$.

Using the multiple kernels in (49) and (51), the hierarchical kernel used in the QP model in (44)-(46) becomes

$$K_2(\boldsymbol{\beta}, \boldsymbol{\gamma}, \hat{\boldsymbol{\gamma}}, \mathbf{x}_{1(i)}, \mathcal{X}_{2(i)}, \mathcal{X}_{3(i)}, \mathcal{X}_{4(i)}) = \\ \beta_1 k_1(\mathbf{x}_{1(i)}, \mathbf{x}_{1(j)}) + \sum_{q=2}^{4} \beta_q \sum_{p_q=1}^{N_q} \gamma_{p_q} \sum_{z_{p_q}=1}^{Z_{p_q}} \hat{\gamma}_{z_{p_q}} k_q^{p_q, z_{p_q}} \left(\mathcal{X}_{q(i, z_{p_q})}^{<p_q>}, \mathcal{X}_{q(j, z_{p_q})}^{<p_q>} \right). \tag{52}$$

In (52), $\mathcal{X}_{q(i, z_{p_q})}^{<p_q>}$ and $\mathcal{X}_{q(j, z_{p_q})}^{<p_q>}$ are vectors representing the $z_{p_q}\text{th}$ rows of the $p_q - \text{mode}$ matrices of the tensors $\mathcal{X}_{q(i)}$ and $\mathcal{X}_{q(j)}$, respectively, and $\hat{\gamma}_{z_{p_q}}$ is the weight of the basic kernel $k_q^{p_q, z_{p_q}} \left(\mathcal{X}_{q(i, z_{p_q})}^{<p_q>}, \mathcal{X}_{q(j, z_{p_q})}^{<p_q>} \right)$. When convenient, $\hat{\boldsymbol{\gamma}}$ will be used to represent the vector with components

$\hat{\gamma}_{z_{p_q}}$ for $z_{p_q} = 1,\cdots,Z_{p_q}$, $p_q = 1,\cdots,N_q$ and $q = 2,\cdots,4$. The basic kernel $k_q^{p_q,z_{p_q}}\left(\mathcal{X}_{q(i,z_{p_q})}^{<p_q>},\mathcal{X}_{q(j,z_{p_q})}^{<p_q>}\right)$ can be a standard Gaussian kernel

$$k_q^{p_q,z_{p_q}}\left(\mathcal{X}_{q(i,z_{p_q})}^{<p_q>},\mathcal{X}_{q(j,z_{p_q})}^{<p_q>}\right) = \exp\left(-\frac{1}{2\sigma^2}\left\|\mathcal{X}_{q(i,z_{p_q})}^{<p_q>} - \mathcal{X}_{q(j,z_{p_q})}^{<p_q>}\right\|^2\right). \tag{53}$$

4.4 Training of the MK-STM

A two-phase iterative strategy (Chen *et al.*, 2007) is employed to train the MK-STM. When the weights $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ in the kernel in (50) (or $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\hat{\boldsymbol{\gamma}}$ in the kernel in (52)) are fixed, the QP problem in (44)-(46) is solved in the first phase with standard SVM training procedures to obtain the Lagrangian multipliers $\boldsymbol{\alpha}$. When the Lagrangian multipliers $\boldsymbol{\alpha}$ are fixed, the LP boosting method adopting the L_1 – norm based shrinkage strategy (Demiriz *et al.*, 2002) is employed in the second phase to solve the LP problem so as to obtain the sparse coefficients $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ (or $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\hat{\boldsymbol{\gamma}}$). The features corresponding to the non-zero components of the coefficients $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ (or $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\hat{\boldsymbol{\gamma}}$) are selected from the multitype multiway data. Therefore, the second phase is also a feature selection process. With fewer features in the models, the classification performance and the comprehensibility of the models can be improved.

When the Lagrangian multipliers $\boldsymbol{\alpha}$ are obtained in the first phase, the weights $\mathbf{w}_1$, $\mathcal{W}_2$, $\mathcal{W}_3$ and $\mathcal{W}_4$ in (38)-(41) can be obtained. Plugging these weights into the primal model in (34)-(36), the coefficients $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ in the kernel (50) can be obtained by solving the following LP problem

$$\min_{\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\xi},\tilde{b}} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\,\alpha_j y_i y_j K_1(\boldsymbol{\beta},\boldsymbol{\gamma},\mathbf{x}_{1(i)},\mathcal{X}_{2(i)},\mathcal{X}_{3(i)},\mathcal{X}_{4(i)}) + \tilde{C}\sum_{i=1}^{n}\xi_i \tag{54}$$

$$\text{s.t.} \quad y_i\left(\sum_{j=1}^{n}\alpha_j y_j K_1(\boldsymbol{\beta},\boldsymbol{\gamma},\mathbf{x}_{1(i)},\mathcal{X}_{2(i)},\mathcal{X}_{3(i)},\mathcal{X}_{4(i)}) + \tilde{b}\right) \geq 1-\xi_i \qquad i = 1,\cdots,n \tag{55}$$

$$\xi_i \geq 0 \qquad i = 1,\cdots,n \tag{56}$$

$$\beta_{\tilde{q}} \geq 0 \qquad \tilde{q} = 1,\cdots,4 \tag{57}$$

$$\gamma_{p_q} \geq 0 \qquad p_q = 1,\cdots,N_q, \tag{58}$$

where $K_1(\boldsymbol{\beta},\boldsymbol{\gamma},\mathbf{x}_{1(i)},\mathcal{X}_{2(i)},\mathcal{X}_{3(i)},\mathcal{X}_{4(i)})$ is given in (50). A two-phase strategy, which is nested in the above mentioned iterative strategy for the problem in (44)-(46), is adopted to solve the above LP problem. The coefficients $\boldsymbol{\gamma}$ are obtained in the first phase using fixed $\boldsymbol{\beta}$ and known $\boldsymbol{\alpha}$. The coefficients $\boldsymbol{\beta}$ are then obtained in the second phase using fixed $\boldsymbol{\gamma}$ and known $\boldsymbol{\alpha}$.

After the training of the MT-STM, the bias $\tilde{b}$ can be computed using any support tensor $\tilde{i}$

$$\tilde{b} = y_{\tilde{i}} - \sum_{i=1}^{n} \alpha_i y_i K_1(\boldsymbol{\beta},\boldsymbol{\gamma},\mathbf{x}_{1(\tilde{i})},\mathcal{X}_{2(\tilde{i})},\mathcal{X}_{3(\tilde{i})},\mathcal{X}_{4(\tilde{i})})\ . \tag{59}$$

When the kernel in (50) is used, the resulting classification function is

$$f(\mathbf{x}_{1(i_0)},\mathcal{X}_{2(i_0)},\mathcal{X}_{3(i_0)},\mathcal{X}_{4(i_0)}) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i K_1(\boldsymbol{\beta},\boldsymbol{\gamma},\mathbf{x}_{1(i_0)},\mathcal{X}_{2(i_0)},\mathcal{X}_{3(i_0)},\mathcal{X}_{4(i_0)}) + \tilde{b}\right), \tag{60}$$

for any observation i_0 with an input $\left(\mathbf{x}_{1(i_0)},\mathcal{X}_{2(i_0)},\mathcal{X}_{3(i_0)},\mathcal{X}_{4(i_0)}\right)$.

When the kernel in (52) is used, the coefficients $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\hat{\boldsymbol{\gamma}}$ can be obtained by solving the following LP problem

$$\min_{\boldsymbol{\beta},\boldsymbol{\gamma},\hat{\boldsymbol{\gamma}},\boldsymbol{\xi},\hat{b}} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\, \alpha_j y_i y_j K_2(\boldsymbol{\beta},\boldsymbol{\gamma},\hat{\boldsymbol{\gamma}},\mathbf{x}_{1(i)},\mathcal{X}_{2(i)},\mathcal{X}_{3(i)},\mathcal{X}_{4(i)}) + \tilde{C}\sum_{i=1}^{n} \xi_i \tag{61}$$

$$\text{s.t.} \quad y_i\left(\sum_{j=1}^{n} \alpha_j y_j K_2(\boldsymbol{\beta},\boldsymbol{\gamma},\hat{\boldsymbol{\gamma}},\mathbf{x}_{1(i)},\mathcal{X}_{2(i)},\mathcal{X}_{3(i)},\mathcal{X}_{4(i)}) + \hat{b}\right) \ge 1 - \xi_i \qquad i = 1,\cdots,n \tag{62}$$

$$\xi_i \ge 0 \qquad i = 1,\cdots,n \tag{63}$$

$$\beta_{\tilde{q}} \ge 0 \qquad \tilde{q} = 1,\cdots,4 \tag{64}$$

$$\gamma_{p_q} \ge 0 \qquad p_q = 1,\cdots,N_q \tag{65}$$

$$\hat{\gamma}_{Z_{p_q}} \ge 0 \qquad z_{p_q} = 1,\cdots,Z_{p_q}\ . \tag{66}$$

where $K_2(\boldsymbol{\beta},\boldsymbol{\gamma},\hat{\boldsymbol{\gamma}},\mathbf{x}_{1(i)},\mathcal{X}_{2(i)},\mathcal{X}_{3(i)},\mathcal{X}_{4(i)})$ is given in (52). Similarly, a three-phase strategy is adopted to solve this LP problem sequentially by solving three LP problems so as to obtain the coefficients $\hat{\boldsymbol{\gamma}}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$.

When the kernel in (52) is used, the resulting classification function is

$$f(\mathbf{x}_{1(i_0)},\mathcal{X}_{2(i_0)},\mathcal{X}_{3(i_0)},\mathcal{X}_{4(i_0)}) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i K_2(\boldsymbol{\beta},\boldsymbol{\gamma},\hat{\boldsymbol{\gamma}},\mathbf{x}_{1(i_0)},\mathcal{X}_{2(i_0)},\mathcal{X}_{3(i_0)},\mathcal{X}_{4(i_0)}) + \hat{b}\right) \tag{67}$$

for any observation i_0 with an input $\left(\mathbf{x}_{1(i_0)},\mathcal{X}_{2(i_0)},\mathcal{X}_{3(i_0)},\mathcal{X}_{4(i_0)}\right)$. The bias $\hat{b}$ can be computed using a way similar to (59) where $\tilde{b}$ is determined but with $K_1(\boldsymbol{\beta},\boldsymbol{\gamma},\mathbf{x}_{1(i_0)},\mathcal{X}_{2(i_0)},\mathcal{X}_{3(i_0)},\mathcal{X}_{4(i_0)})$ replaced by $K_2(\boldsymbol{\beta},\boldsymbol{\gamma},\hat{\boldsymbol{\gamma}},\mathbf{x}_{1(i_0)},\mathcal{X}_{2(i_0)},\mathcal{X}_{3(i_0)},\mathcal{X}_{4(i_0)})$.

5. Non-collaborative Ensemble Learning and Multiple SVM Ensemble

In this section, a NCEL framework is developed. Based on the framework, a data mining model, the M-SVM-E, is formulated to integrate multitype multiway data and perform feature selection from large sparse multiype multiway data in cross-selling.

5.1 The NCEL framework

The M-SVM-E, as a system using multiple SVM classifiers, is an alternative ensemble learning method. The M-SVM-E is trained through the NCEL framework. The NCEL framework for cross-selling recommendations using multitype multiway data includes the following components.

(1) Data sources. Four types of data like those in the CEL framework are used.

(2) Unfolding of multiway data. The multiway input data are transformed into multiple modes each of which is represented as a matrix.

(3) Training of the SVMs. Different modes of the multitype multiway data are used as inputs of different SVMs and each SVM is individually trained. As mentioned before, there are 1, N_2, N_3 and N_4 modes for the demographic, related product, similar customer and historical promotion data, respectively. Therefore, $\hat{N} = (1 + N_2 + N_3 + N_4)$ SVMs are in the system of multiple SVM classifiers. A standard SVM training procedure is used to train the SVMs of different modes to obtain the Lagrangian multipliers

(4) Learning the weights of multiple SVMs. The LP boosting method is used to obtain the weights of these SVMs by solving a LP problem.

The NCEL framework is based on the M-SVM-E as illustrated in Fig. 2. The framework consists of four main components: data sources, unfolding, base learner training and base learner ensemble. Different from the CEL framework, each base learner for the NCEL framework is individually trained and the training and ensemble of the base learners are performed in two separate phases. Classification methods such as the SVMs, artificial neural networks and classification trees can be used as the base learners. Besides LP boosting, other ensemble methods such as majority voting (MV), weighted majority voting (WMV), mean (M), weighted average (WA), decision templates and Dempster-Shafer evidence theory can be used to combine the base learners (Polikar, 2006). When a base learner is not a kernel method such as the SVM, each mode of the multiway data needs to be further vectorized so as to be used as the input of the base learner.

Fig. 2 approximately here

5.2 The M-SVM-E

The training dataset of the SVM for the demographic data is $\tilde{G}_1 = \{(\mathbf{x}_{1(1)}, y_{(1)}), \cdots, (\mathbf{x}_{1(n)}, y_{(n)})\}$. The training dataset of each of the SVMs for the other three types of multiway data is $\tilde{G}_q = \{(\mathcal{X}_{q(1)}^{<p_q>}, y_{(1)}), \cdots, (\mathcal{X}_{q(n)}^{<p_q>}, y_{(n)})\}$, where $\mathcal{X}_{q(i)}^{<p_q>}$ denotes the $p_q -$mode matrix of the tensor $\mathcal{X}_{q(i)}$, for $q = 2, \cdots, 4$ and $p_q = 1, \cdots, N_q$. There are $\hat{N}$ SVMs in the M-SVM-E. For the three types of multiway data, the $N_2 + N_3 + N_4$ training datasets are indexed as

$$g = \underbrace{2, \cdots, 1 + N_2}_{\mathcal{X}_{2(i)}}, \underbrace{2 + N_2, \cdots, (1 + N_2 + N_3)}_{\mathcal{X}_{3(i)}}, \underbrace{(2 + N_2 + N_3), \cdots, \hat{N}}_{\mathcal{X}_{4(i)}}, \tag{68}$$

where the related product data $\mathcal{X}_{2(i)}$, similar customer data $\mathcal{X}_{3(i)}$ and historical promotion data $\mathcal{X}_{4(i)}$ of observation i have N_2, N_3 and N_4 modes and thus training datasets, respectively. The connections among g, q and p_q are specified as follows

$$g = \begin{cases} 1 + p_2, & p_2 = 1, \cdots, N_2;\ q = 2 \\ 1 + N_2 + p_3, & p_3 = 1, \cdots, N_3;\ q = 3 \\ 1 + N_2 + N_3 + p_4, & p_4 = 1, \cdots, N_4;\ q = 4 \end{cases}. \tag{69}$$

The QP model for the demographic data is a standard SVM model

$$\max_{\boldsymbol{\alpha}^1} \quad \left\{ \sum_{i=1}^{n} \alpha_{(i)}^1 - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{(i)}^1 \alpha_{(j)}^1 y_i y_j k^1\left(\mathbf{x}_{1(i)}, \mathbf{x}_{1(j)}\right) \right\} \tag{70}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i \alpha_{(i)}^1 = 0 \tag{71}$$

$$0 \le \alpha_{(i)}^1 \le C \qquad i = 1, \cdots, n, \tag{72}$$

where $\alpha_{(i)}^1$ for $i = 1, \cdots, n$ are the Lagrangian multipliers of the first SVM. In the following, $\boldsymbol{\alpha}^1$ represents the vector with all the elements $\alpha_{(i)}^1$ for $i = 1, \cdots, n$.

For the three types of multiway data, the gth SVM, for $g = 2, \cdots, \hat{N}$, solves the following QP problem

$$\max_{\boldsymbol{\alpha}^g} \quad \left\{ \sum_{i=1}^{n} \alpha_{(i)}^{g} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{(i)}^{g} \alpha_{(j)}^{g} y_i y_j k^g \left(\mathcal{X}_{q(i)}^{<p_q>}, \mathcal{X}_{q(j)}^{<p_q>} \right) \right\} \tag{73}$$

$$\text{s.t.} \quad \sum_{i=1}^{\tilde{n}} y_i \alpha_{(i)}^{g} = 0 \tag{74}$$

$$0 \le \alpha_{(i)}^{g} \le C \qquad i = 1, \cdots, n \,, \tag{75}$$

where $\alpha_{(i)}^{g}$ for $i = 1, \cdots, n$ are the Lagrangian multipliers of the gth SVM. In the following, $\boldsymbol{\alpha}^g$ is used to represent a vector with components $\alpha_{(i)}^{g}$ for $i = 1, \cdots, n$ and for $g = 2, \cdots, \hat{N}$. The following unweighted hierarchical kernel is used in the gth SVM in (73)-(75)

$$k^g (\mathcal{X}_{q(i)}^{<p_q>}, \mathcal{X}_{q(j)}^{<p_q>}) = \sum_{z_{p_q}=1}^{Z_{p_q}} k_q^{p_q, z_{p_q}} \left(\mathcal{X}_{q(i, z_{p_q})}^{<p_q>}, \mathcal{X}_{q(j, z_{p_q})}^{<p_q>} \right). \tag{76}$$

The following projection kernel can also be used

$$k^g (\mathcal{X}_{q(i)}^{<p_q>}, \mathcal{X}_{q(j)}^{<p_q>}) = \exp\left(-\frac{1}{2\sigma^2} \left\| \mathbf{V}_{\mathcal{X}_{q(i),1}}^{<p_q>} \mathbf{V}_{\mathcal{X}_{q(i),1}}^{<p_q>\mathrm{T}} - \mathbf{V}_{\mathcal{X}_{q(j),1}}^{<p_q>} \mathbf{V}_{\mathcal{X}_{q(j),1}}^{<p_q>\mathrm{T}} \right\|_F^2 \right). \tag{77}$$

An important issue in the M-SVM-E is how to combine the local results of multiple individual SVMs to obtain better classification performance. Selective ensemble, as an ensemble strategy, refers to combining the outputs of some instead of all base learners to achieve good performance (Zhou *et al.*, 2002). Sparse ensemble, as a special case of selective ensemble, refers to combining the outputs of all base learners using a sparse weighting vector (Zhang and Zhou, 2011). Hence, only base learners with nonzero weights contribute to the final results of the ensemble. Sparse ensemble is used in the M-SVM-E in this study. The LP boosting method minimizes the L_1-norm soft margin error function (Demiriz *et al.*, 2002). As a sparse ensemble method, it can select the best combination of multiple base learners.

The weight of the base leaner g is represented by β_g and the vector of all base leaners is represented by $\hat{\boldsymbol{\beta}}$ with elements β_g for $g = 1, \cdots, \hat{N}$. A two-phase strategy is employed to train the M-SVM-E. When the weights $\hat{\boldsymbol{\beta}}$ of the base learners are fixed, each of the $\hat{N}$ QP problems in (70)-(72) and in (73)-(75) is solved individually in the first phase with standard SVM training procedures to obtain the Lagrangian multipliers $\boldsymbol{\alpha}^g$ for $g = 1, \cdots, \hat{N}$. When the Lagrangian multipliers $\boldsymbol{\alpha}^g$ for

$g = 1, \cdots, \hat{N}$ are fixed, the weights $\hat{\boldsymbol{\beta}}$ can be obtained in the second phase to combine the local results of the $\hat{N}$ individual SVMs by solving the following LP problem

$$\min_{\hat{\boldsymbol{\beta}}, \boldsymbol{\xi}} \quad \sum_{g=1}^{\hat{N}} \hat{\beta}_g + \lambda \sum_{i=1}^{n} \xi_{(i)} \tag{78}$$

$$\text{s.t.} \quad y_{(i)} \left(\hat{\beta}_1 \sum_{j=1}^{n} \alpha_{(j)}^{1} y_{(j)} k(\mathbf{x}_{1(i)}, \mathbf{x}_{1(j)}) + \sum_{g=2}^{\hat{N}} \hat{\beta}_g \sum_{j=1}^{n} \alpha_{(j)}^{g} y_{(j)} k^{g} \left(\mathcal{X}_{q(i)}^{<p_q>}, \mathcal{X}_{q(j)}^{<p_q>} \right) + \hat{b} \right) \geq 1 - \xi_i \tag{79}$$

$$i = 1, \cdots, n$$

$$\xi_i \geq 0 \qquad i = 1, \cdots, n \tag{80}$$

$$\hat{\beta}_g \geq 0 \qquad g = 1, \cdots, \hat{N} . \tag{81}$$

where $\hat{b}$ is the bias. The resulting classification function is

$$f(\mathbf{x}_{1(i_0)}, \mathcal{X}_{2(i_0)}, \mathcal{X}_{3(i_0)}, \mathcal{X}_{4(i_0)}) = \operatorname{sgn}\left(\hat{\beta}_1 \sum_{i=1}^{n} \alpha_{(i)}^{1} y_{(i)} k(\mathbf{x}_{1(i)}, \mathbf{x}_{1(i_0)}) + \sum_{g=2}^{\hat{N}} \hat{\beta}_g \sum_{i=1}^{n} \alpha_{(i)}^{g} y_{(i)} k^{g} \left(\mathcal{X}_{q(i)}^{<p_q>}, \mathcal{X}_{q(i_0)}^{<p_q>} \right) + \hat{b} \right), \tag{82}$$

for any observation i_0 with an input $\left(\mathbf{x}_{1(i_0)}, \mathcal{X}_{2(i_0)}, \mathcal{X}_{3(i_0)}, \mathcal{X}_{4(i_0)} \right)$. Because each of the $\hat{N}$ SVMs in (70) -(72) and in (73)-(75) has its bias, the bias $\hat{b}$ can be computed by averaging the biases of the SVMs.

6. Computational Experiments

Two databases, AW-Customers and AW-Resellers, are used to test the performance of the MK-STM and the M-SVM-E as well as the CEL and NCEL frameworks. Both of the databases are extracted from the open access databases AdventureWorksDW[1] in Microsoft SQL Server 2005 (Chen *et al.*, 2012). The characteristics of the two databases are shown in Tables 1 and 2, respectively.

Tables 1-2 approximately here

The computational experiments consist of the following steps: data preparation, data preprocessing, model training, parameter selection and model testing. These steps are described in the following.

Data preparation. The original datasets in these two databases are transformed into customer-centered datasets. Both of the transformed databases consist of four, *i.e.*, the demographic, related product, similar customer and historical promotion, datasets. For the AW-Customers database, the variables in the demographic dataset include annual income, total number of children, number of children at home, occupation and the number of automobiles owned. For the AW-Resellers database,

[1] Available at http://msftdbprodsamples.codeplex.com/releases/view/55330.

the variables in the demographic dataset include the number of employees, annual sales and the number of years in business. For these two databases, the RFM variables in the related product and similar customer datasets include the amount spent on the product category (Sales) and the number of products purchased in the specific product category (Quantity) per month by each customer, and the variable in the historical promotion dataset is the number of promotions received per month by each customer. The output of the models is whether or not a customer will purchase a specific product which has not been purchased by the customer over the next three months.

The related product ratio of a specific product, say product A, is defined as the number of units of each product other than product A purchased by the customers who purchased product A divided by that purchased by the customers who did not purchase product A in the training set. The products with high related product ratios are selected as the related products. The customers who have similar Sales and Quantity to a specific customer in the training dataset are the similar customers of the specific customer. The characteristics and more details of the transformed databases are shown in Table 3.

Table 3 approximately here

Data preprocessing. The input data are normalized and the observations with missing values are deleted. The related product, similar customer and historical promotion datasets are transformed into matrices with multiple modes through unfolding. A holdout validation method is used for the AW-Customers database. Each transformed dataset in the AW-Customer database is randomly partitioned into a training set, a validation set and a testing set. A five-fold cross validation method is used for the AW-Reseller database.

Model training. For the MK-STM, the Lagrangian multipliers $\boldsymbol{\alpha}$ are obtained by solving the QP problem in (44)-(46). The Gaussian kernel (53) is used as the basic kernel. When the projection kernel (50) is used, the coefficients $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are obtained by solving the LP problem in (54)-(58). When the hierarchical kernel in (52) is used, the coefficients $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\hat{\boldsymbol{\gamma}}$ are obtained by solving the LP problem in (61)-(66). For the M-SVM-E, one SVM is trained to model the demographic data by solving the QP problem in (70)-(72) and $\hat{N}-1$ individual SVMs are trained to model the multitype multiway data by solving the QP problem in (73)-(75). The outputs of all the individual SVMs are combined to obtain the final results by solving the LP problem in (78)-(81).

Parameter tuning and model testing. The grid-search method (Chen *et al.*, 2012) is used to determine the values of the parameters including C, $\tilde{C}$ and $1/\sigma^2$ in the MK-STM as well as C and $1/\sigma^2$ in the M-SVM-E. The validation set is used for parameter tuning. The trained models with the best parameter values are applied to the testing set to obtain the classification results.

Seven criteria are used to evaluate the classification performance of the models including 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) and the computational time (Time).

7. Computational Results

The computational experiments are carried out in the Matlab 7.4 development environment. The laptop computer used for the computation has an Intel Core i7 processor with a 2.80 GHz clock speed and has 4GB of RAM. The computational results of the MK-STM, the M-SVM-E and some other ensemble learning methods are reported in this section. Comparisons of results of the ensemble learning methods and some other supervised tensor learning methods for cross-selling using multitype multiway data are also reported.

7.1 Performance of the MK-STM

The results of the MK-STM using the projection kernel (50) and the hierarchical kernel (52) on the AW-Customers and AW-Resellers databases are reported in Table 4. As shown in Table 4, the MK-STM with the hierarchical kernel obtained an AUC of 91.39 and a Lift of 3.01 on the AW-Customer database and an AUC of 71.09 and a Lift of 2.44 on the AW-Reseller database. The MK-STM with the hierarchical kernel obtained better AUC and Lift than the MK-STM with the projection kernel. Moreover, the Time of the MK-STM with the hierarchical kernel is far less than that with the projection kernel.

Table 4 approximately here

7.2 Performance of the M-SVM-E and some other ensemble methods

The results of the M-SVM-E using the hierarchical kernel on the AW-Customers and AW-Resellers databases are reported in Tables 5 and 6, respectively. The results of the MK-STM using the hierarchical kernel on these two databases are also listed in these two tables.

Four ensemble methods, *i.e*., the MV, WMV, M and WA, are used to combine the local results of multiple individual SVMs. Their results are compared with those of the MK-STM and the M-SVM-E in Tables 5 and 6. For these four ensemble methods, SVMs are used as the base learners. The SVM with the Gaussian kernel (47) is used for the demographic data. The SVMs with the unweighted hierarchical kernel (76) are used for the multitype multiway data.

For the WMV and WA, the weights of the base learners can be the normalized classification rates of the base learners in the training set or can be those in the weighting strategy in the Adaboost algorithm (Polikar, 2006). By comparing their classification performance, normalized classification rates of the base learners in the training set are used as the weights of the base learners.

It can be seen from Table 5 that the MK-STM obtained far higher AUC and Lift and used much shorter Time than other methods on the AW-Customer database. The M-SVM-E obtained almost the same AUC and used the same Time as the MV, WMV, M and WA. The MV and WMV obtained far lower Lift than other methods.

It can be seen from Table 6 that the MK-STM obtained the second highest AUC and the highest Lift and used the shortest Time, the M obtained the highest AUC and the M-SVM-E obtained the second highest Lift on the AW-Reseller database. The MK-STM used the shortest and the M-SVM-E used the longest Time.

The results in Tables 5 and 6 show that the MK-STM exhibits the best performance on these two databases. These results also show that the M-SVM-E has almost the same performance as the M and WA, while the M and WA have better performance than the MV and WMV.

Tables 5 and 6 approximately here

7.3 Comparisons of different supervised tensor learning methods

The TK, SHTM and STM are three typical supervised tensor learning methods. For the TK, the unweighted hierarchical kernel (76) is used as the basic kernel of the multiplicative kernel in (28). It should be noted that the TK, SHTK and STM can't directly deal with multitype multiway data. Therefore, the related product data and the historical promotion data are merged as a fourth-order tensor which is used as the input of these methods. The results of the TK, SHTK and STM and those of the MK-STM and M-SVM-E on the AW-Customers and AW-Resellers databases are compared in Tables 7 and 8, respectively.

As shown in tables 7 and 8, the MK-STM obtained the highest AUC and Lift on these two databases, the M-SVM-E obtained the second highest AUC on these two databases and the TK obtained the second highest Lift on the AW-Customers database. Although the SHTM took shorter Time, its AUC and Lift are far lower than those of the MK-STM, M-SVM-E and TK due to the use of the linear kernel.

Furthermore, the STM (Tao *et al*., 2007) is a popular supervised tensor learning method. Like the SHTM, the STM uses the linear kernel, and thus is not suitable for nonlinear classification. Using the alternating projection optimization procedure (Tao *et al*., 2007), the computational complexity of the STM is $O(MNn^3)$ where M denotes the number of iterations, N, as used in Section 3, denotes the number of modes of the multiway data and n, as used earlier, is the number of observations in the training set. The computational complexity of the MK-STM, SHTM and TK is the same as that of the SVM, *i.e*., $O(n^3)$. The computational complexity of the M-SVM-E and the four other ensemble learning methods, *i.e*., the MV, WMV, M and WA, is $O(\hat{N}n^3)$ where $\hat{N}$, as used earlier, denotes the number of base learners and $\hat{N} = N$ when only one type of multiway data is used. By comparison, the STM has the highest computational complexity, and thus the highest computational cost.

Tables 7-8 approximately here

8. Conclusions

In this study, two ensemble learning frameworks, the CEL and the NCEL, one collaborative and the other non-collaborative, are developed. Based on these two frameworks, two ensemble learning models, the MK-STM and the M-SVM-E, are proposed as data mining tools for cross-selling using multitype multiway data. In comparison with existing supervised tensor learning methods, the major contributions of this study are (1) multitype multiway data are incorporated into the learning models for cross-selling to improve classification performance so as to improve customer response rate; (2) two novel ensemble learning methods, the MK-STM and the M-SVM-E, are proposed to deal with multitype multiway data and select features with good discriminative abilities from large sparse multitype multiway data; (3) two ensemble learning frameworks, the CEL and the NCEL, are developed to apply the classification and ensemble methods for supervised learning with multitype multiway data represented by tensors.

Computational experiments are conducted on two databases extracted from open access databases. The experimental results show that (1) the MK-STM exhibits the best performance; (2) the ensemble learning methods including the MK-STM, M-SVM-E, M and WA using SVMs as base learners have better performance than the existing supervised tensor learning methods including the TK and SHTM; and (3) the MK-STM, SHTM and TK are the methods with low computational cost.

There are a few directions for further research. The association rules and customer segmentation methods can be used in the ensemble learning frameworks to select the related products and similar customers. In the age of big data, the ensemble learning frameworks and methods, as data mining tools, can be applied to integrate multitype multiway data in the fields of CRM and direct marketing in social media and social commerce. They can also be extended to the problems of regression, clustering and semi-supervised learning.

Acknowledgements:

This work was partially supported by the National Natural Science Foundation of China (Project No. 71101023, 71021061 and 71271051) and the Fundamental Research Funds for the Central Universities, Northeastern University, China (Project No. N120406001 and N110706001).

References

Adomavicius, G., Tuzhilin, A., (2005). Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, **17**(6), 734–749.

Ahn, H, Ahn, J. J., Oh, K. J., Kim, D. H., (2011). Facilitating cross-selling in a mobile telecom market to develop customer classification model based on hybrid data mining techniques. *Expert Systems* *with Applications*, **38**(5), 5005–5012.

Ansell, J., Harrison, T., Archibald, T., (2007). Identifying cross-selling opportunities, using lifestyle segmentation and survival analysis. *Marketing Intelligence & Planning*, **25**(4), 394–410.

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

Chen, Z.-Y., Fan, Z.-P., (2012). Distributed customer behavior prediction using multiplex data: A collaborative MK-SVM approach. *Knowledge-Based Systems*, **35**, 111–119.

Chen, Z.-Y., Fan, Z.-P., Sun, M., (2012). A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *European Journal of Operational Research*, **223**(2), 461–472.

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**(2), 161–175.

Christmann, A., Hable, R. (2012). Consistency of support vector machines using additive kernels for additive models. *Computational Statistics & Data Analysis*, **56**(4), 854–873.

Cui, D., Curry, D. (2005). Prediction in marketing using the support vector machine. *Marketing Science*, **24**(4), 595-615.

Demiriz, A., Bennett, K. P., Shawe-Taylor, J., (2002). Linear programming boosting via column generation. *Machine Learning*, **46**(1-3), 225–254.

Dietterich, T. G., (2000). Ensemble methods in machine learning. In *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, Cagliari, Italy, pp. 1–15.

Hao, Z., He, L., Chen, B., Yang, X., (2013). A linear support higher-order tensor machine for classification. *IEEE Transactions on Image Processing*, **22**(7), 2911–2920.

Hoff, P. D., (2011). Hierarchical multilinear models for multiway data. *Computational Statistics and Data Analysis*, **55**(1), 530–543.

Kamakura, W. A., Kossar, B. S., Wedel, M., (2004). Identifying innovators for the cross-selling of new products. *Management Science*, **50**(8), 1120–1133.

Knott, A., Hayes, A., Neslin, S. A. (2002). Next-product-to-buy models for cross-selling applications.

Journal of Interactive Marketing, **16**(3), 59–75.

Kolda, T., Bader, B., (2009). Tensor decompositions and applications. *SIAM review*, **51**(3), 455–500.

Li, S., Sun, B., Montgomery, A. (2011). Cross-selling the right product to the right customer at the right time. *Journal of Marketing Research*, **48**(4), 683–700.

Lu, H., Plataniotis, K, N., Venetsanopoulos, A. N. (2011). A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, **44** (7), 1540–1551.

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**(2), 2592–2602.

Polikar, P., (2006). Ensemble based systems for decision making. *IEEE Circuits and System Magazine*, **6**(3), 21–45.

Prinzie, A., Van den Poel, D., (2006). Investigating purchasing sequence patterns for financial services using Markov, MTD and MTDg models. *European Journal of Operational Research*, **170**(3), 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**(1), 28–45.

Prinzie, A., Van den Poel, D. (2011). Modeling complex longitudinal consumer behavior with dynamic Bayesian networks: an acquisition pattern analysis application. *Journal of Intelligent Information System*, **36**(3), 283–304.

Rust, R. T., Chung, T. S. (2006). Marketing models of service and relationship. *Marketing Science*, **25**(6), 560-580.

Signoretto, M., De Lathauwer, L., Suykens, J. A. K., (2011). A kernel-based framework to tensorial data analysis. *Neural Networks*, **24**(8), 861–874.

Tao, D., Li, X., Wu, X., Hu, W., Maybank, S. J., (2007). Supervised tensor learning. *Knowledge and Information Systems*, **13**(1), 1–42.

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**(3), 2354–2364.

Zhang, L., Zhou, W. D., (2011). Sparse ensembles using weighted combination methods based on linear programming. *Pattern Recognition*, **44**(1), 97–106.

Zhou, Z.-H., (2012). *Ensemble methods: Foundations and algorithms*. Chapman & Hall/CRC press.

Zhou, Z.-H., Wu, J., Tang, W., (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, **137**(1-2), 239–263.

Zhu, X., Li, B., Wu, X., He, D., Zhang, C., (2011). CLAP: Collaborative pattern mining for distributed information systems. *Decision Support Systems*, **52**(1), 40–51.

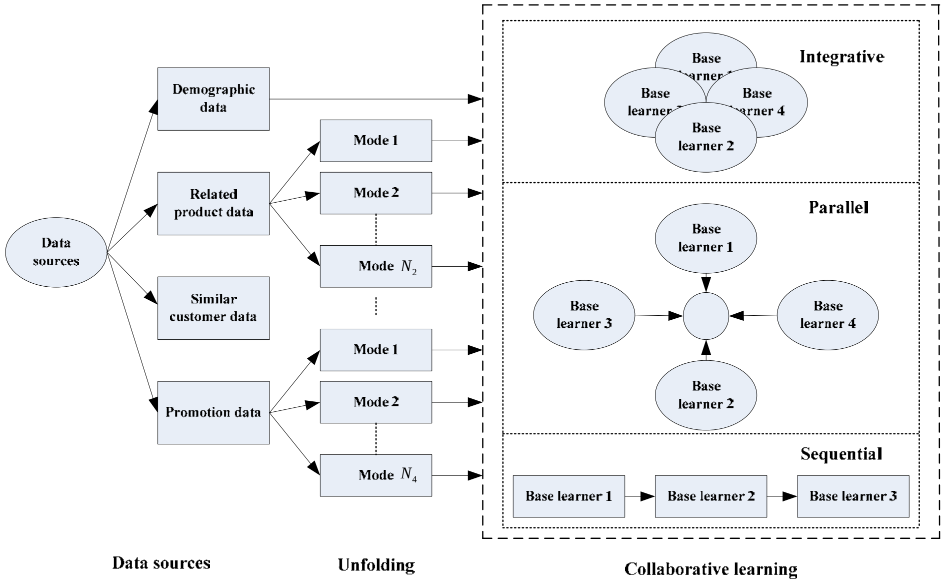

Fig. 1. The CEL framework for cross-selling using multitype multiway data

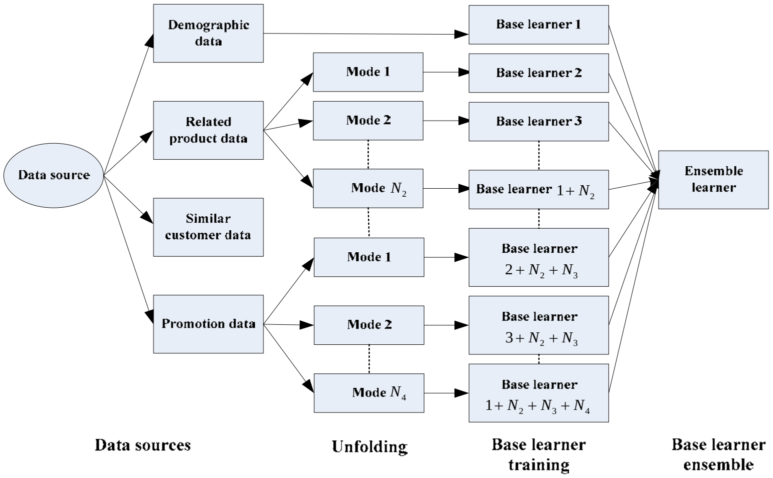

Fig. 2. The NCEL framework for cross-selling using multitype multiway data

Table 1. Characteristics of the AW-Customers database

Datasets	Records	Customer-centered	Timestamp
Customers	18,485	Yes	No
Customer sales	60,398	No	Yes
Product	37	No	No
Promotion	16	No	No
Time	1158	No	No

Table 2. Characteristics of the AW-Resellers database

Datasets	Records	Customer-centered	Timestamp
Resellers	701	Yes	No
Reseller sales	60,855	No	Yes
Product	37	No	No
Promotion	16	No	No
Time	1158	No	No

Table 3. Characteristics of the transformed databases

Databases	n	Demographic data	Related product data	Similar customer data	Historical promotion data
AW-Customers	4649	$m_1 = 5$	$m_2 = 2\,; m_3 = 2$ $T_1 = 24$	$m_4 = 2\,; m_5 = 1\,;$ $m_6 = 2\,; T_2 = 24$	$m_7 = 2\,; m_8 = 1\,;$ $T_3 = 24$
AW-Resellers	359	$m_1 = 3$	$m_2 = 2\,; m_3 = 2$ $T_1 = 18$	$m_4 = 2\,; m_5 = 1\,;$ $m_6 = 2\,; T_2 = 24$	$m_7 = 1\,; m_8 = 1\,;$ $T_3 = 18$

Table 4. Performance of the MK-STM using the hierarchical and projection kernel on the AW-Customers and AW-Resellers databases

Databases	Kernel	PCC	Sensitivity	Specificity	AUC	Lift	Time(s)
AW-Customers	Hierarchical	88.20	69.91	93.54	91.39	3.01	313.94
AW-Customers	Projection	70.80	8.85	88.89	47.20	0.80	4350.20
AW-Resellers	Hierarchical	55.56	83.33	49.38	71.09	2.44	147.88
AW-Resellers	Projection	47.47	83.33	39.51	64.54	1.83	880.74

Table 5. Comparisons of the MK-STM, the M-SVM-E and some other ensemble methods on the AW-Customers database

Classifiers	Ensemble	PCC	Sensitivity	Specificity	AUC	Lift	Time
MK-STM		88.20	69.91	93.54	91.39	3.01	313.94
M-SVM-E		50.20	48.67	50.65	56.78	1.33	409.98
SVM	MV	76.80	0.00	99.22	55.23	0.00	406.60
SVM	WMV	77.00	0.88	99.22	55.48	0.09	406.60
SVM	M	52.80	45.13	55.04	56.87	1.24	406.60
SVM	WA	68.20	33.63	78.29	57.00	1.33	406.60

Table 6. Comparisons of the MK-STM, the M-SVM-E and some other ensemble methods on the AW-Resellers database

Classifiers	Ensemble	PCC	Sensitivity	Specificity	AUC	Lift	Time
MK-STM		55.56	83.33	49.38	71.09	2.44	147.88
M-SVM-E		51.51	88.88	43.21	68.48	1.93	1008.64
SVM	MV	20.20	100.00	2.47	63.34	1.22	564.43
SVM	WMV	20.20	100.00	2.47	62.31	0.61	564.43
SVM	M	22.22	100.00	4.94	72.43	1.22	564.43
SVM	WA	22.22	100.00	4.94	71.16	0.61	564.43

Table 7. Comparisons of the MK-STM, the M-SVM-E and some other tensor learning methods on the AW-Customers database

Methods	PCC	Sensitivity	Specificity	AUC	Lift	Time
MK-STM	88.20	69.91	93.54	91.39	3.01	313.94
M-SVM-E	50.20	48.67	50.65	56.78	1.33	409.98
TK	67.80	31.86	78.29	55.63	1.50	133.22
SHTM	78.20	10.62	97.93	32.27	1.06	253.69
STM	66.80	3.54	85.27	68.16	0.09	461.22

Table 8. Comparisons of the MK-STM, the M-SVM-E and some other tensor learning methods on the AW-Resellers database

Methods	PCC	Sensitivity	Specificity	AUC	Lift	Time
MK-STM	55.56	83.33	49.38	71.09	2.44	147.88
M-SVM-E	51.51	88.88	43.21	68.48	1.93	1008.64
TK	65.45	66.67	29.63	55.73	1.22	111.34
SHTM	81.81	0.00	100.00	51.23	0.00	65.07
STM	77.78	0.00	100.00	46.91	0.00	817.52