Probabilistic user behavior models in online stores for recommender systems

Tomoharu Iwata

Abstract

Recommender systems are widely used in online stores because they are expected to improve both user convenience and online store profit. As such, a number of recommendation methods have been proposed in recent years. Functions required for recommender systems vary significantly depending on business models or/and situations. Although an online store can acquire various kinds of information about user behaviors such as purchase history and visiting time of users, this information has not yet been fully used to fulfill the diverse requirements. In this thesis, we propose probabilistic user behavior models for use in online stores to tailor recommender systems to diverse requirements efficiently using various kinds of user behavior information. The probabilistic model-based approach allows us to systematically integrate heterogeneous user behavior information using rules of the probability theory. In particular, we consider three requirements for recommender systems: predictive accuracy, efficiency, and profitability.

Models that can accurately predict present user behavior, rather than past user behavior, are necessary for recommendations because behaviors may vary with time. We propose a framework for learning models that best describes present samples and apply the framework to learning choice models that predict the next purchase item. In the proposed framework, models are learned by minimizing a weighted error over time that approximates the expected error at the present time.

Efficiency is also an important issue for recommender systems because the systems need frequent updates in order to maintain high accuracy by handling a large number of purchase history data that are accumulated day by day. We present an efficient probabilistic choice model using temporal purchase order information. Fast parameter estimation and high predictive accuracy are achieved by combining multiple simple Markov models based on the maximum entropy principle.

For the profitability requirement, it may be important for online stores to improve customer lifetime value (LTV) rather than to predict future purchases accurately. We present a recommendation method for improving LTV by integrating probabilistic choice models and purchase frequency models. The proposed recommendation method finds frequent purchase patterns of high LTV users, and

recommends items that simulate the found patterns. In addition, the proposed recommendation method is extended to be applicable to subscription services by integrating probabilistic subscription period models with choice models. The effectiveness of the proposed methods is demonstrated via experiments using synthetic data sets and real purchase log data sets of online stores.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and motivation

With the rapid progress of network and database technologies, people can purchase any number of items from anywhere in the world through online stores. Recommender systems are important for online stores because they can help users to find items of interest from an enormous number of items, and they can also help online stores to promote sales by customizing items displayed for each user. In the recommender system used by Amazon.com [36], items are recommended with phrases such as "better together," "customers who bought this item also bought," or "related items." Recommender systems are also widely used in online stores for products such as movies, music and cartoons [55].

To fulfill the growing need for qualified recommender systems, a number of methods have been proposed in recent years [1]. A frequently-used method is nearest-neighbor collaborative filtering [50, 57], in which similarities between users are calculated by using ratings or purchase histories, and a user preference is predicted from the weighted average of similar user preferences. Content filtering is also used for recommendations that predicts interests using item information [40], such as the author of a book or the director of a movie. Hybrid methods of collaborative and contents filtering that integrate rating histories and item information have also been proposed [33, 45].

Functions required for recommender systems vary significantly depending on the business model or/and situation. First, online stores need to accurately predict the user's interests because the recommendation of least favorite items is useless. If numerous users visit a store at the same time, its recommender system is required to be able to deal with each user quickly and efficiently. If many new items are made available each day, the system should be easy to update. Online stores providing

measured services want to encourage users to purchase many items by recommendation. On the other hand, online stores that provide subscription services need to encourage users to extend their subscription periods.

Online stores can obtain various kinds of information about user behavior, such as how frequently did the user purchase items, is the user a loyal customer, when did the user start/stop visiting the online store, what kinds of items did the user like in the past/recently, and when did the user purchase a certain item, as well as ratings, purchase histories, and item information. This information can be used to better understand user behavior and to improve recommender systems. By understanding the interests of recent users, a store can recommend items effectively because a user's interests can change from day to day. By recommending items that are often purchased by loyal customers, a store can improve sales because such a recommendation strategy may increase customer loyalty. Although many recommendation methods have been proposed, as described above, such diverse information has not yet been fully used to fulfill the diverse requirements.

In the present thesis, probabilistic user behavior models for use in online stores are proposed to tailor recommender systems to diverse requirements. The focus of the present thesis is probabilistic models because heterogeneous user behavior information can be systematically integrated using the rules of the probability theory. Probabilistic models are often used for the integration of heterogeneous information, such as texts and images [6], authors [51], time [61], and citations [15], as well as the integration of ratings and item information for recommendations [45]. Most existing recommendation methods do not output probabilities and instead output degrees for user's interests, for instance. It is not simple to combine these degrees with other information in a principled manner in situations involving uncertainty. A probabilistic model for each behavior is constructed as a separate module such as item choices, purchase frequency, and stop visiting, and then these models are combined to fit for a certain purpose. Rather than describing various kinds of behaviors by one probabilistic model, combining modules can flexibly adapt to diverse requirements.

1.2 Overview

Three requirements for recommender systems are considered: predictive accuracy, efficiency, and profitability.

In recommender systems, models that can accurately predict present user behavior, rather than past user behavior, are needed because items that are appropriate for the present user should be recommended. User behavior in an online store may

change with time because items on sale may change from day to day owing to new releases and item withdrawals, and popular items may change according to changes in trends, seasons, and social and economic environments. In Chapter 2, a framework is presented for learning a model that best describes present samples given dynamically changing data, such as purchase log data, the distributions of which differ over time. The proposed method defines a weight for each sample that depends on its time of generation, and learns a model so as to minimize the weighted error of all samples over time. The weighted error over time is shown to approximate the expected error at the present time. Therefore, we can obtain a model for the present data by minimizing the weighted error. The proposed method can fit a model to the present data simply by weighting samples, without the need to modify the model to include the time structure of data. Experiments using synthetic data sets and a real purchase log data set of an online cartoon downloading service show the effectiveness of the proposed method for the analysis of dynamically changing data.

Efficiency is also an important issue for recommender systems because the systems need frequent update in order to maintain high accuracy by handling a large number of purchase log data that are accumulated daily. In Chapter 3, an efficient probabilistic choice model that allows quick update of the recommendation system is presented. Temporal purchase order information is used for item choice modeling because the purchase order information can be useful for choice prediction. For example, when the first volume of a series of DVD movies is purchased, the next purchase would be the second volume. In addition, the interests of users may change, in which case early purchase histories would not be as useful as recent purchase histories for predicting future purchases. Markov models and maximum entropy models have been used for choice models that predict the next purchase item using the purchase history as input. In Markov models, parameters can be estimated and updated quickly and efficiently, but predictions are not always accurate. On the other hand, the accuracy of maximum entropy models is generally high. However, the parameter estimation incurs a high computational cost. Both fast parameter estimation and high predictive accuracy are achieved by combining multiple simple Markov models based on the maximum entropy principle. Experiments using real log data sets of online music, movie, and cartoon downloading services show that the proposed method outperforms other conventional methods in the literature.

In Chapter 4, by combining multiple probabilistic user behavior models, a recommendation method to increase profits, which is one of the most important requirements for online stores, is constructed. The desired user behavior for online stores to increase profits differs depending on their business model. For exam-

ple, in a measured service, the desired behavior is to purchase items frequently, whereas in a subscription service, the desired behavior is long-term subscription. Recommendations are adapted to a business model using probabilistic models of purchase frequency and unsubscription in the case of measured and subscription services, respectively. In measured services, typical purchase patterns are identified for heavy users who purchase many items using purchase frequency models, and items are recommended for a new user using the typical patterns. Even though a recommended item is often purchased by heavy users, the recommendation is not useful if the user is not interested in the recommended item. Therefore, it is also necessary to take the user's interests into consideration by using choice models and combining these models with purchase frequency models so that we can make effective recommendations. This is an example of heterogeneous information integration using probabilistic models, where the information consists of purchase frequencies and user's interests. The recommendation method in order to increase profits for subscription services follows the same procedure as that for measured services, where purchase frequency models are replaced by subscription period models. The proposed method is evaluated using two sets of real log data for measured and subscription services.

1.3 Mathematical notations

Vectors are denoted by lower case bold letters, such as $\boldsymbol{x}$, and all vectors are assumed to be column vectors. Uppercase bold letters, such as $\boldsymbol{X}$, denote matrices or sets. $\boldsymbol{x}^T$ denotes the transpose of vector $\boldsymbol{x}$, and $|\boldsymbol{X}|$ denotes the number of elements in set $\boldsymbol{X}$. The function $I(x)$ is used to denote the indicator function as follows:

$$I(x) = \begin{cases} 1 & \text{if } x \text{ is true,} \\ 0 & \text{otherwise.} \end{cases} \tag{1.1}$$

A list of notations used in this thesis is provided in Table 1.1.

Table 1.1: Notation

$\boldsymbol{U}$	set of users
$\boldsymbol{S}$	set of items
N	number of users, $N = \|\boldsymbol{U}\|$
V	number of items, $V = \|\boldsymbol{S}\|$
u	user, or purchase history of user u
s	item
u_n	nth user, or purchase history of nth user $u_n = (s_1, \cdots, s_{K_n})$
K_n	number of purchased items of user u_n
s_{nk}	kth purchase item of user u_n
d_{nk}	kth purchase time of user u_n
u_{nk}	purchase history of user u_n at the kth purchase $u_{nk} = (s_{n1}, \cdots, s_{n,k-1})$

Chapter 2

Model learning for the latest data

2.1 Introduction

In recommender systems, models that can accurately predict present user behavior, rather than past user behavior, are needed because items that are appropriate for the present user should be recommended. User behaviors in a online store may change with time because items on sale may change from day to day owing to new releases and item withdrawals. Popular items may change according to changes in trends, seasons, and social and economic environments. In such cases, high predictive accuracy in relation to past data does not necessarily guarantee high predictive accuracy for present data.

In this chapter, we present a learning framework for obtaining a model that best describes present samples given dynamically changing data, such as purchase log data, the distributions of which differ over time. Let a sample consist of an input, $x \in \boldsymbol{X}$, an output, $y \in \boldsymbol{Y}$, and the time $d \in \{1, \cdots, D\}$ at which the sample (x, y) was generated. We assume that input x, output y, and time d are discrete variables. Namely, both $\boldsymbol{X}$ and $\boldsymbol{Y}$ consist of discrete symbols. We say that data $\{(x_m, y_m, d_m)\}_{m=1}^{M}$ is changing dynamically when the joint distribution of the input and output depends on the time as follows:

$$P(x, y|d) \neq P(x, y|d'), d \neq d', \tag{2.1}$$

where M is the number of samples. As an example, we can learn a probabilistic choice model that has high predictive accuracy for present data using the proposed framework. In the case of choice models, purchase history $u_{nk} = (s_{n1}, \cdots, s_{n,k-1})$ corresponds to input x_m, purchase item s_{nk} corresponds to output y_m, and purchase time d_{nk} corresponds to time d_m, in which index $m = k + \sum_{n'=1}^{n} K_{n'}$, where K_n is the number of purchased items of user u_n. A choice model represents the probability

of purchasing item y given purchase history x, $R(y|x)$, and it can be directly used for recommendations by suggesting an item $\hat{s}$ with the highest purchase probability as follows:

$$\hat{s} = \arg\max_{y \in \boldsymbol{S}} R(y|x), \tag{2.2}$$

where $\boldsymbol{S}$ represents the set of items. It can also be used as a module of a recommendation method for improving profits described in Chapter 4.

We focus on the problem of predicting the unknown output y generated at time D for an input x given dynamically changing data, where D represents the present time or the latest time in the data. The proposed method defines a weight for each sample that depends on its time of generation and learns a model so as to minimize the weighted error of all samples over time. The model distribution $\hat{P}(x, y|d)$ is calculated at each time d, and the weights are defined so that a mixture of the model distributions over time approximates the distribution at the present time $P(x, y|D)$. The weighted error function is shown to approximate the expected error at the present time. Therefore, we can obtain a model that fits data at the present time by minimizing the weighted error. In broad terms, the proposed method finds time points at which the data distribution is similar to that of the present time and also uses the data at the similar time points to learn the model for the present time. In general, the performance of models can be improved as the number of learning samples increases. By using the data at other time points, the proposed method can learn the model robustly when similar time points exist.

Dynamically changing data is usually modeled by explicitly incorporating the time structure of the data into the model (for example [7, 61]). The proposed method can fit a model to dynamically changing data simply by weighting samples without the need to modify the model to include the time structure of the data.

Here, a certain number of samples are assumed to be generated each time, and their model distributions can be calculated. Examples of such data include purchase log data, news stories, scientific articles, and web surfing data. With purchase log data, when the unit time is assumed as a day, many users purchase various items, and many samples (x, y) are generated each day. In this respect, the data considered herein are different from the data used in the conventional time-series analysis setting, in which only one sample is generated at a time.

The remainder of this chapter is organized as follows. In the next section, a basic model learning method based on error minimization and its problem for predictions of the present data are described. Section 2.3 presents the proposed method and its learning procedures. In Section 2.4, a brief review of related research is presented. In Section 2.5, the proposed method is evaluated experimentally using synthetic data sets and a real purchase log data set. Finally, this chapter is summarized and

future research is discussed in Section 2.6.

2.2 Model learning by error minimization

Let $\{(x_m, y_m, d_m)\}_{m=1}^{M}$ be a training data. If we can assume the stationarity of the training data, a model that fits the data can be learned by minimizing the following empirical error of all samples:

$$E_0(\mathcal{M}) = \sum_{m=1}^{M} J(x_m, y_m; \mathcal{M}), \tag{2.3}$$

where error function $J(x, y; \mathcal{M})$ represents the error of model $\mathcal{M}$ given sample (x, y). Typical examples of such error functions include the negative log likelihood:

$$J(x, y; \mathcal{M}) = -\log P(y|x; \mathcal{M}), \tag{2.4}$$

and the 0-1 loss function:

$$J(x, y; \mathcal{M}) = \begin{cases} 0 & \text{if } f(x; \mathcal{M}) = y, \\ 1 & \text{otherwise,} \end{cases} \tag{2.5}$$

where f is a regression function.

In this chapter, the focus is on finding a model that best describes the samples at the present time D given a dynamically changing data. A simple way to do this is to minimize the empirical error of samples only at the present time, as follows:

$$E_D(\mathcal{M}) = \sum_{m:d_m=D} J(x_m, y_m; \mathcal{M}). \tag{2.6}$$

As the number of samples at the present time $M(D)$ grows to infinity, the value $E_D(\mathcal{M})$ divided by $M(D)$ converges to the expected error at the present time, as follows:

$$\begin{aligned} \lim_{M(D)\to\infty} \frac{1}{M(D)} E_D(\mathcal{M}) &= \sum_{x\in\boldsymbol{X}} \sum_{y\in\boldsymbol{Y}} P(x, y|D) J(x, y; \mathcal{M}) \\ &= \mathcal{E}_D[J(x, y; \mathcal{M})], \end{aligned} \tag{2.7}$$

where $\mathcal{E}$ denotes the expectation. However, when we only use the present samples, the estimated model might have a tendency to overfit the training samples because there are fewer training samples.

2.3 Proposed method

2.3.1 Weighted error

To achieve more robust learning, both past and present data are used because past data often include information that is useful for fitting samples at the present time. Namely, the errors of all samples with weights that depend on the time at which the samples were generated are minimized, as follows:

$$E(\mathcal{M}) = \sum_{m=1}^{M} w(d_m) J(x_m, y_m; \mathcal{M}), \tag{2.8}$$

where $w(d)$ is the weight of a sample at time d. Intuitively, this represents the degree of usefulness of a sample at time d for fitting the model $\mathcal{M}$ at the present time D.

Since a model that best describes present samples is required, it is necessary to determine weights $\{w(d)\}_{d=1}^{D}$ so that the weighted error approximates the expected error at the present time, as follows:

$$E(\mathcal{M}) \;\approx\; \mathcal{E}_D[J(x, y; \mathcal{M})]. \tag{2.9}$$

To achieve this approximation, we determine weights $\{w(d)\}_{d=1}^{D}$ in the following manner. First, we estimate a model distribution $\hat{P}(x, y|d)$ for each time d that approximates the empirical distribution, as follows:

$$\hat{P}(x, y|d) \approx \frac{1}{M(d)} \sum_{m:d_m=d} I\big((x, y) = (x_m, y_m)\big), \tag{2.10}$$

where $M(d)$ is the number of samples at time d. For model distributions $\hat{P}(x, y|d)$, we can use arbitrary distributions that are appropriate for the given data, such as Gaussian, multinomial, and empirical distributions. Second, we estimate the mixture coefficients $\boldsymbol{P} = \{P(d)\}_{d=1}^{D}$ with $\sum_{d=1}^{D} P(d) = 1$ such that the following mixture of model distributions over time approximates the distribution at the present time $P(x, y|D)$, as follows:

$$P(x, y|D) \approx \sum_{d=1}^{D} P(d) \hat{P}(x, y|d). \tag{2.11}$$

See Figure 2.1 for the image of this approximation. Note that both model distributions at past times and the model distribution at the present time D are used to approximate the true distribution at the present time D. If the number of samples

at D is insufficiently large, the model distribution at D that is learned using only data at D might not be close enough to the true distribution. Third, we set the weight as follows:

$$w(d) = \frac{P(d)}{M(d)}. \tag{2.12}$$

By setting weights in this way, the weighted error (2.8) can approximate the expected error at the present time, as follows:

$$\begin{aligned}
E(\mathcal{M}) &= \sum_{m=1}^{M} w(d_m) J(x_m, y_m; \mathcal{M}) \\
&= \sum_{x \in \boldsymbol{X}} \sum_{y \in \boldsymbol{Y}} \sum_{d=1}^{D} w(d) \sum_{m: d_m = d} I\big((x, y) = (x_m, y_m)\big) J(x, y; \mathcal{M}) \\
&\approx \sum_{x \in \boldsymbol{X}} \sum_{y \in \boldsymbol{Y}} \sum_{d=1}^{D} w(d) M(d) \hat{P}(x, y|d) J(x, y; \mathcal{M}) \\
&= \sum_{x \in \boldsymbol{X}} \sum_{y \in \boldsymbol{Y}} \sum_{d=1}^{D} P(d) \hat{P}(x, y|d) J(x, y; \mathcal{M}) \\
&\approx \sum_{x \in \boldsymbol{X}} \sum_{y \in \boldsymbol{Y}} P(x, y|D) J(x, y; \mathcal{M}) \\
&= \mathcal{E}_D[J(x, y; \mathcal{M})].
\end{aligned} \tag{2.13}$$

Therefore, we can obtain a model that best describes data at the present time by minimizing the weighted error $E(\mathcal{M})$ given in (2.8). This method is referred to as the *Mixture Coefficient Weighting* (MCW) method.

Roughly speaking, this method finds time points that are similar to the present time, and uses their data to learn the model at the present time. The similarity is represented by the mixture coefficient $P(d)$. Data at similar time points have large mixture coefficients. The greater the number of similar time points, and thus similar data, that exist in the past, the more robustly the model at the present time can be learned by including these past data along with the present data. If there is no similar time point, the past data do not help to model the present data, and the prediction accuracy of the proposed method is comparable to that of the basic method, as in (2.6). However, time-series data often have a periodic nature, or temporarily close points exhibit similar behavior. In such cases, the proposed method is effective for improving the prediction accuracy at the present time. The proposed method can use arbitrary functional forms of the error functions and models as long as their objective functions to be minimized are written as in (2.3).

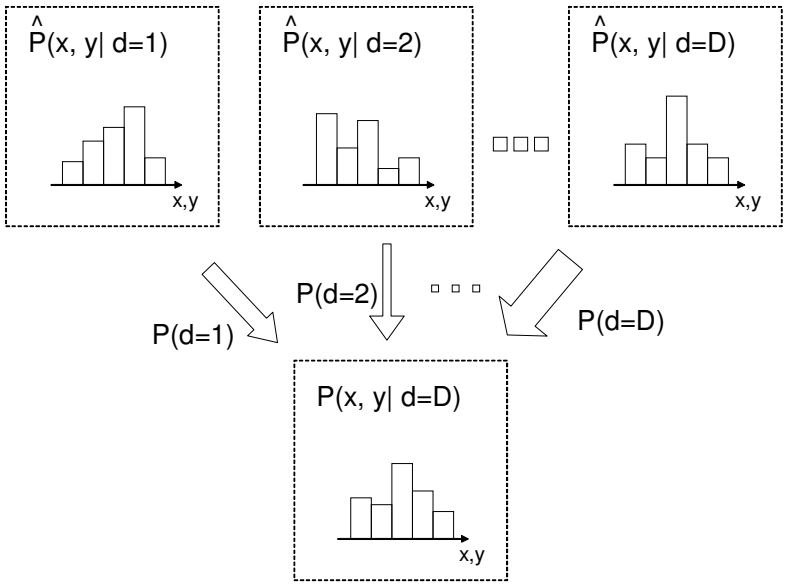


Figure 2.1: Approximation of the distribution at the present time by the mixture of empirical distributions over time.

2.3.2 Estimation of mixture coefficients

We can estimate mixture coefficients that satisfy (2.11) by maximizing the following log likelihood for samples at the present time:

$$\begin{aligned} L(\boldsymbol{P}) &= \sum_{m:d_m=D} \log P(x_m, y_m|D) \\ &= \sum_{m:d_m=D} \log \sum_{d=1}^{D} P(d)\hat{P}_{-m}(x_m, y_m|d), \end{aligned} \tag{2.14}$$

where $\hat{P}_{-m}(x, y|d)$ represents a model distribution at time d that is estimated using data excluding the mth sample. If we estimate mixture coefficients using training samples that are also used for estimating the model distribution, the estimation is biased, and we will obtain unuseful solutions $P(D) = 1$ and $P(d \neq D) = 0$. Therefore, we use leave-one-out cross-validation. Note that when we use an exponential family for the model distribution, the computational cost required for the leave-one-out estimation is no greater than that of the non-leave-one-out estimation by calculating sufficient statistics in advance. Note that the leave-one-out estimation at $d \neq D$ coincides with the non-leave-one-out estimation.

Since the logarithm is a convex function, $L(\boldsymbol{P})$ is also convex with respect to mixture coefficients $\boldsymbol{P}$. Therefore, we can obtain a globally optimum solution by

maximizing $L(\boldsymbol{P})$. Using the EM algorithm [14] allows us to estimate mixture coefficients that maximize $L(\boldsymbol{P})$. Let $\boldsymbol{P}^{(\tau)}$ be an estimate at the τth step. The conditional expectation of the complete-data log likelihood to be maximized is as follows:

$$Q(\boldsymbol{P}|\boldsymbol{P}^{(\tau)}) = \sum_{m:d_m=D} \sum_{d=1}^{D} P(d|x_m, y_m; \boldsymbol{P}^{(\tau)}) \log P(d)\hat{P}(x_m, y_m|d), \tag{2.15}$$

where $P(d|x_m, y_m; \boldsymbol{P}^{(\tau)})$ represents the posterior probability of the mth sample given the current estimate $\boldsymbol{P}^{(\tau)}$. In E-step, we compute the posterior probability with the Bayes rule:

$$P(d|x_m, y_m; \boldsymbol{P}^{(\tau)}) = \frac{P^{(\tau)}(d)\hat{P}(x_m, y_m|d)}{\sum_{d'=1}^{D} P^{(\tau)}(d')\hat{P}(x_m, y_m|d')}. \tag{2.16}$$

In M-step, we obtain the next estimate of the mixture coefficient $P^{(\tau+1)}(d)$ by maximizing $Q(\boldsymbol{P}|\boldsymbol{P}^{(\tau)})$ with respect to $P(d)$ subject to $\sum_{d=1}^{D} P(d) = 1$, as follows:

$$P^{(\tau+1)}(d) = \frac{1}{M(D)} \sum_{m:d_m=D} P(d|x_m, y_m; \boldsymbol{P}^{(\tau)}). \tag{2.17}$$

By iterating the E-step and the M-step until convergence, we obtain a global optimum solution for $\boldsymbol{P}$.

2.3.3 Procedure

The procedure of the proposed method is summarized as follows:

1. Estimate model distributions at each time d that approximate the empirical distribution as follows:

$$\hat{P}(x, y|d) \approx \frac{1}{M(d)} \sum_{m:d_m=d} I\big((x, y) = (x_m, y_m)\big). \tag{2.18}$$

2. Estimate mixture coefficients $\boldsymbol{P}$ so as to maximize the following maximum likelihood:

$$\hat{\boldsymbol{P}} = \arg\max_{\boldsymbol{P}} \sum_{m:d_m=D} \log \sum_{d=1}^{D} P(d)\hat{P}(x_m, y_m|d), \tag{2.19}$$

   using the EM algorithm as in (2.16) and (2.17).

3. Set weights as follows:

$$w(d) = \frac{\hat{P}(d)}{M(d)}, \tag{2.20}$$

for $1 \leq d \leq D$.

4. Obtain model $\mathcal{M}$ by minimizing the weighted error function as follows:

$$\hat{\mathcal{M}} = \arg\min_{\mathcal{M}} \sum_{m=1}^{M} w(d_m) J(x_m, y_m; \mathcal{M}), \tag{2.21}$$

while fixing weights $\{w(d)\}_{d=1}^{D}$.

The order of computational complexity of the proposed method is the same as that of the basic model learning procedure as in (2.3) once the weights are determined. Therefore, additional computational cost is needed only for the weight estimation. The complexity for each iteration of an EM-step is $O(M(D)D)$, which means that it increases linearly with the number of samples at the present time and the number of time points.

2.3.4 Extension to continuous variables

The input and output were assumed to be discrete variables. Based on the same framework, we can learn a model for the present data when the input and/or output are continuous variables by considering a mixture of continuous distributions at each time that approximates the continuous distribution at the present time. In the continuous case, the model distribution should be a continuous distribution such as Gaussian and Gaussian mixture distributions. The justification of the proposed weights follows the same lines as in the discrete case (2.13) by replacing the summations with integrations as required, as follows:

$$\begin{aligned}
E(\mathcal{M}) &= \sum_{m=1}^{M} w(d_m) J(x_m, y_m; \mathcal{M}) \\
&= \int_x \int_y \sum_{d=1}^{D} w(d) \sum_{m=1}^{M} I\big((x_m, y_m) = (x, y)\big) J(x, y; \mathcal{M}) dx dy \\
&\approx \int_x \int_y \sum_{d=1}^{D} w(d) M(d) \hat{P}(x, y|d) J(x, y; \mathcal{M}) dx dy \\
&= \int_x \int_y \sum_{d=1}^{D} P(d) \hat{P}(x, y|d) J(x, y; \mathcal{M}) dx dy
\end{aligned}$$

$$
\begin{aligned}
&\approx \int_x \int_y P(x,y|D) J(x,y;\mathcal{M}) dxdy \\
&= \mathcal{E}_D[J(x,y;\mathcal{M})]. \qquad (2.22)
\end{aligned}
$$

2.3.5 Weights when output distributions differ

The proposed method for a situation in which the joint distributions of the input and output differ over time $P(x,y|d) \neq P(x,y|d')$ have been described. When the output distributions are assumed to differ $P(y|d) \neq P(y|d')$ while the conditional distributions are equal over time $P(x|y,d) = P(x|y,d')$, the proposed method can be simplified. In this case, first, we estimate a model distribution of output y at each time, as follows:

$$
\hat{P}(y|d) \approx \frac{1}{M(d)} \sum_{m:d_m=d} I(y = y_m). \qquad (2.23)
$$

Next, we determine the mixture coefficients $\boldsymbol{P}$ such that a mixture of the model output distributions at each time approximates the output distribution at the present time $P(y|D)$ as follows:

$$
P(y|D) \approx \sum_{d=1}^{D} P(d) \hat{P}(y|d), \qquad (2.24)
$$

and set weights by

$$
w(d) = \frac{P(d)}{M(d)}. \qquad (2.25)
$$

Then, the weighted error approximates the expected error at the present time, as follows:

$$
\begin{aligned}
E(\mathcal{M}) &= \sum_{m=1}^{M} w(d_m) J(x_m, y_m; \mathcal{M}) \\
&= \sum_{x \in \boldsymbol{X}} \sum_{y \in \boldsymbol{Y}} \sum_{d=1}^{D} w(d) \sum_{m:d_m=d} I\big((x,y) = (x_m, y_m)\big) J(x,y;\mathcal{M}) \\
&\approx \sum_{x \in \boldsymbol{X}} \sum_{y \in \boldsymbol{Y}} \sum_{d=1}^{D} w(d) M(d) \hat{P}(x|y,d) \hat{P}(y|d) J(x,y;\mathcal{M}) \\
&\approx \sum_{x \in \boldsymbol{X}} \sum_{y \in \boldsymbol{Y}} P(x|y,D) \sum_{d=1}^{D} w(d) M(d) \hat{P}(y|d) J(x,y;\mathcal{M})
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{x\in\boldsymbol{X}}\sum_{y\in\boldsymbol{Y}} P(x|y,D)\sum_{d=1}^{D} P(d)\hat{P}(y|d)J(x,y;\mathcal{M}) \\
&\approx \sum_{x\in\boldsymbol{X}}\sum_{y\in\boldsymbol{Y}} P(x|y,D)P(y|D)J(x,y;\mathcal{M}) \\
&= \sum_{x\in\boldsymbol{X}}\sum_{y\in\boldsymbol{Y}} P(x,y|D)J(x,y;\mathcal{M}) \\
&= \mathcal{E}_D[J(x,y;\mathcal{M})], \qquad (2.26)
\end{aligned}
$$

where we used assumption $\hat{P}(x|y,d) \approx P(x|y,D)$ for all d. In the same way, we can simplify the proposed method when the input distributions differ $P(x|d) \neq P(x|d')$ while the conditional distributions are equal over time $P(y|x,d) = P(y|x,d')$.

2.4 Related research

A number of methods that attempt to model dynamically changing data have been proposed, in which the time structure of data is included in the models [7, 61]. Unlike these methods, the proposed method can fit the model to dynamically changing data simply by weighting samples, and there is no need to modify the model. Moreover, since the proposed method does not directly model the dynamics, such as the transition probability from $d-1$ to d in the Markov model, the proposed method is applicable even when data distributions change drastically.

One method for weighting samples according to the generated time is the exponential decay weighting method [16]:

$$
w(d) = \exp\bigl(-\lambda(D-d)\bigr), \qquad (2.27)
$$

where the weight decays corresponding to the period before the present time. Since the decay rate is constant, this method is not appropriate for periodic data.

In terms of weighing samples, the proposed method is related to the covariate shift or sample selection bias [4, 25, 60, 63]. Covariate shift refers to a situation, in which the input distribution of training samples $P(x|d)$ for $d \neq D$ is different from that of test samples $P(x|D)$ and the conditional probabilities do not change over time $P(y|x,d) = P(y|x,D)$. In order to learn models under the covariate shift, the importance weighted method has been proposed [58], in which samples are weighted depending on the ratio of the input likelihoods:

$$
w(d,x) = \frac{P(x|D)}{P(x|d)}. \qquad (2.28)
$$

After weights have been determined according to (2.28), this method minimizes the following function:

$$E_I(\mathcal{M}) = \sum_{m=1}^{M} w(d_m, x_m) J(x_m, y_m; \mathcal{M}). \tag{2.29}$$

When the number of samples at time d grows to infinity, the weighted error corresponding to time d, $E_I(\mathcal{M}, d)$ where $E_I(\mathcal{M}) = \sum_{d=1}^{D} E_I(\mathcal{M}, d)$, divided by $M(d)$ converges to the expected error at the present time, as follows:

$$\begin{aligned}
\lim_{M(d)\to\infty} \frac{1}{M(d)} E_I(\mathcal{M}, d) &= \lim_{M(d)\to\infty} \frac{1}{M(d)} \sum_{m:d_m=d} w(d, x_m) J(x_m, y_m; \mathcal{M}) \\
&= \sum_{x\in\boldsymbol{X}} \sum_{y\in\boldsymbol{Y}} P(x, y|d) \frac{P(x|D)}{P(x|d)} J(x, y; \mathcal{M}) \\
&= \sum_{x\in\boldsymbol{X}} \sum_{y\in\boldsymbol{Y}} P(x|d) P(y|x, d) \frac{P(x|D)}{P(x|d)} J(x, y; \mathcal{M}) \\
&= \sum_{x\in\boldsymbol{X}} \sum_{y\in\boldsymbol{Y}} P(x|D) P(y|x, D) J(x, y; \mathcal{M}) \\
&= \sum_{x\in\boldsymbol{X}} \sum_{y\in\boldsymbol{Y}} P(x, y|D) J(x, y; \mathcal{M}) \\
&= \mathcal{E}_D[J(x, y; \mathcal{M})],
\end{aligned} \tag{2.30}$$

where we used $P(y|x, d) = P(y|x, D)$. In this method, we need to know the true input distributions of the learning and test samples, or we need estimate the model distributions. Note that this method requires the estimation of weights for each pair of time and input (d, x), where the number of weights is $D|\boldsymbol{X}|$. Therefore, when the number of samples is small compared with the number of input elements $|\boldsymbol{X}|$, the weight estimation might not be robust because of data sparseness. On the other hand, we can robustly estimate weights with the proposed method because the weights depend only on time d, where the number of weights is D. The advantage of weighting of Equation (2.28) is that it can adjust weights according to the input when data distributions are partly similar and partly dissimilar to the present data over the input space.

Alternatively, if the output distributions are different $P(y|d) \neq P(y|D)$, and the conditional probabilities remain unchanged $P(x|y, d) = P(x|y, D)$, a method that weights samples depending on the ratio of the output distributions:

$$w(d, y) = \frac{P(y|D)}{P(y|d)}, \tag{2.31}$$

has been proposed [35].

We consider the relationship between the proposed method and the importance weighted method in detail. The joint distribution at D can be calculated using the joint distribution for each time and the likelihood ratio as follows:

$$P(x, y|D) = \frac{1}{D} \sum_{d=1}^{D} \frac{P(x, y|D)}{P(x, y|d)} P(x, y|d). \tag{2.32}$$

However, this is useless because the joint distribution at D is needed in the likelihood ratio. Under the covariate shift, $P(y|x, d) = P(y|x, D)$, the above equation can be rewritten as follows:

$$P(x, y|D) = \frac{1}{D} \sum_{d=1}^{D} \frac{P(x|D)}{P(x|d)} P(x, y|d). \tag{2.33}$$

Therefore, the ratio between input likelihoods is used for the weight in the importance weighted method, in which the input likelihoods $P(x|d)$ are usually estimated individually. On the other hand, in the proposed method, the ratio between joint likelihoods is parameterized by one parameter $P(d)$ as follows:

$$P(d) = \frac{1}{D} \frac{P(x, y|D)}{P(x, y|d)}, \tag{2.34}$$

where the parameter does not depend on input x and output y. The parameters $\boldsymbol{P} = \{P(d)\}_{d=1}^{D}$ are estimated so as to approximate the joint distribution at D without estimating distributions explicitly.

2.5 Experimental results

2.5.1 Synthetic data

We evaluated the proposed method using the following four sets of synthetic data:

- **Periodic:** The distribution changes periodically. A sample at time d is generated from the following distribution:

$$P(x, y|d) = \frac{\exp(\pi_{x,y,d})}{\sum_{x' \in \boldsymbol{X}} \sum_{y' \in \boldsymbol{Y}} \exp(\pi_{x',y',d})}, \tag{2.35}$$

  where $\pi_{x,y,d}$ is sampled from $\pi_{x,y,d} \sim \mathcal{N}(0, \gamma_0)$ if $0 \leq d \leq C - 1$, $\pi_{x,y,d} = \pi_{x,y,d \bmod C}$ otherwise, and $\mathcal{N}(\mu, \sigma^2)$ represents a Gaussian with mean μ and variance σ^2. Here, $\gamma_0 = 1.0$, and $C = 7$.

- **Random:** The distribution changes randomly. First, $\{\pi^g_{x,y}\}_{g=1}^G$ are sampled from $\pi^g_{x,y} \sim \mathcal{N}(0, \gamma_0)$. Then, at each time a random integer r with range $[1, G]$ is sampled, and a sample is generated as follows:

$$P(x, y|d) = \frac{\exp(\pi^r_{x,y})}{\sum_{x' \in \boldsymbol{X}} \sum_{y' \in \boldsymbol{Y}} \exp(\pi^r_{x',y'})}. \tag{2.36}$$

  Here, $G = 7$. This data can be also generated by randomly permutating Periodic data over time.

- **Gradual:** The distribution changes gradually with time. In (2.35), $\pi_{x,y,d}$ is sampled from $\pi_{x,y,d} \sim \mathcal{N}(0, \gamma_0)$ if $t = 0$, $\pi_{x,y,d} \sim \mathcal{N}(\pi_{x,y,d-1}, \gamma)$ otherwise. Here, $\gamma = 0.1$.

- **Drastic:** The distribution changes drastically every τ. In (2.35), $\pi_{x,y,d}$ is sampled from $\pi_{x,y,d} \sim \mathcal{N}(0, \gamma_0)$ if $d \mod \tau = 0$, $\pi_{x,y,d} = \pi_{x,y,d-1}$ otherwise. Here, $\tau = 30$.

We assumed that $\boldsymbol{X}$ and $\boldsymbol{Y}$ consist of ten kinds of symbols, and therefore $P(x, y|d)$ has 100 values at each d, as shown in Figure 2.2. Here, $D = 100$.

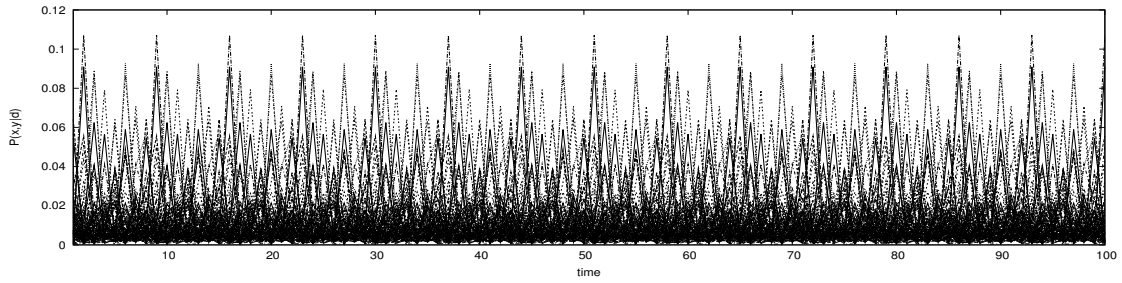
Empirical distributions with leave-one-out cross-validation were used as model distributions, as follows:

$$\hat{P}(x, y|d) = \frac{M(d, x, y) - \delta_{d,D}}{M(d) - \delta_{d,D}}, \tag{2.37}$$

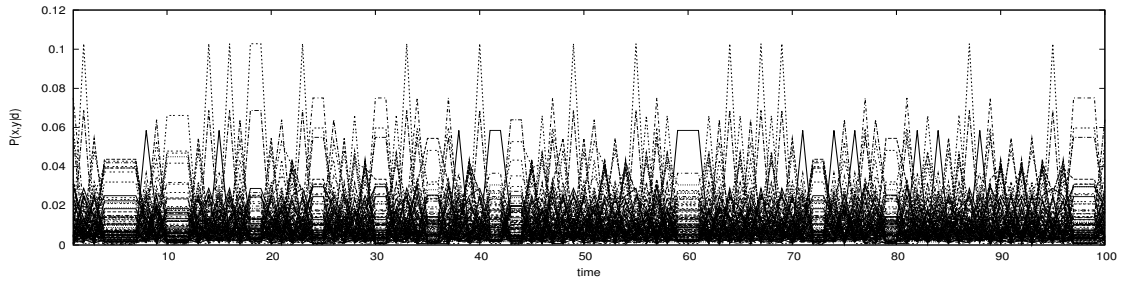
where $M(d, x, y)$ is the number of samples at d with input x and output y. In addition, $\delta_{d,D}$ is Kronecker's delta, i.e., $\delta_{d,D} = 1$ if $d = D$ and 0 otherwise, and is used for the leave-one-out cross-validation, where we simply subtract one from both of the numbers of samples $M(d, x, y)$ and $M(d)$ for the empirical distribution at the present time D. For practical reasons, we introduce Laplace smoothing parameter $\alpha = 10^{-8}$ to avoid the zero probability problem when we estimate the empirical distribution. The first order Markov model was used as model $\mathcal{M}$, assuming that x is the item purchased just before item y, and the negative log likelihood was used as error function $J(x, y; \mathcal{M})$. The weighted error to be minimized is as follows:

$$E\big(\{R(s_j|s_i)\}\big) = -\sum_{m=1}^{M} w(d_m) \log R(y_m|x_m), \tag{2.38}$$

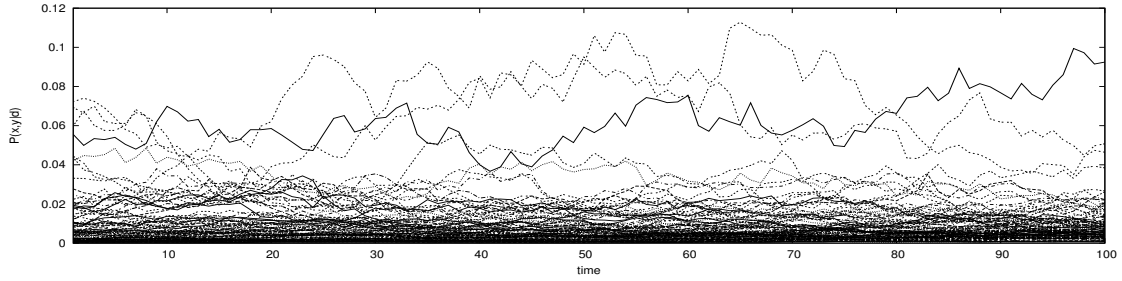
where $R(s_j|s_i)$ is the probability of purchasing item s_j after item s_i ($0 \leq R(s_j|s_i) \leq 1$, $\sum_{s \in \boldsymbol{S}} R(s|s_i) = 1$), and $\boldsymbol{X} = \boldsymbol{S}$, $\boldsymbol{Y} = \boldsymbol{S}$ in this case. Parameters $\{R(s_j|s_i)\}$



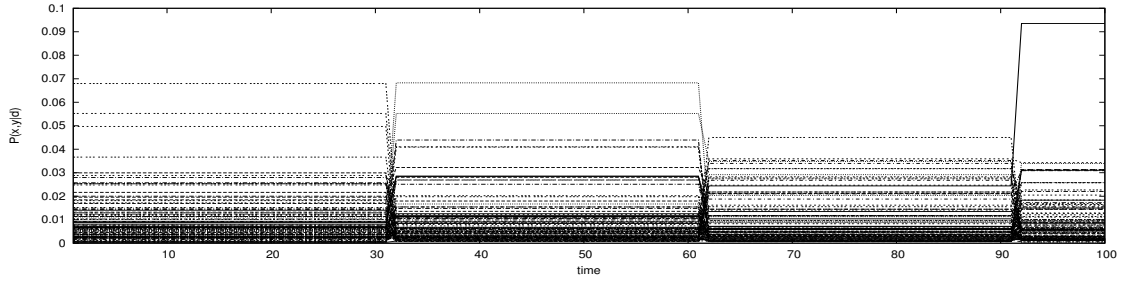
(a) Periodic



(b) Random



(c) Gradual



(d) Drastic

Figure 2.2: Examples of distributions of synthetic data.

that minimize the weighted error can be obtained by the following equation with Laplace smoothing parameter β:

$$\hat{R}(s_j|s_i) = \frac{\sum_{m=1}^{M} I\big((x_m = s_i) \wedge (y_m = s_j)\big) w(d_m) + \beta}{\sum_{m=1}^{M} I(x_m = s_i) w(d_m) + \beta V}, \tag{2.39}$$

where we used $\beta = 10^{-2}$ in the experiments.

The following six weighting methods were compared:

- **MCW** : Mixture Coefficient Weighting (proposed method).
- **Exp** : Exponential decay (2.27).
- **Input** : Ratio of input likelihoods (2.28).
- **Output** : Ratio of output likelihoods (2.31).
- **NoWeight** : No weight. $w(d) = 1$ for all d.
- **Present** : Only samples at the present time have weight. $w(D) = 1$ and $w(d \neq D) = 0$.

The hyper-parameter λ in Exp was estimated using a golden section search [46], which is an optimization method for one-dimensional parameters, with 10-fold cross-validation. Note that the range of the search was $[0, 10]$. The likelihood ratio was determined using the empirical distributions in Input and Output.

Six sets of learning samples were generated, in which the numbers of samples at each time were 64, 128, 256, 512, 1,024, and 2,048, respectively. The number of test samples was 8,192. We evaluated the predictive performance of each method from the perplexity of the held-out data at the present time D:

$$Perp(D) = \exp\left(-\frac{1}{M(D)} \sum_{m:d_m=D} \log R(y_m|x_m)\right). \tag{2.40}$$

A lower perplexity represents higher predictive performance. We generated 100 evaluation data sets with different $P(x, y|d)$ for Periodic, Random, Gradual, and Drastic data. Table 2.1 shows the average perplexities over the 100 evaluation sets. The value in parentheses is the average of the ratio of the perplexities between MCW and the other method, which is defined as follows:

$$ratio(D) = \frac{Perp_{\mathrm{MCW}}(D)}{Perp_{\mathrm{method}}(D)}, \tag{2.41}$$

where $Perp_{\text{method}}(D)$ represents the perplexity of the method. The value after $\pm$ in the table represents the standard deviation. The method is better than the proposed method if $ratio(D) > 1$.

The perplexities of the proposed method in the Periodic and Random data are the lowest for all numbers of learning samples. This result indicates that the proposed method performs well when some time points whose distributions are similar to the present time exist in the past, even if the locations of the similar time points are random. On the other hand, the perplexities of Exp in the Periodic and Random data are high because temporarily close points do not have similar distributions to the present time in these data. It is obvious that the exponential decay method, in which the weight changes with a constant ratio, is best suited to the gradually changing data, and the perplexity of Exp in the Gradual data is in fact the lowest. The proposed method exhibits almost the same performance as Exp, even for the Gradual data. With the Drastic data, the perplexities of the proposed method and Exp are comparable. This is reasonable because the distribution is the same when the time is close to the present time in both the Drastic and Gradual data, and this characteristic can be approximated by Exp, as well as by the proposed method. Since the joint distributions of both input and output change in all data, the performance of Input and Output that assume only input or output distribution changes is not high. The perplexity of NoWeight does not decrease with the number of learning samples. This is because NoWeight uses the samples at different time points, where their distributions are not the same as that at the present time. The perplexity of Present is high when the number of learning samples is small and decreases as the number of learning samples increases, and when the number of samples at D is sufficient, the performance approaches that of the proposed method. Since the proposed method does not explicitly model the change of distribution, it can handle various kinds of distribution changes, and this results in high predictive performance for almost all data sets.

Figure 2.3 and Figure 2.4 show the estimated weights of Periodic, Random, Gradual, and Drastic data when the numbers of learning samples at each time are 256 and 2,048, respectively. Note that the weights are normalized so that the maximum is one. In the Periodic data, the weights have periodically high weights every seven time points. In the Gradual data, the weights gradually decay as the time diverges from the present time. In the Drastic data, only the weights after $d = 90$ have high values. These results show that the estimated weights describe the characteristics of given data, although the estimated weights have some variance when there are fewer learning samples.

In addition, the proposed method was evaluated while changing the number of learning samples at D while fixing the number at other time points $d \neq D$.

Table 2.1: Perplexities for synthetic data with different numbers of learning samples at each time. The value in parentheses is the ratio of the perplexities between MCW and the other method with the standard deviation. The bold values represent those that were lower than others statistically significantly at the one-sigma level.

(a) Periodic

$M(d)$	**MCW**	**Exp**	**Input**	**Output**	**NoWeight**	**Present**
64	**7.72**	9.44 (.82±.06)	9.34 (.83±.06)	9.85 (.79±.08)	9.29 (.83±.05)	24.25 (.33±.06)
128	**7.10**	8.73 (.81±.04)	9.27 (.76±.04)	9.18 (.77±.04)	9.24 (.77±.04)	15.02 (.48±.05)
256	**6.87**	7.91 (.87±.02)	9.25 (.74±.04)	9.00 (.76±.03)	9.22 (.74±.04)	10.24 (.68±.05)
512	**6.77**	7.36 (.92±.01)	9.24 (.73±.04)	8.90 (.76±.03)	9.22 (.73±.04)	8.10 (.84±.03)
1024	**6.73**	7.03 (.96±.01)	9.24 (.73±.04)	8.86 (.76±.03)	9.21 (.73±.04)	7.23 (.93±.02)
2048	**6.72**	6.87 (.98±.01)	9.24 (.73±.04)	8.84 (.76±.03)	9.22 (.73±.04)	6.91 (.97±.01)

(b) Random

$M(d)$	**MCW**	**Exp**	**Input**	**Output**	**NoWeight**	**Present**
64	**7.65**	9.17 (.84±.07)	9.39 (.81±.05)	9.81 (.79±.08)	9.30 (.82±.05)	23.12 (.34±.05)
128	**7.07**	8.59 (.83±.06)	9.36 (.76±.05)	9.20 (.77±.04)	9.28 (.76±.04)	14.95 (.48±.05)
256	**6.83**	7.85 (.87±.04)	9.33 (.73±.05)	9.00 (.76±.04)	9.25 (.74±.04)	10.25 (.67±.05)
512	**6.73**	7.28 (.92±.02)	9.32 (.72±.04)	8.91 (.76±.04)	9.24 (.73±.04)	8.00 (.84±.03)
1024	**6.70**	6.96 (.96±.01)	9.31 (.72±.04)	8.87 (.75±.04)	9.23 (.73±.04)	7.17 (.93±.02)
2048	**6.68**	6.82 (.98±.01)	9.31 (.72±.04)	8.84 (.76±.04)	9.23 (.72±.04)	6.86 (.97±.01)

(c) Gradual

$M(d)$	**MCW**	**Exp**	**Input**	**Output**	**NoWeight**	**Present**
64	7.89	**7.15** (1.11±.05)	7.59 (1.04±.05)	8.17 (.98±.10)	7.59 (1.04±.05)	22.67 (.36±.06)
128	7.22	**6.99** (1.03±.02)	7.54 (.96±.03)	7.64 (.95±.03)	7.56 (.96±.03)	15.00 (.49±.06)
256	**6.95**	**6.87** (1.01±.01)	7.52 (.92±.02)	7.49 (.93±.02)	7.54 (.92±.02)	10.18 (.69±.05)
512	**6.83**	**6.80** (1.00±.00)	7.51 (.91±.02)	7.42 (.92±.02)	7.53 (.91±.02)	8.08 (.85±.03)
1024	**6.76**	**6.75** (1.00±.00)	7.50 (.90±.02)	7.39 (.92±.02)	7.53 (.90±.02)	7.16 (.95±.02)
2048	**6.73**	**6.73** (1.00±.00)	7.50 (.90±.02)	7.37 (.91±.02)	7.52 (.90±.02)	6.86 (.98±.01)

(d) Drastic

$M(d)$	**MCW**	**Exp**	**Input**	**Output**	**NoWeight**	**Present**
64	**7.74**	**7.39** (1.05±.05)	10.25 (.76±.06)	10.70 (.73±.08)	10.21 (.76±.06)	22.21 (.36±.06)
128	**7.09**	**7.01** (1.01±.02)	10.19 (.70±.05)	10.00 (.71±.05)	10.18 (.70±.05)	14.89 (.49±.06)
256	**6.80**	**6.83** (1.00±.01)	10.17 (.67±.05)	9.76 (.70±.04)	10.16 (.67±.05)	10.18 (.67±.05)
512	**6.69**	6.73 (.99±.00)	10.17 (.66±.05)	9.64 (.69±.04)	10.15 (.66±.04)	7.97 (.84±.04)
1024	**6.64**	**6.67** (1.00±.00)	10.16 (.65±.05)	9.59 (.69±.04)	10.15 (.65±.04)	7.10 (.94±.02)
2048	**6.62**	**6.64** (1.00±.00)	10.16 (.65±.05)	9.58 (.69±.04)	10.15 (.65±.04)	6.79 (.98±.01)

Table 2.2: Perplexities for synthetic data with different numbers of learning samples at D. The number of learning samples at $d \neq D$ is 2,048. The value in parentheses is the ratio of the perplexities between MCW and the other method with the standard deviation.

(a) Periodic

$M(d)$	**MCW**	**Exp**	**Input**	**Output**	**NoWeight**	**Present**
64	**6.81**	9.54 (.72±.06)	9.30 (.73±.04)	10.01 (.70±.09)	9.22 (.74±.04)	23.45 (.30±.06)
128	**6.77**	8.93 (.76±.04)	9.30 (.73±.04)	9.14 (.74±.04)	9.22 (.73±.03)	15.09 (.45±.05)
256	**6.72**	7.94 (.85±.03)	9.29 (.72±.04)	8.95 (.75±.04)	9.22 (.73±.03)	10.27 (.66±.05)
512	**6.70**	7.34 (.91±.01)	9.28 (.72±.04)	8.87 (.76±.03)	9.21 (.73±.03)	8.10 (.83±.03)
1024	**6.70**	7.00 (.96±.01)	9.27 (.72±.04)	8.82 (.76±.03)	9.19 (.73±.03)	7.21 (.93±.02)

(b) Random

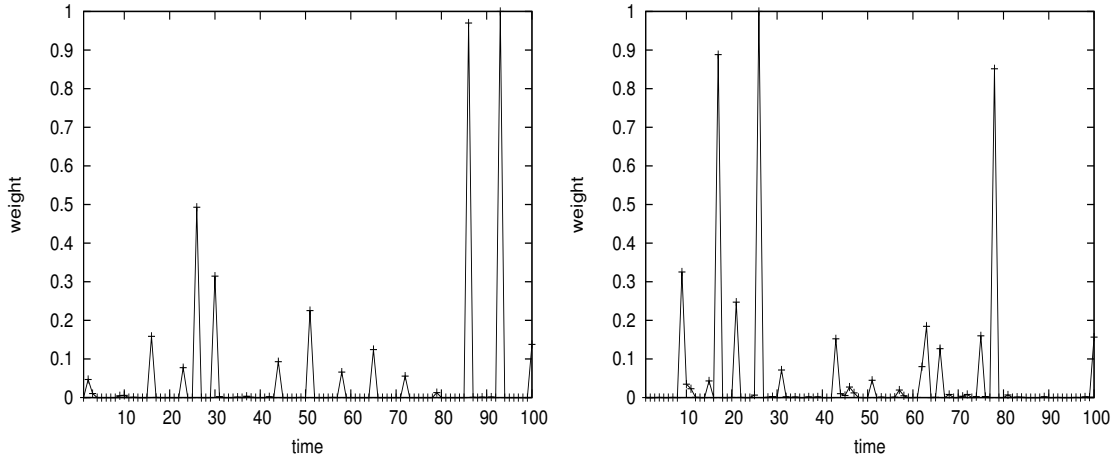
$M(d)$	**MCW**	**Exp**	**Input**	**Output**	**NoWeight**	**Present**
64	**6.79**	8.88 (.78±.11)	9.27 (.73±.05)	10.07 (.69±.09)	9.20 (.74±.04)	22.73 (.31±.05)
128	**6.73**	8.50 (.80±.09)	9.27 (.73±.04)	9.20 (.73±.05)	9.19 (.73±.04)	15.08 (.45±.06)
256	**6.69**	7.73 (.87±.05)	9.26 (.72±.04)	8.97 (.75±.04)	9.19 (.73±.04)	10.18 (.66±.05)
512	**6.67**	7.22 (.92±.03)	9.25 (.72±.04)	8.89 (.75±.04)	9.18 (.73±.04)	8.06 (.83±.04)
1024	**6.66**	6.93 (.96±.01)	9.24 (.72±.04)	8.82 (.76±.04)	9.16 (.73±.04)	7.15 (.93±.02)

(c) Gradual

$M(d)$	**MCW**	**Exp**	**Input**	**Output**	**NoWeight**	**Present**
64	**6.87**	**6.81** (1.01±.02)	7.44 (.92±.03)	8.27 (.85±.10)	7.45 (.92±.03)	23.56 (.30±.05)
128	**6.78**	**6.75** (1.01±.01)	7.43 (.91±.02)	7.56 (.90±.03)	7.45 (.91±.02)	14.94 (.46±.05)
256	**6.74**	**6.72** (1.00±.01)	7.43 (.91±.02)	7.42 (.91±.02)	7.45 (.91±.02)	10.09 (.67±.05)
512	**6.72**	**6.70** (1.00±.00)	7.43 (.90±.02)	7.36 (.91±.02)	7.45 (.90±.02)	8.00 (.84±.03)
1024	**6.70**	**6.70** (1.00±.00)	7.42 (.90±.02)	7.33 (.92±.02)	7.44 (.90±.02)	7.14 (.94±.02)

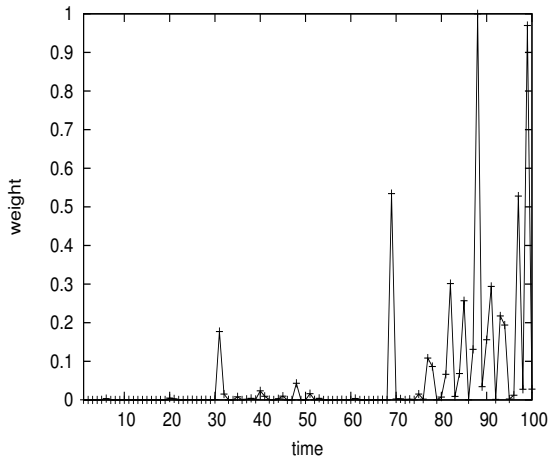
(d) Drastic

$M(d)$	**MCW**	**Exp**	**Input**	**Output**	**NoWeight**	**Present**
64	**6.66**	**6.65** (1.00±.01)	10.33 (.65±.06)	11.45 (.61±.10)	10.33 (.65±.05)	22.86 (.30±.06)
128	**6.63**	**6.62** (1.00±.01)	10.33 (.64±.06)	9.98 (.66±.05)	10.33 (.64±.05)	14.66 (.46±.05)
256	**6.60**	**6.61** (1.00±.00)	10.33 (.64±.06)	9.79 (.68±.05)	10.33 (.64±.05)	10.10 (.66±.05)
512	**6.58**	**6.60** (1.00±.00)	10.31 (.64±.06)	9.70 (.68±.05)	10.31 (.64±.05)	7.93 (.83±.03)
1024	**6.58**	**6.6**0 (1.00±.00)	10.28 (.64±.06)	9.62 (.68±.05)	10.27 (.64±.05)	7.08 (.93±.02)

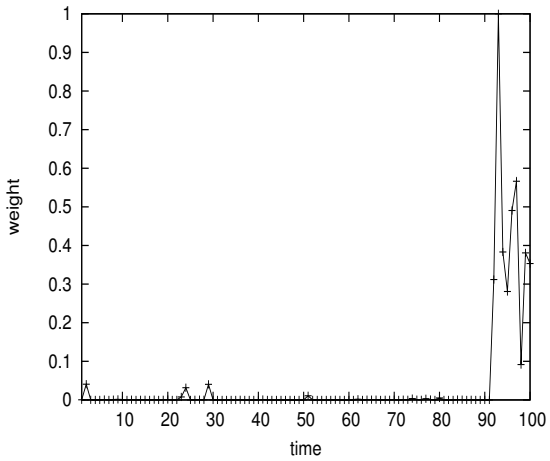


(a) Periodic

(b) Random



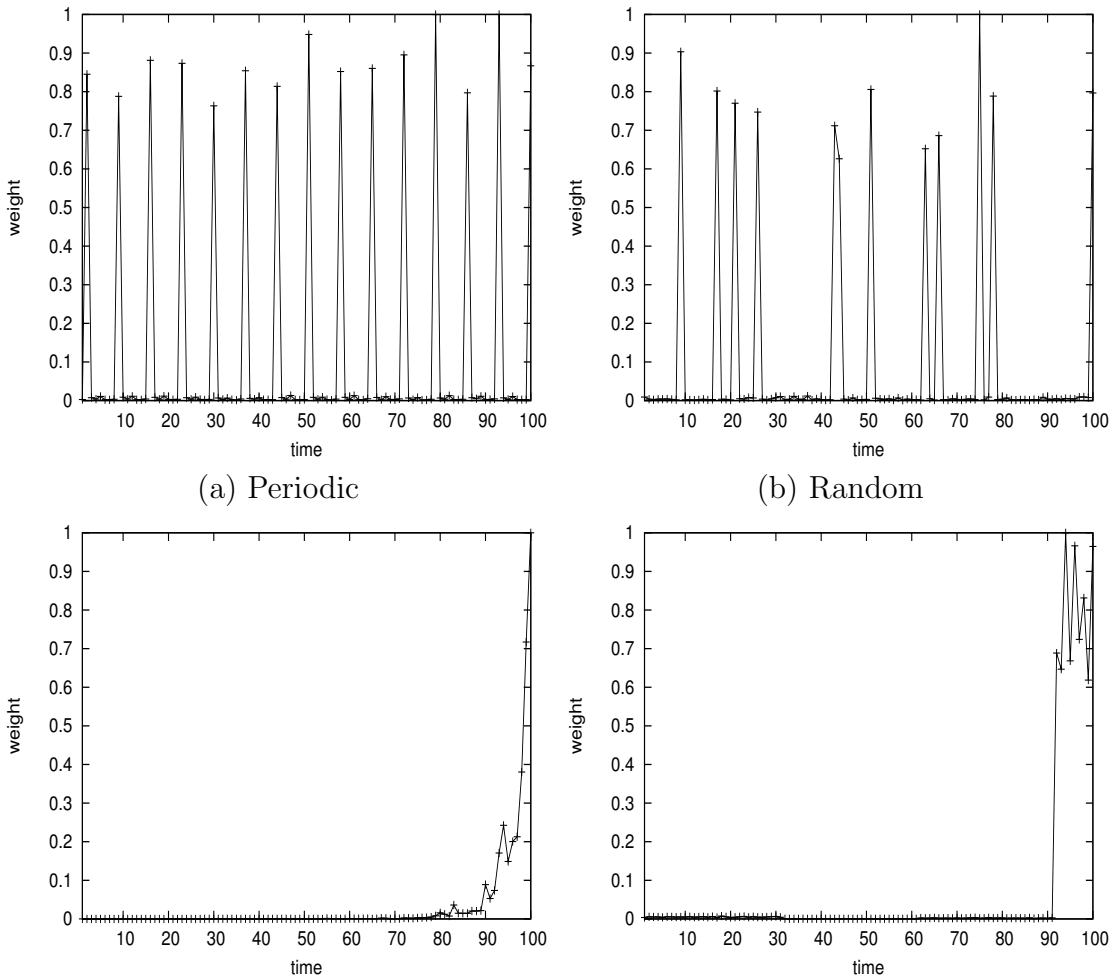
(c) Gradual



(d) Drastic

Figure 2.3: Estimated weights for synthetic data when the number of learning samples at each time is 256.

We generated sets of learning samples, in which the numbers of samples at D are 64, 128, 256, 512, and 1,024, respectively, and the number at other time points $d \neq D$ is 2,048. Table 2.2 shows the average perplexities over the 100 evaluation sets. The perplexities of the proposed method were low compared with those of other methods, especially as regards the Periodic and Random data. Compared with results for fewer learning samples at $d \neq D$ (Table 2.1), the perplexities of the proposed method in the case of more samples at $d \neq D$ were improved. One reason for this is that the number of samples that can be used to learn the model increases. Another reason is that the weights can be estimated robustly because



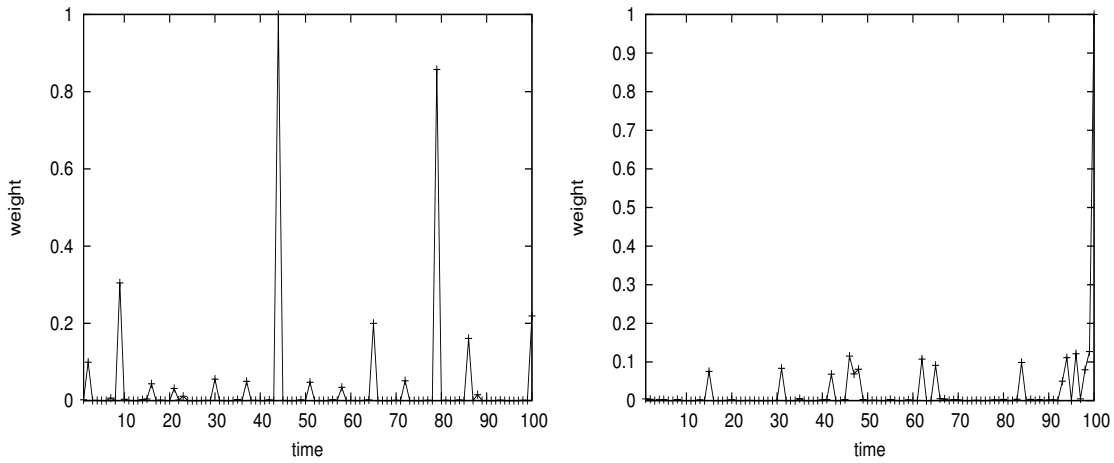
(a) Periodic (b) Random

(c) Gradual (d) Drastic

Figure 2.4: Estimated weights for synthetic data when the number of learning samples at each time is 2,048.

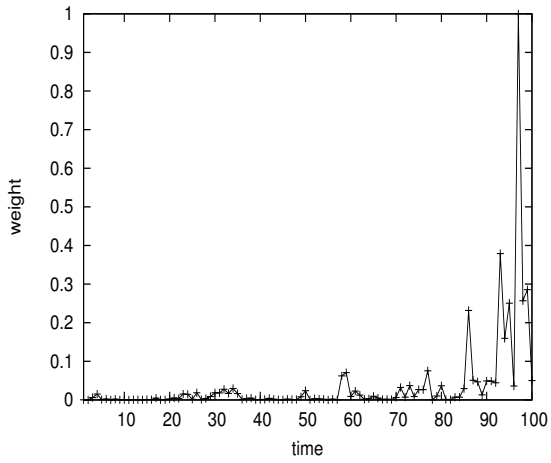
the variances of the estimated empirical distributions decrease when using more learning samples at $d \neq D$.

Figure 2.5 shows the weights estimated when the number of learning samples at D is 256 and the number at $d \neq D$ is 2,048. By using more samples at other time points, the estimated weights become stable in the sense that past time points that have similar distributions have similar weights. The past weights were smaller than that at the present time because the weight is divided by the number of samples $w(d) = P(d)/M(d)$.

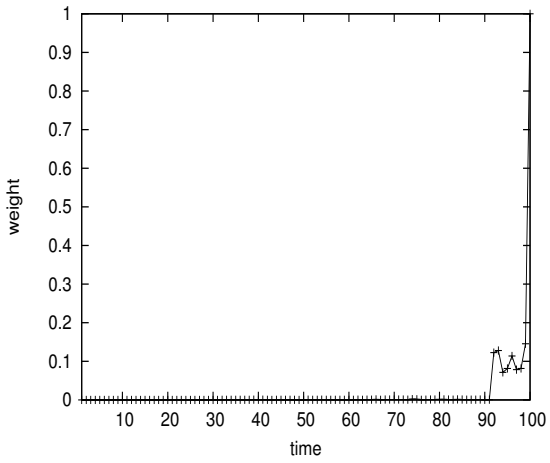


(a) Periodic

(b) Random



(c) Gradual



(d) Drastic

Figure 2.5: Estimated weights for synthetic data when the number of learning samples at the present time is 256 and the number at other time points is 2,048.

As described in Section 2.3, the computational complexity of a single EM iteration for the weight estimation is $O(M(D)D)$. That is, the computational time increases linearly with the number of samples at the present time and the number of time points. The computational time was measured experimentally on a PC having a 3.6-GHz Xeon CPU with 2 GB of memory. Figure 2.6 shows the average computational time over 100 Random data sets with the standard deviation changing the number of samples from 200 to 2,000 (a), and changing the number of samples at the present time from 200 to 2,000 while fixing the number at other times to 2,048 (b). The increase in the computational time is nearly linear with the number

Table 2.3: Average perplexities for the cartoon purchase log. The value in parentheses is the ratio of perplexities between MCW and the other method with the standard deviation.

MCW	Exp	Input	Output	NoWeight	Present
32.18	**33.73**(.95±.06)	43.59(.75±.16)	46.21(.70±.12)	46.94(.71±.17)	75.95(.46±.11)

of samples at the present time, and this result is consistent with the theoretical computational complexity. Note that this value includes the computational time for model learning. The computational cost for model learning is far smaller than the cost for weight estimation in this experiment. For example, approximately 0.05 seconds is required for learning the Markov model when the number of samples at each time is 2,048. The Markov model can be learned quickly by simply counting the numbers of samples. Since the proposed method simply weighs samples without modifying the model, the fast learning feature of the Markov model is retained. Figure 2.6 (c) shows the computational time when the number of time points changes, where the number of samples at each time is 512. The computational cost increases linearly with the number of the time points. Figure 2.6 (d) shows the computational time when the numbers of elements of input and output changes, where there are 512 samples at each time. The computational time does not increase despite the increase in the number of elements. This result implies that the proposed weighting method is efficient, even if the data is of very high dimension.

2.5.2 Real data

The proposed method was also evaluated using a real purchase log data set of a cartoon downloading service for cell phones from 1 April 2005 to 31 March 2006. The numbers of users, items, and transactions were 164,538, 175, and 1,018,741, respectively. Here, a cartoon with several volumes was regarded as one item. Items and users that appeared less than two times in the purchase histories were omitted. If an item was purchased more than once by a user, only the first purchase was considered. Assuming that the unit time was one day, we set the present time D to each day from the start date to the end data and created 365 data sets. Empirical distributions were used for model distributions, the first-order Markov model was used for model $\mathcal{M}$ and the negative log likelihood was used for error function $J(x, y; \mathcal{M})$, as in the synthetic data experiments. The perplexity was used for the evaluation measurements. For each D, the perplexity was evaluated by 10-fold cross-validation.

Table 2.3 shows the average perplexity over all days, $Perp = \frac{1}{\tilde{D}} \sum_{D=1}^{\tilde{D}} Perp(D)$,

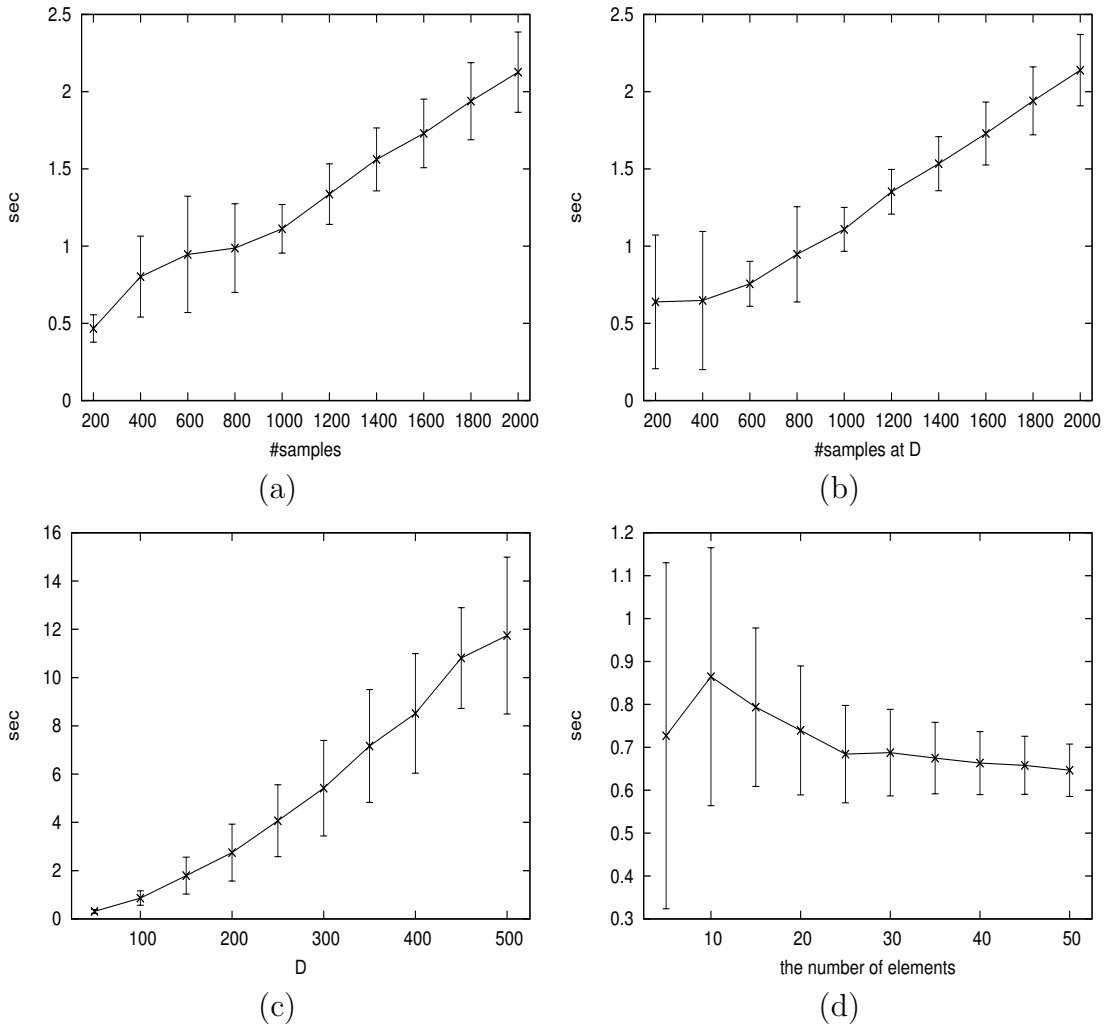


Figure 2.6: Computational time (second) of the proposed method changing (a) the number of learning samples at each time, (b) the number of learning samples at the present time, (c) the number of time points, and (d) the numbers of input and output elements.

where $\tilde{D}$ is the number of D, or $\tilde{D} = 365$ in this data. The value in parentheses is the average of the ratio of perplexities between MCW and the other methods (2.41) with the standard deviation. Figure 2.7 shows the daily ratio of perplexities. The average perplexity of the proposed method is the lowest, and the daily perplexities are the lowest for most days. This result indicates that the models that fit samples at the present time can be learned appropriately using the proposed method. Although the average perplexity of Exp is comparable to that of the proposed method, the ratios

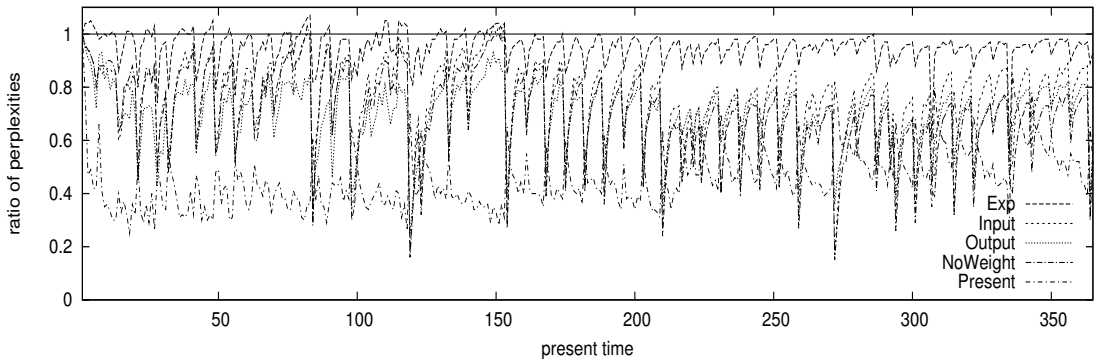


Figure 2.7: Daily ratio of perplexities between MCW and the other method for the cartoon purchase log.

of perplexities between the proposed method and Exp after $D = 154$ were not higher than one, which means that the daily perplexities of Exp after $D = 154$ were not better than those of the proposed method. The old-time weights in Exp are likely to be zero because the weights decay constantly. Therefore, Exp cannot effectively use all of the past data when D is large. On the other hand, the proposed method can use even data from a long time ago by tuning the weight at each time correctly, and this feature resulted in lower perplexities for a larger D. The perplexities of Input and Output, where samples at the same time can have different weights, are higher than those of the methods that can have a constant weight in samples at the same time, i.e., MCW and Exp. If the true distribution $P(x|d)$ or $P(y|d)$ is known at each time, and the assumption of the covariate shift is true, the perplexities of Input and Output should be lower. However, since it is difficult to estimate the true distributions correctly, and the assumption of the covariate shift might not hold, their perplexities were high. Since the proposed method uses weights that depend only on time, and the number of weights is smaller than those of Input and Output, the weight estimation is more robust. The perplexity of Present is the highest, which is reasonable because there are few training samples.

Figure 2.8 shows the daily share of items, which represents the percentage of transactions for each item each day. The shares vary from day to day, which means that the cartoon data are indeed dynamically changing data. The perplexities of the proposed method and Present tend to decrease on days when the share changes significantly, while that of NoWeight tends to increase on these days. As an example, in the period shown in Figure 2.9 and Figure 2.10, a new item was released on the 119th day, and the share changed greatly on that day. Most items that increased their shares were newly released items. Since data for newly released

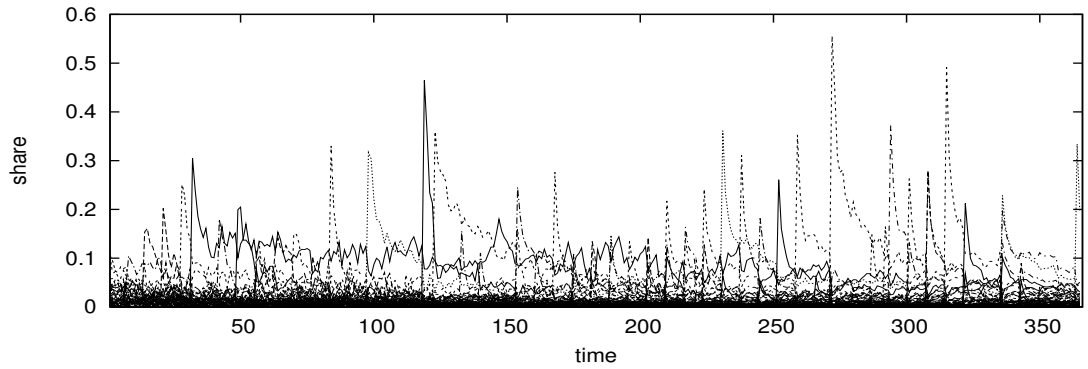


Figure 2.8: Daily share of items in the cartoon purchase log.

items are scarce, the performance of NoWeight that treats all samples equally is likely to be low. On the other hand, since Present emphasizes the present day, Present performs well when there is a new release. However, the perplexity of Present is usually high because it uses relatively fewer samples for training, as compared to the other methods. The proposed method can automatically adapt the weights using the daily distribution information so as to use samples only on the present day if an item is newly released. The proposed method can also use past samples if the share remains unchanged.

Figures 2.11 (a) and (b) show the weights obtained using the proposed method, where the present time is the 118th day and 119th day, respectively. Since a new item was released on the 119th day, and the share changed significantly on that day, weights excluding the present time are almost zero. On the other hand, some weights in the 118th data set have volume even on past days.

2.6 Summary

In this chapter, a theoretical framework was proposed for better model learning at the present time by effectively using past data, in which the samples are weighted depending on the time generated. The weights are determined as the mixture coefficients of a mixture of the empirical distributions through time that approximates the distribution at the present time. Using the proposed method, models can be learned robustly using past samples as well with adequate weights. Experiments using synthetic data sets and a real log data set of an online cartoon downloading service have demonstrated that the proposed method could fit the model at the present time.

Experiments have demonstrated the effectiveness of the proposed method using

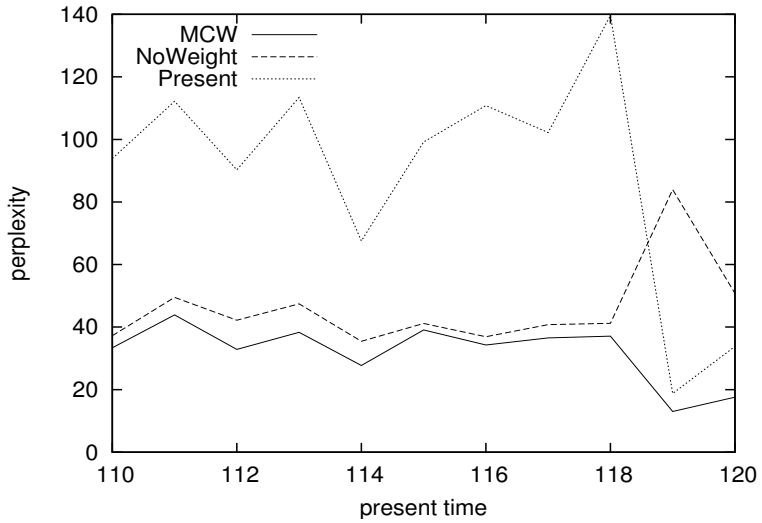


Figure 2.9: Daily perplexities for the cartoon purchase log from the 110th day to the 120th day.

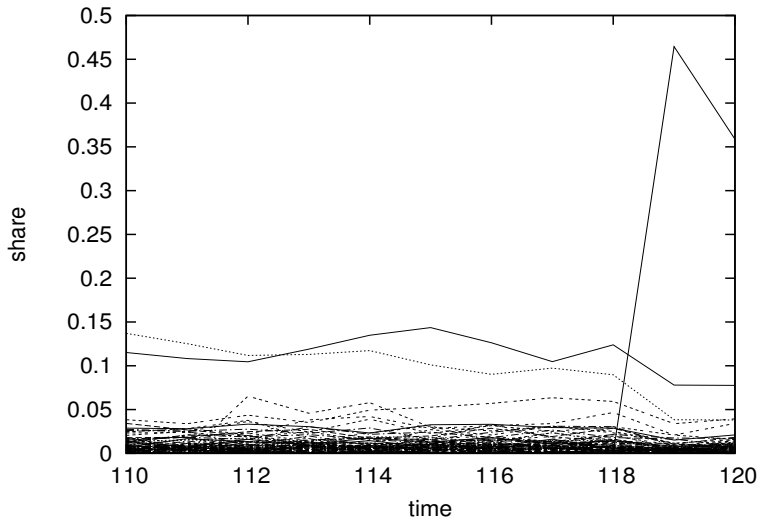
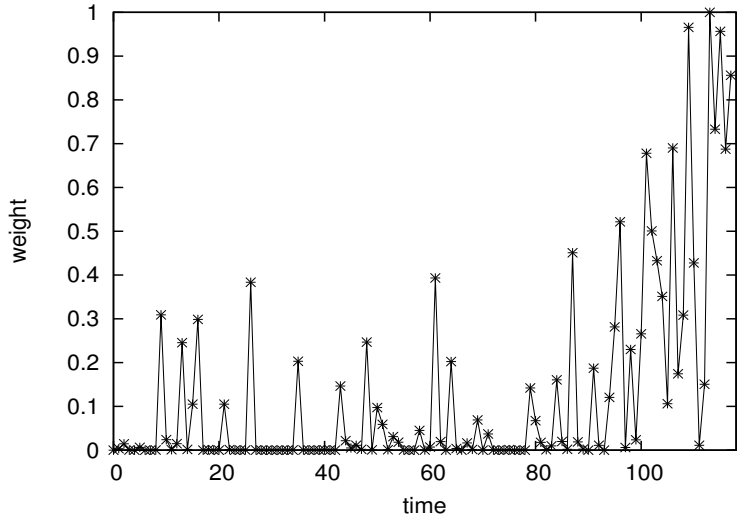
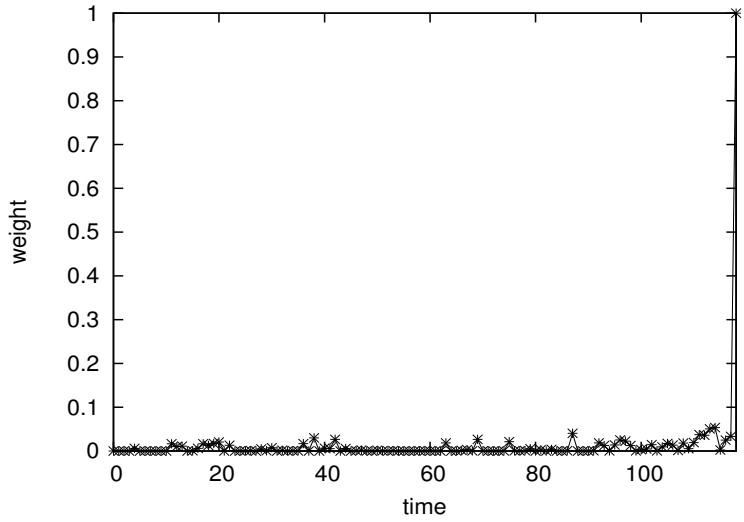


Figure 2.10: Daily share in the cartoon purchase log from the 110th to the 120th day.

purchase log data. The proposed method can also be applied to various kinds of dynamically changing data, such as news stories, scientific articles, and web surfing data, where a sufficient number of samples are generated at each time. Simple Markov models were used as the models in the experiments. Further verification of the proposed method by applying it to other types of data and other types of



(a) 118th day



(b) 119th day

Figure 2.11: Estimated weights for the cartoon purchase log on the 118th day (a), and on the 119th day.

models is necessary.

The proposed method was introduced as a method to learn models for dynamically changing data, assuming that d is time. By considering that d is another discrete variable, the proposed method can be used for other types of problems, such as domain adaptation [13, 12] and multi-task learning [62]. For example, d could be the country of the user, and we could fit the model to domestic users by

using the data of users in other countries with weights determined by the proposed method. Domain adaptation techniques are used in spam filtering [5] or natural language processing [12, 31], for example. In the future, the application area of the proposed method will be investigated extensively.

Chapter 3

Efficient choice model using temporal purchase order information

3.1 Introduction

In the previous chapter, a framework for learning models that can accurately predict present data was proposed. Efficiency is an important issue for recommender systems because the systems require frequent update to maintain high accuracy by handling a large number of purchase log data that are accumulated day by day. In this chapter, a computationally efficient probabilistic choice model that considers temporal purchase order information, which can be used for recommendations by suggesting an item that maximizes the choice probability, is proposed.

Most recommendation methods do not consider the order in which items are purchased [50, 53]. However, the recent purchase history can be more informative than the early purchase history for predicting the next purchase item. For example, when the first volume of a series of DVD movies is purchased, the next purchase might be the second volume. Moreover, since the interests of some users change, their early purchase histories would not be useful for predicting the next item. For computational efficiency, some methods consider only the previously purchased item [28, 33], and disregard a lot of purchased item information.

In some recommendation methods, rating or item information is used as the input. However, it is often difficult for stores to obtain such information. For example, stores need to ask their users to evaluate items for the purpose of collecting rating data sets. On the other hand, online stores that actually sell items usually already have purchase histories. In such cases, purchase histories constitute basic

data for online stores, and the focus herein is choice models that depend on purchase histories.

Markov models and maximum entropy models are used for choice modeling [32], as methods that take sequential information into account. Although we can estimate and update the parameters of Markov models efficiently, their predictive performance is limited. On the other hand, maximum entropy models incur greater computational cost for the estimation of their parameters, but their predictive performance is high. The proposed method achieves low computational cost and high predictive performance by combining multiple simple Markov models using the maximum entropy principle. The parameters of the proposed method can be updated quickly. Fast parameter update is important for online stores because a large number of purchase histories are accumulated at any given moment.

The remainder of this chapter is organized as follows. In Section 3.2, the Markov models and maximum entropy models that are basis of the proposed method are described. In Section 3.3, an efficient choice model that uses sequential information is presented. In Section 3.4, a brief description of related research is presented. In Section 3.5, the proposed method is evaluated using three sets of log data of online music, movie, and cartoon distribution services, and the low computational cost and high predictive performance are demonstrated. In the last section, this chapter is summarized and future research is described.

3.2 Conventional methods

3.2.1 Markov models

Markov models are widely used as probabilistic models that can employ sequential information. In an Lth Markov model, the next purchase item depends on the previous L purchased items as follows:

$$R(s_k|u_k) = R(s_k|s_{k-1}, \cdots, s_{k-L}), \tag{3.1}$$

where s_k is the kth purchase item and $u_k = (s_1, \cdots, s_{k-1})$ is the purchase history at the kth purchase. The probability of purchasing item s_k given purchase history u_k in the Lth Markov model can be written as the following equation using the maximum a posteriori (MAP) estimation:

$$\hat{R}(s_k|u_k) = \frac{\sum_{n=1}^{N}\sum_{v=L+1}^{K_n} I\big((s_k = s_{nv}) \wedge (s_{k-1} = s_{n,v-1}) \wedge \cdots \wedge (s_{k-L} = s_{n,v-L})\big) + \beta}{\sum_{n=1}^{N}\sum_{v=L+1}^{K_n} I\big((s_{k-1} = s_{n,v-1}) \wedge \cdots \wedge (s_{k-L} = s_{n,v-L})\big) + \beta V}, \tag{3.2}$$

where V is the number of items, and β is a hyper-parameter that can be estimated by a leave-one-out cross-validation. The denominator and numerator represent the number of purchase sequences $(s_k, \cdots, s_{k-L})$ and $(s_{k-1}, \cdots, s_{k-L})$ in the purchase histories of all users, respectively.

Since the parameter of the Markov model can be written as above in a closed form and the model can be calculated solely by a simple summation, its computational cost for estimation is low. Moreover, its parameter can be easily updated after new data are added to the training data set, as described hereinafter. First, we count and memorize the numbers of purchase sequences in the data that correspond to the denominator and numerator in (3.2). Then, when new data are added, we simply add the numbers of sequences in the new data to the counts in memory. In this process, we do not need to read the existing data again. However, the robust estimation of high-order Markov models is difficult because the number of parameters of the Lth Markov model is $O(V^{L+1})$ and the number becomes huge compared with the number of training samples for high-order Markov models.

We can reduce the number of parameters of a Markov model by assuming the conditional independence of the l-previous purchased item s_{k-l} and the l'-previous purchased item $s_{k-l'}$ given s_k as follows:

$$R(s_{k-1}, \cdots, s_{k-L}|s_k) = \prod_{l=1}^{L} R_l(s_{k-l}|s_k), \tag{3.3}$$

in which we still use information of L previous purchased items. Here, $R_l(s'|s)$ is referred to as an l-gapped Markov model, which represents a probability that the l-previous purchased item is s', given that the present purchased item is s. In accordance with the Bayes rule, the probability of purchasing item s_k given purchase history u_k becomes as follows with the L gapped Markov models assuming the conditional independence:

$$\begin{aligned} R(s_k|u_k) &= \frac{R(s_k)R(s_{k-1}, \cdots, s_{k-L}|s_k)}{\sum_{s\in\boldsymbol{S}} R(s)R(s_{k-1}, \cdots, s_{k-L}|s)} \\ &= \frac{R(s_k)\prod_{l=1}^{L} R_l(s_{k-l}|s_k)}{\sum_{s\in\boldsymbol{S}} R(s)\prod_{l=1}^{L} R_l(s_{k-l}|s)}, \end{aligned} \tag{3.4}$$

where $R(s)$ is the prior probability of purchasing item s. The number of parameters in this model is $O(LV^2)$, and it is far smaller than the number of the normal Markov model $O(V^{L+1})$ in the high-order case. The MAP estimation of the prior is as follows:

$$\hat{R}(s_k) = \frac{\sum_{n=1}^{N}\sum_{v=L+1}^{K_n} I(s_k = s_{nv}) + \beta}{\sum_{n=1}^{N} K_n - NL + \beta V}, \tag{3.5}$$

where the denominator is the number of purchases, and the numerator is the number of item s_k purchases. The MAP estimation of the l-gapped Markov model is as follows:

$$\hat{R}_l(s_{k-l}|s_k) = \frac{\sum_{n=1}^{N}\sum_{v=L+1}^{K_n} I\big((s_k = s_{nv}) \wedge (s_{k-l} = s_{n,v-l})\big) + \beta}{\sum_{n=1}^{N}\sum_{v=L+1}^{K_n} I(s_k = s_{nv}) + \beta V}, \tag{3.6}$$

where the denominator is the number of item s_k purchases, and the numerator is the number of a purchase pattern in which the l-previous item of s_k is item s_{k-l}.

3.2.2 Maximum entropy models

The maximum entropy model estimates a probability distribution that maximizes entropy under the constraints in the given data, and this model has been used in various fields of research such as collaborative filtering [33, 43, 64] and natural language processing [42, 49]. In the maximum entropy model, the probability of purchasing item s_k given purchase history u_k is as follows:

$$R(s_k|u_k) = \frac{1}{Z(u_k)} \exp\Big(\sum_{c\in\boldsymbol{C}} \alpha_c y_c(u_k, s_k)\Big), \tag{3.7}$$

where c is a feature index, $\boldsymbol{C}$ is a set of feature indices, α_c is an unknown parameter to be estimated, y_c is a feature of the purchase history, and $Z(u_k) = \sum_{s\in\boldsymbol{S}} \exp\Big(\sum_{c\in\boldsymbol{C}} \alpha_c y_c(u_k, s)\Big)$ is the normalization term. We use the following l-previous purchased item as the feature for considering sequential information:

$$y_{lij}(u_k, s_k) = \begin{cases} 1 & \text{if } s_i = s_{k-l} \text{ and } s_j = s_k, \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

The unknown parameters $\boldsymbol{\alpha} = \{\alpha_c\}_{c\in\boldsymbol{C}}$ can be estimated by maximizing the following log likelihood $L(\boldsymbol{\alpha})$ using optimization techniques such as quasi-Newton methods [37]:

$$\begin{aligned} L(\boldsymbol{\alpha}) &= \sum_{n=1}^{N}\sum_{k=1}^{K_n} \log R(s_{nk}|u_{nk}) \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K_n}\sum_{c\in\boldsymbol{C}} \alpha_c y_c(u_{nk}, s_{nk}) \\ &\quad - \sum_{n=1}^{N}\sum_{k=1}^{K_n} \log \sum_{s\in\boldsymbol{S}} \exp\Big(\sum_{c\in\boldsymbol{C}} \alpha_c y_c(u_{nk}, s)\Big), \end{aligned} \tag{3.9}$$

In maximum entropy models, we can obtain a global optimum solution. By using a Gaussian prior with a zero mean on unknown parameter α_c, overfitting can be reduced [9]. A Gaussian prior is used in the present experiments. See Appendix A.2 for details of the parameter estimation.

In some cases, discriminative models such as maximum entropy models have higher predictive performance than generative models such as Markov models [48], and it has been experimentally confirmed in the collaborative filtering problem [32]. However, an iterative optimization method such as a quasi-Newton method is required for the parameter estimation in the maximum entropy model, and this model incurs a high computational cost for its parameter estimation and updating as the number of parameters increases compared with Markov models.

3.3 Proposed method

A choice model that achieves high computational efficiency comparable to that of Markov models and high predictive performance comparable to that of maximum entropy models is proposed by integrating multiple gapped Markov models using the maximum entropy principle. First, we estimate the prior probability of purchasing item s_k, $R(s_k)$, using Equation (3.5), and the l-gapped Markov models, $R_l(s_{k-l}|s_k)$, $l = 1, \cdots, L$, using Equation (3.6). We use their log likelihoods as features of the maximum entropy model:

$$y_0(u_k, s_k) = \log \hat{R}(s_k), \tag{3.10}$$

$$y_l(u_k, s_k) = \log \hat{R}_l(s_{k-l}|s_k), \quad l = 1, \cdots, L. \tag{3.11}$$

This means that we maximize the entropy under the constraints that the expectations of their log likelihoods with the empirical distribution and with model $R(s_k|u_k)$ are the same as follows:

$$\sum_{n=1}^{N} \sum_{k=1}^{K_n} \log \hat{R}(s_{nk}) = \sum_{n=1}^{N} \sum_{k=1}^{K_n} \sum_{s \in \boldsymbol{S}} R(s|u_{nk}) \log \hat{R}(s), \tag{3.12}$$

$$\sum_{n=1}^{N} \sum_{k=l+1}^{K_n} \log \hat{R}_l(s_{n,k-l}|s_{nk}) = \sum_{n=1}^{N} \sum_{k=l+1}^{K_n} \sum_{s \in \boldsymbol{S}} R(s|u_{nk}) \log \hat{R}_l(s_{n,k-l}|s). \tag{3.13}$$

Then, we obtain the probability of item s_k given purchase history u_k:

$$R(s_k|u_k) \;=\; \frac{1}{Z(u_k)} \exp\Big(\alpha_0 \log \hat{R}(s_k) + \sum_{l=1}^{L} \alpha_l \log \hat{R}_l(s_{k-l}|s_k)\Big)$$

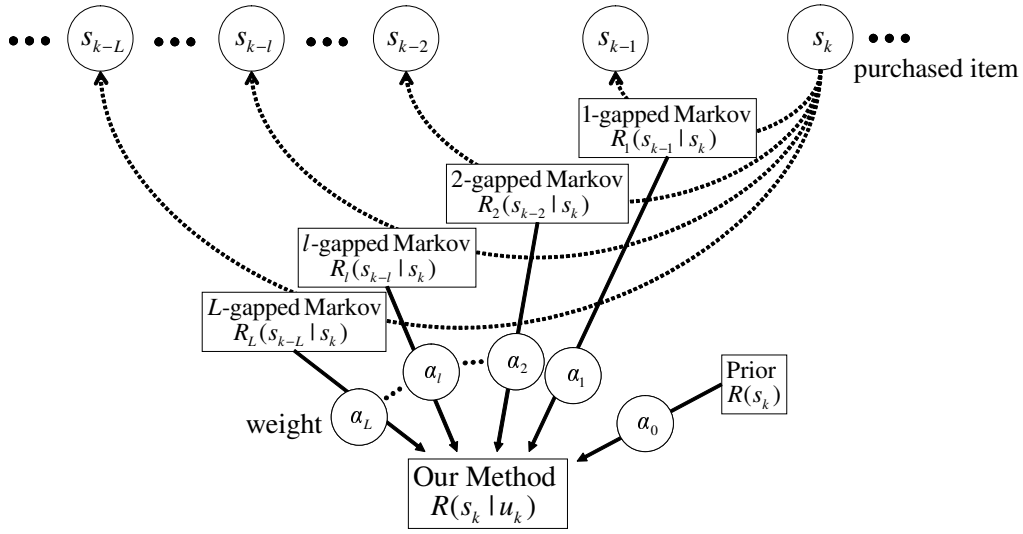


Figure 3.1: Hybrid model of multiple gapped Markov models with weights.

$$= \frac{1}{Z(u_k)} \hat{R}(s_k)^{\alpha_0} \prod_{l=1}^{L} \hat{R}_l(s_{k-l}|s_k)^{\alpha_l}, \tag{3.14}$$

where $\boldsymbol{\alpha} = \{\alpha_l\}_{l=0}^{L}$ are unknown parameters, α_0 represents the weight of the prior probability, α_l represents the weight of the l-gapped Markov model, and $Z(u_k) = \sum_{s \in \boldsymbol{S}} \exp\Big(\alpha_0 \log \hat{R}(s) + \sum_{l=1}^{L} \alpha_l \log \hat{R}_l(s_{k-l}|s)\Big)$ is the normalization term. If all of the weights are equal to 1, $\alpha_l = 1, l = 0, \cdots, L$, the proposed method coincides with Equation (3.4), which assumes the conditional independence of L gapped Markov models. Figure 3.1 shows a diagram of the proposed method.

We can obtain a global optimum solution for the unknown parameters $\boldsymbol{\alpha}$ by maximizing the log likelihood in the same way as in maximum entropy models. If we estimate $\boldsymbol{\alpha}$ using samples that are used in estimating the priors and gapped Markov models, the estimated $\boldsymbol{\alpha}$ is likely to overfit. To avoid overfitting, we estimate $\boldsymbol{\alpha}$ by C-fold cross-validation. First, we divide a set of user indices $\boldsymbol{N} = \{n\}_{n=1}^{N}$ into C subsets of user indices, $\{\boldsymbol{N}_c\}_{c=1}^{C}$, and estimate prior $\hat{R}(s_k; \boldsymbol{N}_{-c})$ and gapped Markov model $\hat{R}_l(s_{k-l}|s_k; \boldsymbol{N}_{-c})$ employing a user subset $\boldsymbol{N}_{-c} = \{\boldsymbol{N}_j\}_{j \neq c}$ for each c as follows:

$$\hat{R}(s_k; \boldsymbol{N}_{-c}) = \frac{\sum_{n \in \boldsymbol{N}_{-c}} \sum_{v=1}^{K_n} I(s_k = s_{nv}) + \beta}{\sum_{n \in \boldsymbol{N}_{-c}} K_n + \beta V}, \tag{3.15}$$

$$\hat{R}_l(s_{k-l}|s_k; \boldsymbol{N}_{-c}) = \frac{\sum_{n \in \boldsymbol{N}_{-c}} \sum_{v=l+1}^{K_n} I\big((s_k = s_{nv}) \wedge (s_{k-l} = s_{n,v-l})\big) + \beta}{\sum_{n \in \boldsymbol{N}_{-c}} K_n - |\boldsymbol{N}_c| l + \beta V}. \tag{3.16}$$

Then, by maximizing the summation of the log likelihood of $\boldsymbol{N}_c$ that have not been used for estimating the priors and gapped Markov models:

$$L(\boldsymbol{\alpha}) = \sum_{c=1}^{C} \sum_{n \in \boldsymbol{N}_c} \sum_{k=1}^{K_n} \log R(s_{nk}|u_{nk}; \boldsymbol{N}_{-c}), \tag{3.17}$$

we can obtain unknown parameters $\boldsymbol{\alpha}$, where

$$R(s_k|u_k; \boldsymbol{N}_{-c}) = \frac{1}{Z(u_k)} \hat{R}(s_k; \boldsymbol{N}_{-c})^{\alpha_0} \prod_{l=1}^{L} \hat{R}_l(s_{k-l}|s_k; \boldsymbol{N}_{-c})^{\alpha_l}. \tag{3.18}$$

In the experiments, we used a Gaussian prior with mean 0 for unknown parameters $\boldsymbol{\alpha}$.

The number of parameters in the proposed method is $LV(V-1)+V+L$. We can quickly estimate and update $LV(V-1)+V-1$ parameters that correspond to the prior probabilities and L gapped Markov models, as described above. The number of parameters that incur a high computational cost for estimation is only $L+1$, which is corresponds to the weights $\boldsymbol{\alpha}$. Therefore, the total computation time is much less than that needed for standard maximum entropy models with features considering sequential information that have $O(LV^2)$ costly parameters.

We can further reduce the computational cost by updating only the priors and gapped Markov models while fixing weights $\boldsymbol{\alpha}$. Although the priors and gapped Markov models are likely to change based on the new release of items and changes in trends, weights $\boldsymbol{\alpha}$ are not likely change rapidly due to the addition of new data.

3.4 Related research

The integration of generative models by using the maximum entropy principle has been proposed for a document classifier that has multiple components, such as title, author, and references [18, 48]. In the proposed method, generative models are integrated in order to take sequential information into account and to realize fast parameter estimation and updating.

For the robust estimation of high-order Markov models, deleted interpolation has been proposed in language modeling, which interpolates a Markov model using multiple lower order Markov models [29]. In deleted interpolation, a huge memory is needed for high-order Markov models. In language modeling, the regularity of word sequences is high because they are restricted by grammar, and high-order Markov models are needed. On the other hand, since the regularity of purchase sequences is lower than that of word sequences, multiple gapped Markov models are

Table 3.1: Start dates, end dates, numbers of users, items, and transactions of evaluation data sets.

	start	end	#users	#items	#transactions
Music4	2005/4/1	2005/4/30	247	132	1,508
Music5	2005/4/1	2005/5/31	1,120	348	7,588
Music6	2005/4/1	2005/6/30	2,104	561	15,216
Movie	2007/1/1	2007/1/1	3,085	1,569	25,363
Cartoon1	2005/4/1	2006/1/31	42,184	153	453,386
Cartoon2	2005/4/1	2006/2/28	53,830	161	599,196
Cartoon3	2005/4/1	2006/3/31	69,217	175	808,182

used instead of high-order Markov models. A mixture of gapped Markov models is proposed in [47, 54]. The mixture coefficients and gapped Markov models are estimated using the EM algorithm [14], which requires iterative procedures.

3.5 Experiments

3.5.1 Data sets

The proposed method was evaluated experimentally using three sets of purchase log data, related music, movies, and cartoons.

The music data were purchase history log data of an online music distribution service. Three sets of music data, which were log data from 1 April 2005 to 30 April 2005, 31 May 2005, and 30 June 2005 were constructed and referred to as Music4, Music5, and Music6, respectively. The movie data were purchase history log data of an online movie distribution service for 1 January 2007. The cartoon data were purchase history log data of a cartoon distribution service for cell phones. Three sets of cartoon data, which were log data from 1 April 2005 to 31 January 2006, 28 February 2006, and 31 March 2006, were constructed and are referred to as Cartoon1, Cartoon2, and Cartoon3, respectively. Here, a cartoon that had several volumes was regarded as one item. From all of the data sets, items that occurred fewer than ten times and users that occurred fewer than five times in the purchase histories of all users were omitted. If an item was purchased more than once by a user, purchases after the second purchase were omitted. Table 3.1 shows the number of users, items, and transactions for music, movie, and cartoon data sets. The last purchase item of each user was predicted, and the purchase histories excluding the last purchase items were used as training data. Here, samples that

were not present in the training data were omitted from the test data.

3.5.2 Compared methods

The following eight models were compared with the proposed method (OurMethod).

- Markov1: 1st-order Markov model. $L = 1$ in Equation (3.2).
- Markov2: 2nd-order Markov model. $L = 2$ in Equation (3.2).
- Markov3: 3rd-order Markov model. $L = 3$ in Equation (3.2).
- GapMarkov: Combination of gapped Markov models with the assumption of their conditional independence. Equation (3.4).
- MaxEntSeq: Maximum entropy model with features considering sequential information as in Equation (3.8).
- MaxEnt: Maximum entropy model with features not considering sequential information as follows:

$$y_{ij}(u_k, s_k) = \begin{cases} 1 & \text{if } s_i \in u_k \text{ and } j = s_k \\ 0 & \text{otherwise,} \end{cases} \tag{3.19}$$

  where $u_k = \{s_v\}_{v=1}^{v-1}$ is the set of purchased items, and the feature represents whether an item has been purchased.

- Cosine: Item-based collaborative filtering based on the cosine similarity [53]. The cosine similarity between item s_i and item s_j is defined as follows:

$$sim(s_i, s_j) = cos(\boldsymbol{r}_i, \boldsymbol{r}_j) = \frac{\boldsymbol{r}_i^T \boldsymbol{r}_j}{\| \boldsymbol{r}_i \| \| \boldsymbol{r}_j \|}, \tag{3.20}$$

  where $\boldsymbol{r}_i = (r_{i1}, \cdots, r_{iN})^T$ is a column vector that represents the users who purchased item s_i, in which

$$r_{in} = \begin{cases} 1 & \text{if user } u_n, \\ 0 & \text{otherwise,} \end{cases} \tag{3.21}$$

  and $\| \cdot \|$ is the Euclidean norm. The probability of purchasing item s is proportional to the summation of the cosine similarities of items in the purchase history u_k, as follows:

$$R(s_k|u_k) \propto \sum_{s \in u_k} sim(s_k, s), \tag{3.22}$$

which implies that items similar to purchased items are likely to be purchased. This method is mainly used for the prediction of unknown ratings. The number of parameters is $O(V^2)$. Although user-based collaborative filtering methods, which use similarities between users instead of items, have been also proposed [50], these methods require more computational time and memory in general.

- PLSA: Probabilistic latent semantic analysis [21, 23, 24], in which the probability that user u purchases item s is as follows:

$$R(s|u) = \sum_{z=1}^{Z} P(s|z)P(z|u), \tag{3.23}$$

where z is a latent class, Z is the number of latent classes, $P(s|z)$ is the probability that a user in class z purchases item s, and $P(z|u)$ is the probability that user u belongs to class z. The probabilities $P(s|z)$ and $P(z|u)$ can be estimated by maximizing the following likelihood:

$$L = \sum_{n=1}^{N} \sum_{k=1}^{K_n} \log R(s_{nk}, u_n), \tag{3.24}$$

using the EM algorithm, where

$$R(s_{nk}, u_n) = \sum_{z=1}^{Z} P(z)P(s_{nk}|z)P(u_n|z). \tag{3.25}$$

This method is used for information retrieval and natural language processing as well as collaborative filtering [22]. The number of parameters is $O(NZ + VZ)$.

- Mixture: Mixture of gapped Markov models [47, 54], in which the probability of purchasing item s_k is modeled by the convex combination of gapped Markov models, as follows:

$$R(s_k|u_k) = \sum_{l=1}^{L} P(l)R_l(s_k|s_{k-l}), \tag{3.26}$$

where $P(l) \geq 0$ is the mixture coefficient, $\sum_{l=1}^{L} P(l) = 1$, and $R_l(s_k|s_{k-l})$ is the probability of purchasing item s_k given the l previous purchased item s_{k-l}.

The mixture coefficients $P(l)$ and probabilities $R_l(s_k|s_{k-l})$ can be estimated by maximizing the following likelihood:

$$L = \sum_{n=1}^{N} \sum_{k=1}^{K_n} \log R(s_{nk}|u_{nk}), \tag{3.27}$$

using the EM algorithm. The number of parameters is $O(LV^2)$, which is the same as that in the proposed method.

Markov, GapMarkov, MaxEntSeq, and OurMethod take sequential information into account, whereas MaxEnt, Cosine and PLSA do not. MaxEnt uses the anteroposterior relationship information because it directly models the posterior probability $R(s_k|u_k)$ of item s_k given purchase history u_k. On the other hand, Cosine and PLSA do not use even the anteroposterior relationship information because the cosine similarity is symmetric $sim(s_i, s_j) = sim(s_j, s_i)$ and PLSA models the joint probability $P(s, u)$ of item s and user u.

The weights $\boldsymbol{\alpha}$ in OurMethod were estimated by 10-fold cross-validation ($C = 10$). The hyper-parameters β in Markov models, prior probabilities, and gapped Markov models are estimated by leave-one-out cross-validation. See Appendix A.3 for the estimation with leave-one-out cross-validation. In maximum entropy models, the following features, which represent an item's popularity, were also used:

$$y_i(u_k, s_k) = \begin{cases} 1 & \text{if } s_i = s_k, \\ 0 & \text{otherwise,} \end{cases} \tag{3.28}$$

and their priors were Gaussian with mean 0 and variance 1.

3.5.3 Results

Table 3.2 and Table 3.3 show the accuracies and top-3 accuracies, respectively. The predictive accuracy is calculated as follows:

$$Acc = \frac{1}{N} \sum_{n=1}^{N} I\left(s_{n,K_n} = \hat{s}'(u_{n,K_n})\right) \times 100, \tag{3.29}$$

where $\hat{s}'(u)$ is the predicted next purchase item as follows:

$$\hat{s}'(u) = \arg\max_{s \in \boldsymbol{S}_{-u}} R(s|u), \tag{3.30}$$

which is the highest purchase probability item. Here, $\boldsymbol{S}_{-u}$ represents the set of items other than items in purchase history u. The top-3 accuracy represents the rate at

Table 3.2: Accuracies of choice models.

	Music4	Music5	Music6	Movie	Cartoon1	Cartoon2	Cartoon3
Markov1	15.7	12.8	11.4	39.1	15.8	16.2	16.1
Markov2	10.6	6.4	7.1	31.1	16.2	18.3	17.1
Markov3	2.1	1.4	2.0	30.1	10.4	14.4	14.4
GapMarkov	**19.5**(2)	12.9(1)	10.9(2)	39.4(1)	17.7(3)	20.7(4)	18.3(5)
MaxEntSeq	**19.5**(1)	**14.7**(8)	12.5(6)	38.7(3)	19.1(5)	**21.8**(6)	**20.9**(7)
MaxEnt	15.7	10.4	9.1	27.7	17.3	19.1	18.8
Cosine	8.9	5.7	4.6	5.4	8.9	8.4	8.4
PLSA	12.3(5)	8.3(5)	6.9(15)	6.1(50)	11.3(5)	8.8(5)	9.6(30)
Mixture	16.1(3)	12.8(1)	11.5(2)	39.5(7)	16.4(2)	16.7(2)	16.1(1)
OurMethod	**19.5**(2)	14.0(6)	**12.9**(8)	**39.6**(3)	**19.5**(10)	21.7(10)	19.6(10)

Table 3.3: Top-3 accuracies of choice models.

	Music4	Music5	Music6	Movie	Cartoon1	Cartoon2	Cartoon3
Markov1	30.5	24.7	21.9	44.4	29.1	29.6	28.8
Markov2	15.3	12.0	11.4	32.7	28.3	31.0	30.2
Markov3	2.1	3.2	3.3	31.0	17.8	22.6	23.5
GapMarkov	32.2(2)	24.3(2)	22.2(1)	44.3(1)	31.7(4)	34.4(4)	32.8(6)
MaxEntSeq	**36.4**(3)	**28.0**(3)	25.0(3)	43.9(3)	**34.5**(7)	36.7(7)	**35.6**(8)
MaxEnt	25.4	22.6	18.4	35.3	34.1	35.0	33.4
Cosine	17.4	14.4	12.1	10.3	17.9	17.3	18.3
PLSA	26.7(10)	19.6(25)	14.7(15)	13.5(50)	22.9(40)	21.5(5)	23.8(30)
Mixture	31.4(3)	24.7(1)	23.2(6)	44.7(10)	30.1(2)	31.6(2)	30.1(2)
OurMethod	34.7(6)	27.6(7)	**25.6**(10)	**45.0**(6)	34.4(10)	**42.7**(4)	35.1(10)

which the purchased item is included in the three highest purchase probability items, as follows:

$$Acc_3 = \frac{1}{N}\sum_{n=1}^{N} I\big(s_{n,K_n} = \hat{s'}(u_{n,K_n}) \vee s_{n,K_n} = \hat{s'}_2(u_{n,K_n}) \vee s_{n,K_n} = \hat{s'}_3(u_{n,K_n})\big) \times 100, \tag{3.31}$$

where $\hat{s'}_2(u_{n,K_n})$ and $\hat{s'}_3(u_{n,K_n})$ represent the second and third highest purchase probability items, respectively. With GapMarkov, MaxEntSeq, Mixture, and OurMethod, we set the maximum gap as $L = 1, 2, \cdots, 10$, and the highest accuracy is shown. The value in parenthesis is the maximum gap L. With PLSA, we set the number of latent classes as $Z = 5, 10, \cdots, 50$, and the highest accuracy is shown.

The value in parenthesis for PLSA is the number of latent classes Z. The accuracies of OurMethod and MaxEntSeq are comparably high. The accuracy of GapMarkov is lower than that of OurMethod, which indicates that the assumption of conditional independence is not appropriate. In all of the data sets, the accuracy of Markov3 was lower than that of Markov1 and Markov2 because the number of parameters becomes huge in higher-order Markov models, and it is difficult to estimate them robustly. The accuracy of MaxEntSeq is higher than that of MaxEnt for all data sets, which implies that sequential information is important for the prediction of the next purchase items. The accuracies of Cosine and PLSA are low because they do not consider sequential information in purchase histories. This result indicates that they are inadequate for the prediction of the next purchase items, even though they are widely used for unknown rating prediction. The accuracies of Mixture are largely the same as those of Markov1. The estimated mixture coefficient of the 1-gapped Markov model is almost one, whereas those of the other gapped Markov models are almost zero for each type of data. This result indicates that the 1-gapped Markov model is likely to be dominant when gapped Markov models are combined to form a mixture model for purchase history modeling, and the mixture model would not use l-gapped Markov models ($l \geq 2$) effectively. On the other hand, the proposed method combines gapped Markov models to form a product model [19].

Figure 3.2 shows the weights $\boldsymbol{\alpha}$ in OurMethod with maximum gap $L = 10$. The shorter the gap l is, the higher the weight α_l becomes. This result implies that recently purchased items are more informative for the prediction of the next purchase item, which is an intuitive result. Note that the weights do not change greatly even if the end dates are different.

Figure 3.3 shows the relationship between the accuracy and the maximum gap L in OurMethod, MaxEntSeq, and GapMarkov. The accuracy of OurMethod did not decrease. On the other hand, the accuracies of MaxEntSeq and GapMarkov decreased as L increased because overfitting occurred in high-order MaxEntSeq, and the assumption of the conditional independence of gapped Markov models becomes inadequate with increases in the maximum gap.

The computational cost of the proposed method can be reduced if we update only the priors and gapped Markov models while fixing weights $\boldsymbol{\alpha}$, as described above. The effect on the accuracy of fixing weights $\boldsymbol{\alpha}$ was evaluated. OurMethod2 in Figure 3.4 shows the accuracy of the proposed method, in which the weights $\boldsymbol{\alpha}$ were estimated using the log data until d days before the end date, and the priors and gapped Markov models were estimated using all log data. Here, hyper-parameters β in the priors and gapped Markov models were estimated using the log data d days before the end date. OurMethod and MaxEntSeq in Figure 3.4 were

Table 3.4: Computational time (second) of choice models

	Music4	Music5	Music6	Movie	Cartoon1	Cartoon2	Cartoon3
Markov1	0.03	0.12	0.30	2.00	1.61	2.13	2.96
Markov2	0.06	0.69	2.15	11.04	4.50	5.66	7.58
Markov3	0.05	0.76	2.65	9.35	20.45	27.42	40.05
GapMarkov	0.04	0.23	0.67	5.16	2.82	3.83	5.23
MaxEntSeq	12.77	456.56	1969.55	9316.09	132001.62	216770.25	347511.57
MaxEnt	6.28	217.54	916.47	8267.11	16046.90	21597.25	42994.86
Cosine	0.01	0.06	0.18	0.97	7.97	11.36	16.08
PLSA	1.07	10.31	29.05	117.69	239.01	356.28	478.59
Mixture	0.17	1.52	3.87	14.53	753.77	1242.36	1553.69
OurMethod	3.66	87.21	345.90	2968.72	5267.85	7931.55	10444.11
OurMethod2	0.08	0.45	1.12	7.49	4.39	5.59	7.50

estimated using the log data d days before. The accuracy of OurMethod2 does not decrease, and is comparable to those of OurMethod and MaxEntSeq that were estimated using all of the log data. On the other hand, there is a reduction in the accuracies of models that were estimated using the log data until d days before the end date. This result indicates that high predictive performance can be achieved by updating only the priors and gapped Markov models having parameters that be updated easily.

The computational time was measured experimentally on a PC having a 3.6-GHz Xeon CPU with 2 GB of memory. Table 3.4 shows the results. OurMethod2 represents a method that estimates only the priors and gapped Markov models. With GapMarkov, MaxEntSeq, Mixture, OurMethod, and OurMethod2, the maximum gap was set as $L = 10$. With PLSA, the number of latent classes was set as $Z = 10$. Markov, GapMarkov, Cosine, and OurMethod2 were fast and finished their parameter estimation in a few minutes with the largest data set Cartoon3. Although OurMethod, which also estimates the weights, needs more time than these methods, it was approximately 30 times faster than the comparably high performance method MaxEntSeq with the cartoon data sets.

3.6 Summary

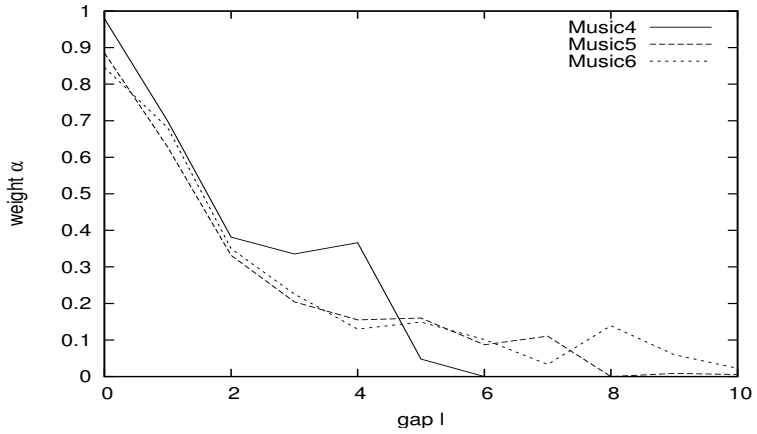
In this chapter, an efficient choice model was proposed that uses sequential information, in which multiple simple Markov models are integrated by the maximum entropy principle. High predictive performance and low computational cost were

demonstrated through experiments using real data sets.

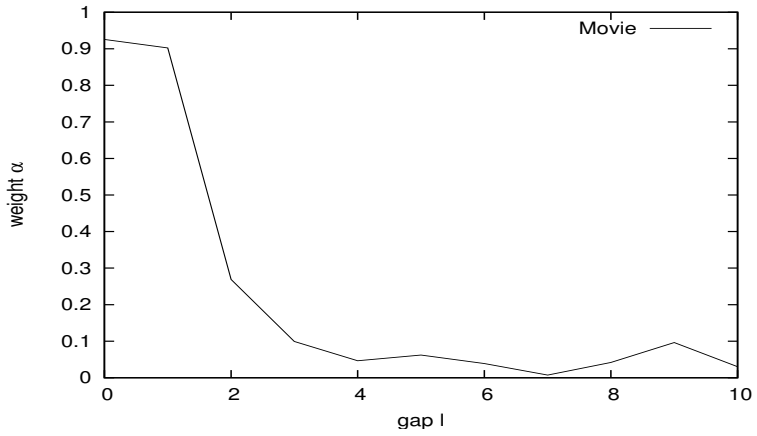
The proposed method may be extended for various applications. Even though the purchase history was used as the input, content information and user attributes can also be used to improve the proposed model. In the framework of the maximum entropy principle, this information can easily be integrated as integrated multiple Markov models. The weights of gap Markov models were estimated individually. The weights can be modeled by the exponential distribution as follows:

$$\alpha_l = \lambda \exp(-\lambda l). \tag{3.32}$$

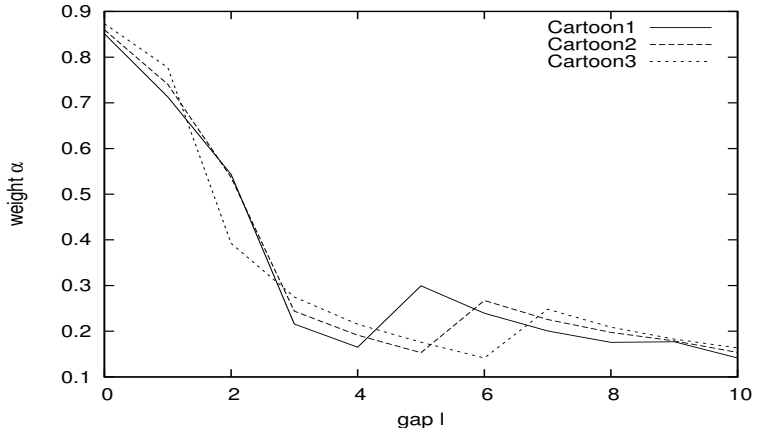
Estimating weights individually has several advantages. For example, weights can be estimated from data without prior knowledge, and the global optimum of the estimation is guaranteed. On the other hand, modeling weights has also an advantage in that the number of parameters can be reduced. Thus, further research into the modeling of weights is needed. In addition, the computational time needed to update parameters should be evaluated experimentally, and the number of transactions per second that the proposed method can handle should be demonstrated.



(a) Music

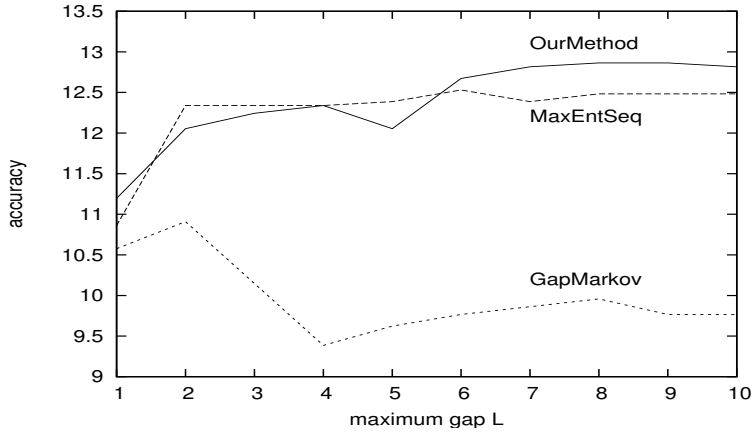


(b) Movie

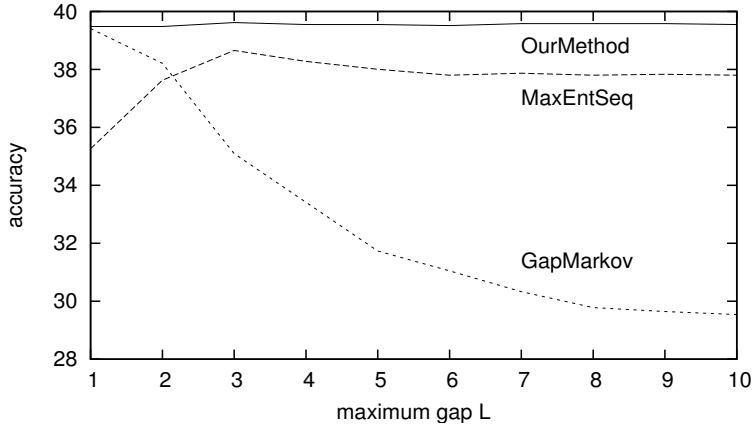


(c) Cartoon

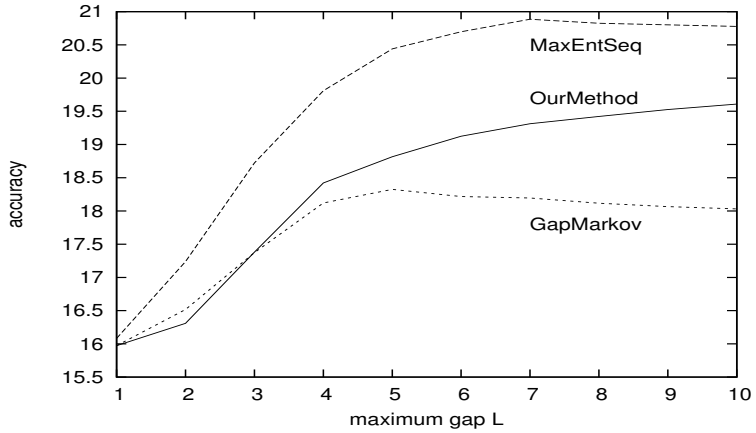
Figure 3.2: Estimated weights of gap Markov models when the maximum gap is ten.



(a) Music6

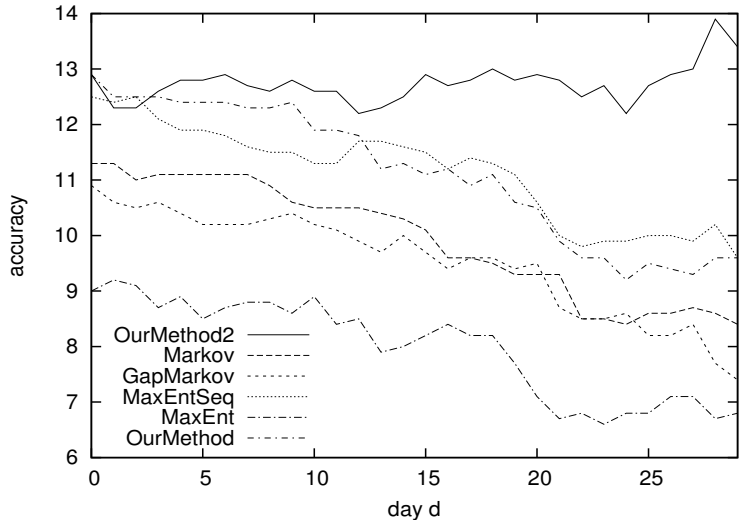


(b) Movie

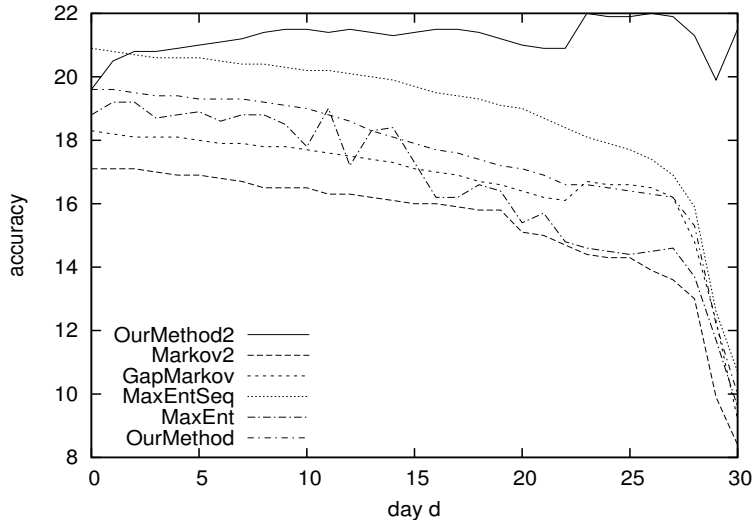


(c) Cartoon3

Figure 3.3: Accuracies of choice models with different maximum gaps.



(a) Music6



(b) Cartoon3

Figure 3.4: Accuracies of choice models estimated using data before the end date.

Chapter 4

Recommendation method for improving customer lifetime values

4.1 Introduction

Customer lifetime value (LTV) is defined as the net present value of profit that a customer generates over his/her entire purchase history, and it is used to evaluate the relationship between the store and each customer in marketing. Improving LTV, or developing the relationship with each existing customer to a higher level, is important for stores to increase long-term profits because acquiring new customers is often expensive.

The calculation of LTV depends on service types, which can be categorized as either measured or subscription services. With a measured service, users pay for individual purchased items. Therefore, LTV is roughly proportional to the purchase count of the user. With a subscription service, on the other hand, users pay for the periodic (e.g. monthly or yearly) use of magazines, music, movies, software, cell-phone services, etc. LTV in subscription services is therefore proportional to the subscription period of the user.

Recommendation is one way for online stores to influence user behavior. For example, the probability that the recommended item is purchased increases if the user did not know about the item before the recommendation or if the user is newly attracted to the item by the recommendation. In this chapter, a novel recommendation method is proposed that influences users so as to increase LTV for improving profits. Conventional recommendation methods recommend items that best coincide with a user's interests in order to maximize the purchase probability

[33, 40, 43, 45, 50], as described in previous chapters. Although these methods can increase short-term sales, they do not necessarily maximize long-term profits. For example, if an online store recommends an electronic product that has a lot of peripheral devices, the user is likely to revisit the store to purchase peripheral devices in the future. A recommendation of a DVD that is the first of a series can lead to the purchase of other DVDs in the series.

In the proposed method, LTV is modeled in a probabilistic form. Since the calculation of LTV differs between measured and subscription services, as described above, different probabilistic models are needed for each service, and LTV is modeled as purchase frequency models and subscription period models, respectively. The proposed method finds frequent purchase patterns among high LTV users using a probabilistic model for LTV and recommends items for a new user that simulate the found patterns. Since the possibility of purchasing the recommended item depends on the user's interests, the user's interests are taken into consideration using a probabilistic choice model in order to generate effective recommendations. To find the patterns, a probabilistic model in survival analysis [8, 10, 26, 34] is used. Maximum entropy models are used to estimate user's interests. Then, the found patterns are combined with the estimated user's interests in a probabilistically principled framework. Since a higher LTV is the result of higher user satisfaction, the proposed method benefits both users and online stores. Therefore, the proposed method can be seen as a tool for customer relationship management (CRM) [3]. CRM is important in terms of improving relationships between online stores and their users.

The remainder of this chapter is organized as follows. In the next section, a brief review of related research is presented. In Section 4.3, a recommendation method for improving LTV for measured services is proposed. In Section 4.4, the validity of the proposed method is demonstrated using the log data of an online music store. In Section 4.5, the proposed method is modified for subscription services. In Section 4.6, this method is applied to the log data of an online cartoon distribution service. Finally, this chapter is summarized and a discussion of future research is presented in Section 4.7.

4.2 Related research

A number of recommendation methods, such as collaborative filtering [43, 50], content filtering [40], and their hybrids [33, 45], have been proposed. These approaches recommend items that best coincide with a user's interests to maximize the purchase probability. Unlike these methods, the goal here is to improve LTV.

The use of LTV estimation to identify loyal customers has been studied by various researchers using survival analysis and data mining techniques [2, 17, 30, 39, 41, 59]. However, these techniques are not used for recommendation. Piatetsky-Shapiro and Masand [44] and Rosset et al. [52] proposed models for estimating the effects of marketing activity on LTV. A recommendation can be considered as a marketing activity. By focusing on the recommendation of items, the proposed method becomes an automatic recommendation framework that can learn from log data. A recommendation method for improving long-term profits using Markov decision process (MDP) has been proposed in [56]. Since the proposed method explicitly models purchase frequencies or subscription, as compared with the MDP-based method, models can be estimated robustly using log data.

4.3 Proposed method

4.3.1 Recommendation for improving customer lifetime values

The proposed method recommends item $\hat{s}$ that maximizes $P\big(l|u, r(s)\big)$, which is the probability of improving LTV of user u when item s is recommended, as follows:

$$\hat{s} = \arg\max_{s \in \boldsymbol{S}} P\big(l|u, r(s)\big), \tag{4.1}$$

where l represents an event in which the LTV is improved, $r(s)$ represents an event in which item s is recommended, and $P\big(l|u, r(s)\big) + P\big(\bar{l}|u, r(s)\big) = 1$, where $P\big(\bar{l}|u, r(s)\big)$ is the probability of not improving the LTV of user u when item s is recommended. Although the recommendation of one item is assumed above, we can also recommend m items with the highest $P\big(l|u, r(s)\big)$ values. In real applications, candidates for recommendation would be a subset of item set $\boldsymbol{S}$, such as a set of items not yet purchased by the user.

In general, $P\big(l|u, r(s)\big)$ cannot be directly estimated because it is not possible to observe whether the LTV is improved given a recommendation. Therefore, we decompose $P\big(l|u, r(s)\big)$ into two components with a number of assumptions so that $P\big(l|u, r(s)\big)$ can be estimated from data that can be easily obtained by online stores. Let s' be the item purchased by user u when item s is recommended. If the recommendation does not influence the user's purchase behavior, it is natural to think that the recommendation does not influence the LTV either. Therefore, we assume that improving LTV l and recommendation $r(s)$ are independent conditioned on purchased item s' and user u. Based on this assumption, $P\big(l|u, r(s)\big)$ can be decomposed into two components: LTV model $Q(l|u, s')$ and choice model

$R\big(s'|u, r(s)\big)$, as follows:

$$\begin{aligned} P\big(l|u, r(s)\big) &= \sum_{s' \in \boldsymbol{S}} P\big(l, s'|u, r(s)\big) \\ &= \sum_{s' \in \boldsymbol{S}} Q\big(l|u, s', r(s)\big) R\big(s'|u, r(s)\big) \\ &= \sum_{s' \in \boldsymbol{S}} Q(l|u, s') R\big(s'|u, r(s)\big). \end{aligned} \tag{4.2}$$

where the LTV model $Q(l|u, s')$ is the probability of the LTV being improved when user u purchases item s', the choice model $R\big(s'|u, r(s)\big)$ is the probability of purchasing item s' when item s is recommended to user u, and the conditional independence assumption $Q(l|u, s', r(s)) = Q(l|u, s')$ was used in the third equality. The conditional dependence assumption means that situations, in which the purchase of an item lead to further purchases, such as the purchase of an electronic product with many peripheral devices, are considered. The probability $Q(l|u, s')$ can be obtained from purchase frequency models, as described in Section 4.3.3, where purchase frequency models can be estimated from the log data that online stores have usually collected using survival analysis techniques, as described in the next section. The choice model $R\big(s'|u, r(s)\big)$ is estimated with maximum entropy models assuming that the item choice of a user depends on the user's purchase history and the recommended item, as described in Section 4.3.4. Although, in a strict sense, a variable that represents the purchase history should be used instead of u in $R\big(s'|u, r(s)\big)$, u was used in order to keep the equations simple. The framework of the proposed method is summarized in Figure 4.1.

4.3.2 Purchase frequency models

Assuming the profit generated by an item is constant for all items, the LTV is proportional to the purchase frequency in measured services. We derive the probability of improving LTV given the purchased item $Q(l|u, s')$ using purchase frequency models as described in the next section. In this section, purchase frequency models and their estimation procedures are presented.

Let e_{nk} be the status of the kth purchase of user u_n, which represents whether it is the last purchase, as follows:

$$e_{nk} = \begin{cases} 0 & \text{if the } k\text{th purchase of user } u_n \text{ is the last purchase,} \\ 1 & \text{otherwise.} \end{cases} \tag{4.3}$$

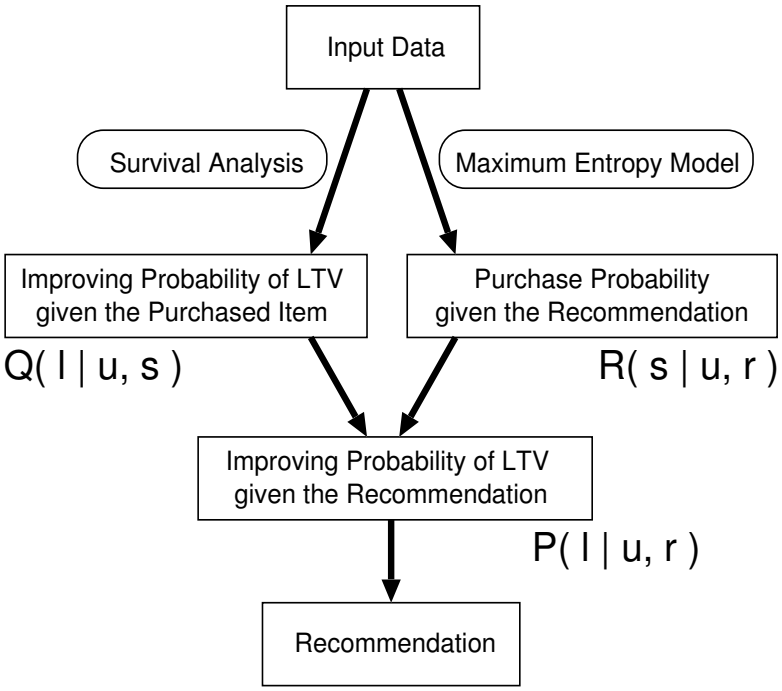


Figure 4.1: Framework of LTV improving recommendation.

Let t_{nk} be the interpurchase time of the kth purchase of user u_n, as follows:

$$t_{nk} = \begin{cases} d_{end} - d_{nk} & \text{if } e_{nk} = 0, \\ d_{n,k+1} - d_{nk} & \text{if } e_{nk} = 1, \end{cases} \tag{4.4}$$

where d_{nk} is the time of the kth purchase of user u_n, and d_{end} is the last time that the given log data were modified. We assume that the interpurchase time t is a discrete variable. The status e_{nk} and the interpurchase time t_{nk} are obtained from the purchase log. The purchase log lists the user, the item, and the time of each purchase. See Table 4.1 for an example purchase log. Figure 4.2 shows the relationships among the interpurchase time t, the purchase time d, the last modification time d_{end}, and the status e. As input data for modeling purchase frequencies, we use a set of users, purchase histories, interpurchase times, and statuses, as in shown in Table 4.2.

The purchase frequency, or interpurchase time, is modeled using frailty models [8, 26], which are used in survival analysis for modeling repeated events. The purchases are repeated events in the sense that a user purchase items repeatedly, as shown in Figure 4.2. The purchase frequencies can differ among users even if their preferences are the same, some heavy users purchase many items and some users

Table 4.1: Example purchase log.

user	item	purchase time
u_1	s_3	2004/8/16 12:06:28
u_1	s_1	2004/8/16 13:01:21
u_2	s_2	2004/8/16 18:51:43
u_1	s_6	2004/8/16 21:35:06
u_3	s_2	2004/8/17 16:42:11
⋮	⋮	⋮
u_N	s_{10}	2005/10/28 23:15:14

Table 4.2: Example input data of the recommendation method for measured services.

user	interpurchase time	status	purchase history
u_1	3293	1	s_3
u_1	21022	1	$s_3,\ s_1$
u_1	3253802	0	$s_3,\ s_1,\ s_6$
u_2	43261	1	s_2
u_2	243039	0	$s_2,\ s_8$
⋮	⋮	⋮	

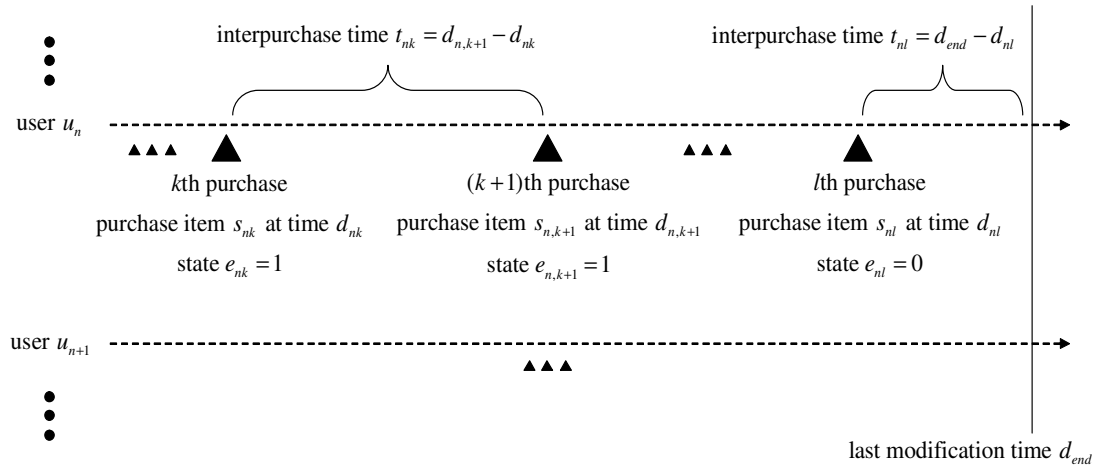


Figure 4.2: Relationships among interpurchase times, purchase times, the last modification time, and status.

purchase only a few items. Frailty models can account for such heterogeneity across users by incorporating a user specific effect into models. Let $h(t|\boldsymbol{x}, u)$ be a hazard function that represents the instantaneous rate of purchases in interpurchase time t of user u with purchase history $\boldsymbol{x}$. See Appendix A.1 for details about hazard functions. $\boldsymbol{x} = (x_b)_{b \in \boldsymbol{B}}$ is a column vector of features for the purchase history, where $\boldsymbol{B}$ is a set of feature indices. Examples of features include whether the user has purchased item s_i, or whether the user has purchased item s_i directly after item s_j. In frailty models, the hazard function $h(t|\boldsymbol{x}, u)$ can be represented as follows:

$$h(t|\boldsymbol{x}, u) = \lambda_0(t)\lambda_u \exp(\boldsymbol{\lambda}^T \boldsymbol{x}), \tag{4.5}$$

where $\lambda_0(t)$ is the baseline hazard function, λ_u is the frailty effect of user u for handling heterogeneity, and $\boldsymbol{\lambda} = (\lambda_b)_{b \in \boldsymbol{B}}$ is an unknown parameter vector. Under the frailty models, the global optimum of the estimation is guaranteed, and $Q(l|u, s')$ can be written in a closed form as described below.

We can estimate unknown parameters $\boldsymbol{\lambda}_u = \{\lambda_{u_n}\}_{n=1}^N$ and $\boldsymbol{\lambda}$ by maximizing the log partial likelihood using optimization methods such as quasi-Newton methods. See Appendix A.2 for details. The log partial likelihood with the Breslow approximation [10] is defined as follows:

$$\begin{aligned} PL(\boldsymbol{\lambda}_u, \boldsymbol{\lambda}) &= \log \prod_{t \in \boldsymbol{T}} \frac{\prod_{(n,k) \in \boldsymbol{D}(t)} h(t|\boldsymbol{x}_{nk}, u_n)}{\left(\sum_{(m,j) \in \boldsymbol{E}(t)} h(t|\boldsymbol{x}_{mj}, u_m)\right)^{|\boldsymbol{D}(t)|}} \\ &= \sum_{t \in \boldsymbol{T}} \sum_{(n,k) \in \boldsymbol{D}(t)} \left(\log \lambda_{u_n} + \boldsymbol{\lambda}^T \boldsymbol{x}_{nk}\right) \\ &\quad - \sum_{t \in \boldsymbol{T}} |\boldsymbol{D}(t)| \log \sum_{(m,j) \in \boldsymbol{E}(t)} \lambda_{u_m} \exp(\boldsymbol{\lambda}^T \boldsymbol{x}_{mj}), \end{aligned} \tag{4.6}$$

where $\boldsymbol{T}$ is a set of interpurchase times, $\boldsymbol{D}(t) = \{(n,k)|t_{nk} = t \wedge e_{nk} = 1\}$ is the set of purchases for which interpurchase time is equal to t and which is not the last purchase. $\boldsymbol{E}(t) = \{(n,k)|t_{nk} \geq t\}$ is the set of purchases for which interpurchase time is no less than t, and $\boldsymbol{x}_{nk}$ is the feature vector of the purchase history of user u_n at the kth purchase. Note that we do not need to estimate the baseline hazard function $\lambda_0(t)$ in the estimation of unknown parameters $\boldsymbol{\lambda}_u$ and $\boldsymbol{\lambda}$.

In frailty models, features that have high $\lambda_b \in \boldsymbol{\lambda}$ (> 0) characterize purchase patterns in a short interpurchase time, which is equivalent to a high purchase frequency, and features that have low $\lambda_b \in \boldsymbol{\lambda}$ (< 0) characterize patterns in a long interpurchase time. These patterns are informative for the online store. For example, they enable the store to understand the relationship between purchase history and purchase frequency, or to determine new items to be distributed to increase purchase frequency.

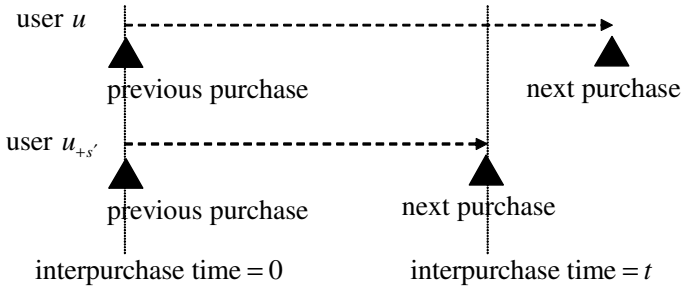
4.3.3 Probability of increasing the purchase frequency given a purchased item

With measured services, if the interpurchase time is shortened, the purchase frequency or LTV increases. Therefore, we assume that $Q(l|u, s')$ is the probability of shortening the interpurchase time when user u purchases item s'. We derive $Q(l|u, s')$ from hazard function $h(t|\boldsymbol{x}, u)$.

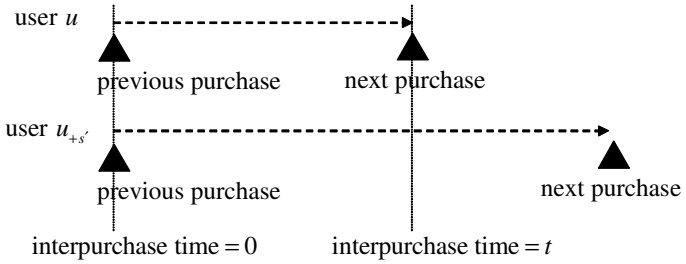
Let $\boldsymbol{x}$ be the purchase history of user u, and let $\boldsymbol{x}_{+s'}$ be the updated purchase history when user u purchases item s'. For simplicity, we refer to the user who purchases item s' as $u_{+s'}$. We assume that either u or $u_{+s'}$ purchases an item at interpurchase time t, while neither user purchases the next item before interpurchase time t (Figure 4.3), and either u or $u_{+s'}$ purchases an item at some future time, $\sum_{t=0}^{\infty} Pr(u \text{ or } u_{+s'} \text{ purchases an item at interpurchase time } t) = 1$. At t, the hazard functions of u and $u_{+s'}$ are $h(t|\boldsymbol{x}, u)$ and $h(t|\boldsymbol{x}_{+s'}, u_{+s'})$, respectively, where we assume that the frailty effect does not change by the purchase, $\lambda_u = \lambda_{u_{+s'}}$. The probability that user $u_{+s'}$ purchases the item at t (case 1 in Figure 4.3) is equal to the probability of shortening the interpurchase time when user u purchases item s' as follows:

$$
\begin{aligned}
Q(l|u, s') &= Pr(\text{interpurchase time of } u_{+s'} \text{ is shorter than that of } u) \\
&= \sum_{t=0}^{\infty} Pr(u \text{ or } u_{+s'} \text{ purchases an item at interpurchase time } t) \\
&\quad \times Pr(u_{+s'} \text{ is the one who purchases an item at } t) \\
&= \sum_{t=0}^{\infty} Pr(u \text{ or } u_{+s'} \text{ purchases an item at interpurchase time } t) \\
&\quad \times \frac{h(t|\boldsymbol{x}_{+s'}, u_{+s'})}{h(t|\boldsymbol{x}, u) + h(t|\boldsymbol{x}_{+s'}, u_{+s'})} \\
&= \frac{1}{1 + \exp\Big(-\boldsymbol{\lambda}^T(\boldsymbol{x}_{+s'} - \boldsymbol{x})\Big)}, \qquad (4.7)
\end{aligned}
$$

which is a sigmoid function. The probability that user $u_{+s'}$ purchases an item at t is fixed across interpurchase time t because the baseline hazard function $\lambda_0(t)$ is canceled out by considering the ratio of hazard functions in frailty models. Note that the probability of improving LTV does not depend on frailty effect λ_u. While we can recommend an item that maximizes $Q(l|u, s')$, the user may not purchase the recommended item if the user is not interested in the item at all. In this case, the recommendation is useless with respect to improving LTV. Therefore, it is necessary to consider whether the recommended item is purchased by the user taking the user's interests into consideration.



CASE 1: User $u_{+s'}$ purchases an item.



CASE 2: User u purchases an item.

Figure 4.3: User u or user $u_{+s'}$ purchases an item at interpurchase time t.

4.3.4 Probability of purchasing an item given a recommendation

The estimation of $R\big(s'|u, r(s)\big)$, which is the probability that user u purchases item s' when item s is recommended, is now explained. Let $R(s'|u)$ be the probability that user u purchases item s' without recommendations, where $\sum_{s' \in \boldsymbol{S}} R(s'|u) = 1$. The recommendation of item s increases the probability of the item being purchased. We assume that the probability increases γ times as follows:

$$R\big(s'|u, r(s)\big) = \begin{cases} \frac{1}{Z\left(u, r(s)\right)} \gamma R(s'|u) & s = s', \\ \frac{1}{Z\left(u, r(s)\right)} R(s'|u) & \text{otherwise}, \end{cases} \tag{4.8}$$

where $Z\big(u, r(s)\big) = 1 + (\gamma - 1)R(s|u)$ is the normalization term, and $\gamma \geq 1$. γ represents the effect of the recommendation on user purchase behavior, and depends on the way that the recommendation is presented in the online store, including considerations such as display size and position.

If an item matches the user's interests, the probability of the user purchasing the item becomes high, and if it does not match, the probability is low. Therefore,

$R(s'|u)$ represents the degree of agreement between the interests of user u and item s'. Since conventional recommendation methods suggest items that coincide with a user's interests, we can use conventional methods to obtain $R(s'|u)$. Maximum entropy models [33, 43, 64], which estimate a probabilistic distribution that maximizes entropy under the constraints of the given data, are employed. In maximum entropy models, the probability that user u purchases item s' is as follows:

$$R(s'|u) = \frac{1}{Z(u)} \exp\Big(\sum_{c \in \boldsymbol{C}} \alpha_c y_c(u, s')\Big), \tag{4.9}$$

where c is a feature index, $\boldsymbol{C}$ is a set of feature indices, α_c is an unknown parameter to be estimated, y_c is a feature of the purchase history, and $Z(u) = \sum_{s \in \boldsymbol{S}} \exp\Big(\sum_{c \in \boldsymbol{C}} \alpha_c y_c(u, s)\Big)$ is the normalization term.

The log likelihood of the maximum entropy model is:

$$\begin{aligned} L(\boldsymbol{\alpha}) &= \sum_{n=1}^{N} \sum_{k=1}^{K_n} \log R(s_{nk}|u_{nk}) \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K_n} \sum_{c \in \boldsymbol{C}} \alpha_c y_c(u_{nk}, s_{nk}) \\ &\quad - \sum_{n=1}^{N} \sum_{k=1}^{K_n} \log \sum_{s \in \boldsymbol{S}} \exp\Big(\sum_{c \in \boldsymbol{C}} \alpha_c y_c(u_{nk}, s)\Big), \end{aligned} \tag{4.10}$$

where $\boldsymbol{\alpha} = \{\alpha_c\}_{c \in \boldsymbol{C}}$ is an unknown parameter vector. The unknown parameters $\boldsymbol{\alpha}$ can be estimated by maximizing log likelihood $L(\boldsymbol{\alpha})$ using optimization techniques such as quasi-Newton methods. A Gaussian prior is used in the present experiments.

4.4 Experimental results for a measured service

4.4.1 Evaluation of purchase frequency models

The proposed recommendation method for improving LTV on measured services was evaluated by employing the log data of an online music download service in Japan from 1 April 2005. The unit time was set as one day.

To evaluate the proposed purchase frequency model, a comparison of the frailty models $h(t|\boldsymbol{x}, u) = \lambda_0(t)\lambda_u \exp(\boldsymbol{\lambda}^T \boldsymbol{x})$ that use both the purchase history and the user heterogeneity as their covariates, the Cox proportional hazards models [11] $h(t|\boldsymbol{x}) = \lambda_0(t) \exp(\boldsymbol{\lambda}^T \boldsymbol{x})$ that use the purchase history but not the user heterogeneity, and models that do not use purchase history $h(t)$ was performed. $\boldsymbol{x}_n = (x_{ni})_{i=1}^{V}$

Table 4.3: Number of users, transactions, and features in the log data for purchase frequency model evaluation.

	2005/08/31	2005/09/30	2005/10/31
number of users	10,923	13,612	17,123
number of transactions	55,416	74,582	102,165
number of features	4,234	5,662	8,283

were used as features for the frailty models and Cox proportional hazards models, where

$$x_{ni} = \begin{cases} 1 & \text{if user } u_n \text{ has purchased item } s_i, \\ 0 & \text{otherwise,} \end{cases} \tag{4.11}$$

and features that appeared fewer than ten times in the learning data were omitted.

Three sets of data consisting of the log data up to 31 August 2005, 30 September 2005, and 31 October 2005 were used. Purchase frequency models were evaluated based on the predictive performance of the last interpurchase time of each user. The number of users, transactions, and features in the data set were as listed in Table 4.3. For the predictive performance measurements, we used the perplexity as follows:

$$Perp = \exp\left(-\frac{1}{\sum_{t \in \boldsymbol{T}} |\boldsymbol{D}(t)|} \log \frac{h(t|\boldsymbol{x}_{nk}, u_n)}{\sum_{(m,j) \in \boldsymbol{E}(t)} h(t|\boldsymbol{x}_{mj}, u_m)} \right). \tag{4.12}$$

A higher perplexity for the test samples indicates a higher predictive performance of the model. Table 4.4 shows the results. The perplexities for the test samples of the frailty models were higher than those of the Cox proportional hazards models and the model that does not use purchase histories. This result shows that frailty models with both of purchase history and user heterogeneity information can predict purchase frequencies more precisely than models without them.

4.4.2 Evaluation of choice models in a measured service

The maximum entropy models described in Section 4.3.4, which estimate the probability that user u purchases item s', $R(s'|u)$, were evaluated. First-order Markov transitions were used as features because the last purchased item was considered to have revealed the user's interests:

$$y_{ij}(u, s) = \begin{cases} 1 & \text{if item } s_i \text{ is the last purchased item of user } u \text{ and } s_j = s, \\ 0 & \text{otherwise.} \end{cases} \tag{4.13}$$

Table 4.4: Perplexities of purchase frequency models.

	2005/08/31		2005/09/30		2005/10/31	
	learning	test	learning	test	learning	test
without purchase histories	30546.3	5399.2	40823.0	6680.8	56162.2	8358.2
Cox models	19850.8	3952.1	26662.1	4934.5	36497.5	6198.1
frailty models	18657.4	**3904.9**	25084.4	**4880.5**	34406.5	**6130.3**

Three sets of samples were used consisting of the log data up to 31 August 2005, 30 September 2005, and 31 October 2005, from which transitions to the same item, items that appeared fewer than ten times, and users that purchased no more than one item were omitted. We divided each set of samples into learning and test samples. The number of transitions and items were as shown in Table 4.5. We compared maximum entropy models with uniform distributions and multinomial distributions. Uniform distributions do not use the information in the log data at all. Multinomial distributions use the information about the number of each item purchased by all users, but do not consider individual interests. The unknown parameters of the multinomial distribution were estimated by the maximum likelihood method.

The perplexity was used for the evaluation measurements as follows:

$$Perp = \exp\left(-\frac{1}{\sum_{n=1}^{N} K_n}\sum_{n=1}^{N}\sum_{k=1}^{K_n}\log R(s_{nk}|u_{nk})\right). \tag{4.14}$$

Table 4.6 shows the results. The perplexities of the maximum entropy models for the test samples were higher than those of uniform and multinomial distributions. In addition, the maximum entropy models were evaluated using the accuracy of next purchase prediction, which is widely used as a evaluation measurement for collaborative filtering tasks. The item predicted to be purchased next is the highest purchase probability item, as follows:

$$\hat{s'}(u) = \arg\max_{s\in\boldsymbol{S}} R(s|u), \tag{4.15}$$

and the predictive accuracy is calculated as follows:

$$Acc = \frac{1}{\sum_{n=1}^{N} K_n}\sum_{n=1}^{N}\sum_{k=1}^{K_n} I(s_{nk} = \hat{s'}(u_{nk})) \times 100. \tag{4.16}$$

Table 4.7 shows the results. The accuracies of the maximum entropy models are the highest. This means that maximum entropy models with purchase histories

Table 4.5: Number of transitions and items in the log data of a measured service for choice model evaluation.

	2005/08/31		2005/09/30		2005/10/31	
	learning	test	learning	test	learning	test
number of transitions	45,395	4,783	61,904	6,089	85,262	8,598
number of items	1,007		1,288		1,642	

Table 4.6: Perplexities of choice models for a measured service.

	2005/08/31		2005/09/30		2005/10/31	
	learning	test	learning	test	learning	test
uniform distribution	1007.0	1007.0	1288.0	1288.0	1642.0	1642.0
multinomial distribution	279.8	283.4	356.7	297.7	437.0	414.1
maximum entropy model	120.1	**165.2**	142.2	**212.3**	163.5	**215.9**

Table 4.7: Accuracies of choice models for a measured service.

	2005/08/31		2005/09/30		2005/10/31	
	learning	test	learning	test	learning	test
uniform distribution	0.10	0.10	0.08	0.08	0.06	0.06
multinomial distribution	5.16	3.71	4.36	3.27	3.51	2.58
maximum entropy model	13.60	**7.32**	13.42	**7.62**	13.47	**8.40**

can predict user purchase behavior and interests more precisely than those without them.

4.4.3 Purchase frequencies and purchase probabilities

Conventional recommendation methods recommend items that have a high probability of being purchased. If high-purchase-frequency users tend to purchase high-purchase-probability items, conventional methods are sufficient to improve LTV. The relationship between purchase frequencies and purchase probabilities was investigated using the frailty model estimated using the log data up to 31 October 2005.

The hazard function in frailty models is multiplied by $\exp(\lambda_i)$ with the existence of feature x_i. The effect on improving LTV of the purchase of an item s_i was

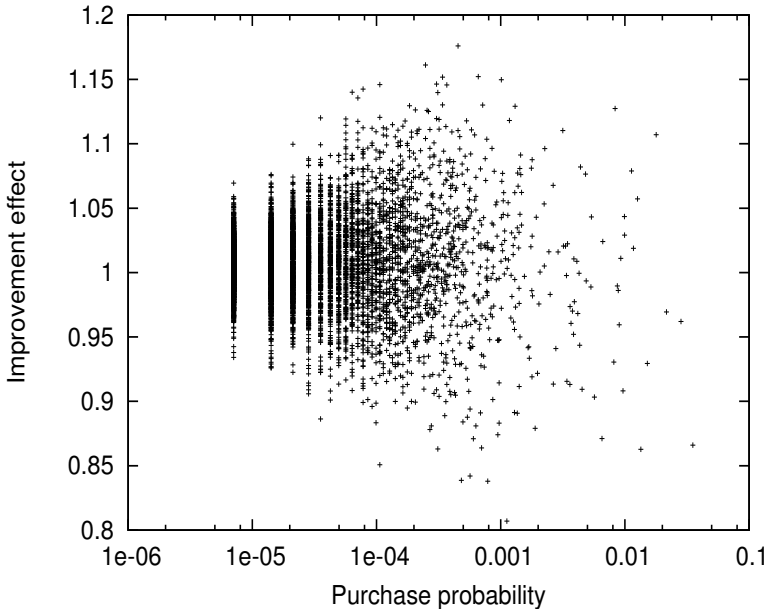


Figure 4.4: Purchase probabilities vs. LTV improving effects.

expressed by $\exp(\lambda_i)$, and the purchase probability was expressed by the multinomial distribution parameter estimated by the maximum likelihood. Figure 4.4 shows a scatter plot of the effect on the LTV due to the purchase, $\exp(\lambda_i)$, and the purchase probability, $R(s_i)$. The correlation coefficient was -0.052, and the effect on the LTV and the purchase probability are not correlated. This result implies that recommendations that suggest items that have a high probability of being purchased do not necessarily improve the LTV.

4.4.4 Simulation

In Section 4.4.1, the ability of frailty models to predict purchase frequencies was demonstrated, and in Section 4.4.2, the ability of maximum entropy models to predict user purchase behavior was demonstrated. Here, the effectiveness of the proposed method is examined by simulation. User behavior was simulated using the frailty model and the maximum entropy model that were estimated using the log data from 1 April 2005 to 31 October 2005.

The function of Algorithm 1 is to generate a purchase history, where d is the time, u is the purchase history, u_{+s} is the updated history when item s is purchased, ϕ is an empty history, $MaxTime$ is the time period for the simulation,

Algorithm 1 Simulation algorithm of a user behavior in a measured service.

1: Set $d \leftarrow 0$, $u \leftarrow \phi$
2: **while** $d \leq MaxTime$ **do**
3: **if** $u = \phi$ **then**
4: Sample $s \sim Multinomial\Big(\{R(s)\}_{s \in \boldsymbol{S}}\Big)$
5: **else**
6: $\hat{s} \leftarrow \arg\max_{s \in \boldsymbol{S}} P\big(l|u, r(s)\big)$
7: Sample $s \sim Multinomial\left(\{R(s|u, r(\hat{s}))\}_{s \in \boldsymbol{S}}\right)$
8: **end if**
9: Set $u \leftarrow u_{+s}$
10: Sample $t \sim Exponential\Big(\lambda_0 \lambda_u \exp(\boldsymbol{\lambda}^T \boldsymbol{x})\Big)$
11: Set $d \leftarrow d + t$
12: **end while**
13: Output u

$Multinomial(\boldsymbol{\psi})$ is the multinomial distribution of one event with j's success probability ψ_j, and $Exponential(\lambda)$ is the exponential distribution with parameter λ, $p(t) = \lambda \exp(-\lambda t)$. The first item that the user purchases is determined according to $R(s)$, which is the probability of purchasing item s first (line 4). If the user has purchased items, a recommendation is generated using the proposed method (line 6), and the item that the user purchases is determined according to choice model $R\Big(s|u, r(\hat{s})\Big)$ (line 7). The interpurchase time is sampled from the exponential distribution (line 10), and the time is updated (line 11). Unknown parameters $R(s|u)$, $R(s)$, and λ_u were estimated using the log data by the maximum likelihood method.

The proposed method was compared with the following recommendation methods:

- **Q Recommend** recommends an item that is most likely to increase the purchase frequency when the user purchases the item. Line 6 in Algorithm 1 is changed as follows:

$$\hat{s} \leftarrow \arg\max_{s \in \boldsymbol{S}} Q(l|u, s). \tag{4.17}$$

  This recommendation does not take the user's interests into consideration.

- **R Recommend** recommends an item that best coincides with the user's

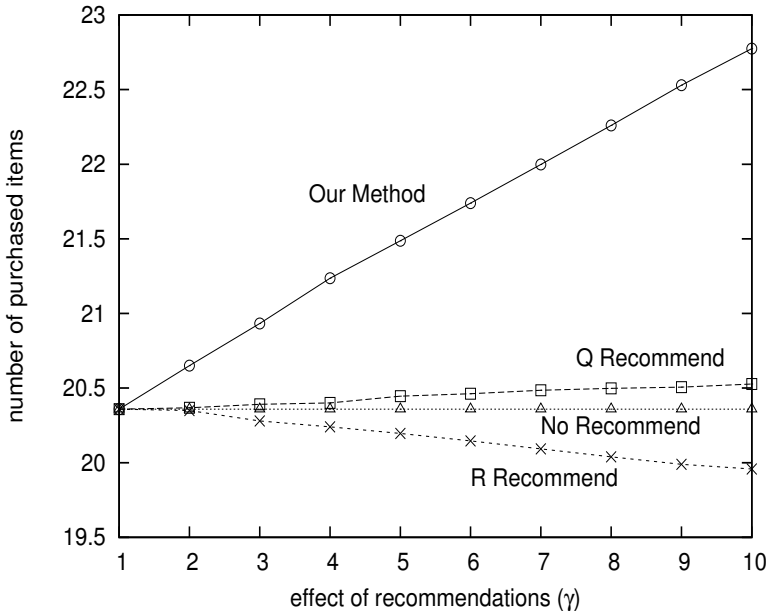


Figure 4.5: Average number of purchased items in simulations.

interests. Line 6 is changed as follows:

$$\hat{s} \leftarrow \arg\max_{s \in \boldsymbol{S}} R(s|u). \tag{4.18}$$

This recommendation is the same strategy as that of conventional methods.

- **No Recommend** does not recommend any items. The item that the user purchases is determined solely according to the user's interests. Line 6 is omitted, and line 7 is changed as follows:

$$\text{Sample } s \sim Multinomial\Big(\{R(s|u)\}_{s \in \boldsymbol{S}}\Big). \tag{4.19}$$

This recommendation can also be achieved by using $\gamma = 1$ with Algorithm 1, which means that the recommendation has no effect on user purchase behavior.

A total of $171,230$ user histories were generated with recommendations by each method where $1 \leq \gamma \leq 10$, in which each estimated λ_u was used ten times. The time period for the simulation was set to 365 days. Figure 4.5 shows the average number of purchased items. The proposed method was more successful than the others in increasing the number of purchased items. The number of purchased items

increases with increases in γ. This result indicates that if recommendations can influence user behavior, or $\gamma > 1$, the proposed method can increase the purchase frequency. Moreover, the purchase frequency can be increased further by improving the influence of the recommendations. Q Recommend also increase the number of purchased items, although the effect was smaller than that of the proposed method because Q Recommend may recommend items that have low probabilities of being purchased by the user. On the other hand, the proposed method recommends items taking user's interests into account in order to improve the recommendations. R Recommend reduces the number of purchased items because the purchase frequency is negatively correlated with the purchase probability, as shown in Figure 4.4.

4.5 Recommendation for subscription services

This section describes a recommendation method designed to improve the LTV for subscription services. This method is obtained by modifying the proposed method for measured services described in Section 4.3. With subscription services, the LTV is proportional to the subscription period and does not depend on the purchase frequency. Therefore, the probability of improving the LTV given the purchased item is modified to $Q'(l|u, s')$, which represents the probability of extending the subscription period given the purchased item.

4.5.1 Subscription period models

The subscription period is modeled using Cox proportional hazards models [11]. Let $h'(\tau|\boldsymbol{x})$ be the hazard function, which represents the instantaneous rate of unsubscription at period τ of users with purchase history $\boldsymbol{x} = (x_b)_{b\in\boldsymbol{B}}$. In Cox proportional hazards models, the hazard function $h'(\tau|\boldsymbol{x})$ is as follows:

$$h'(\tau|\boldsymbol{x}) = \beta_0(\tau)\exp(\boldsymbol{\beta}^T\boldsymbol{x}), \tag{4.20}$$

where $\beta_0(t)$ is the baseline hazard function, and $\boldsymbol{\beta} = (\beta_b)_{b\in\boldsymbol{B}}$ is the unknown parameter vector.

Let e'_n be the status of user u_n, which represents still subscribing or already unsubscribed, as follows:

$$e'_n = \begin{cases} 0 & \text{if user } u_n \text{ is still subscribing,} \\ 1 & \text{if user } u_n \text{ has already unsubscribed.} \end{cases} \tag{4.21}$$

The subscription period τ_n of user u_n is obtained as follows:

$$\tau_n = \begin{cases} d_{end} - d_n^{start} & \text{if } e'_n = 0, \\ d_n^{end} - d_n^{start} & \text{if } e'_n = 1, \end{cases} \tag{4.22}$$

Table 4.8: Example subscription log.

user	status	subscribed time	unsubscribed time
u_1	1	2004/8/16 11:50:30	2005/01/08 20:14:11
u_2	0	2004/8/16 18:01:28	
u_3	1	2004/8/17 16:10:51	2004/08/25 13:01:06
u_4	1	2004/8/17 21:39:29	2004/08/29 07:21:51
u_5	0	2004/8/18 01:44:17	
$\vdots$	$\vdots$	$\vdots$	$\vdots$
u_N	0	2005/10/28 23:10:03	

where d_n^{start} is the subscribed time of user u_n, d_n^{end} is the unsubscribed time of user u_n, and d_{end} is the last time the log was modified. The subscription period τ is assumed to be a discrete variable. The status e'_n and the subscription period τ_n are obtained from the subscription log. The subscription log consists of the subscribed time, the status and, where relevant, the unsubscribed time of each user. Table 4.8 shows an example subscription log. Figure 4.6 shows the relationships among subscription period τ, subscribed time d^{start}, unsubscribed time d^{end}, last modification time d_{end}, and status e'. Note that unsubscription is not a repeated event in the sense that one user can only unsubscribe one time. As input data for modeling subscription periods, a set of status e', subscription period τ, and purchase history $\boldsymbol{x}$, in which the subscription period at each purchase is also needed, are used, as in shown in Table 4.9. The subscription period at the kth purchase of user u_n is defined by $\tau_{nk} = d_{nk} - d_n^{start}$, where d_{nk} is the time of the the kth purchase of user u_n. Note that purchase history $\boldsymbol{x}_n$ must be treated as time-dependent variables because purchase history $\boldsymbol{x}_n$ changes when user u_n purchases an item.

Unknown parameter vector $\boldsymbol{\beta}$ can be estimated by maximizing the log partial likelihood as follows:

$$\begin{aligned} PL(\boldsymbol{\beta}) &= \log \prod_{\tau \in \boldsymbol{\Upsilon}} \frac{\prod_{n \in \boldsymbol{D}'(\tau)} h'(\tau | \boldsymbol{x}_n(\tau))}{\left(\sum_{m \in \boldsymbol{E}'(\tau)} h'(\tau | \boldsymbol{x}_m(\tau)) \right)^{|\boldsymbol{D}'(\tau)|}} \\ &= \sum_{\tau \in \boldsymbol{\Upsilon}} \sum_{n \in \boldsymbol{D}'(\tau)} \boldsymbol{\beta}^T \boldsymbol{x}_n(\tau) \\ &\quad - \sum_{\tau \in \boldsymbol{\Upsilon}} |\boldsymbol{D}'(\tau)| \log \sum_{m \in \boldsymbol{E}'(\tau)} \exp(\boldsymbol{\beta}^T \boldsymbol{x}_m(\tau)), \end{aligned} \tag{4.23}$$

where $\boldsymbol{\Upsilon}$ is a set of subscription periods, $\boldsymbol{D}'(\tau) = \{n | \tau_n = \tau \wedge e'_n = 1\}$ is the set of users unsubscribed at τ, $\boldsymbol{E}'(\tau) = \{n | \tau_n \geq \tau\}$ is the set of users subscribing

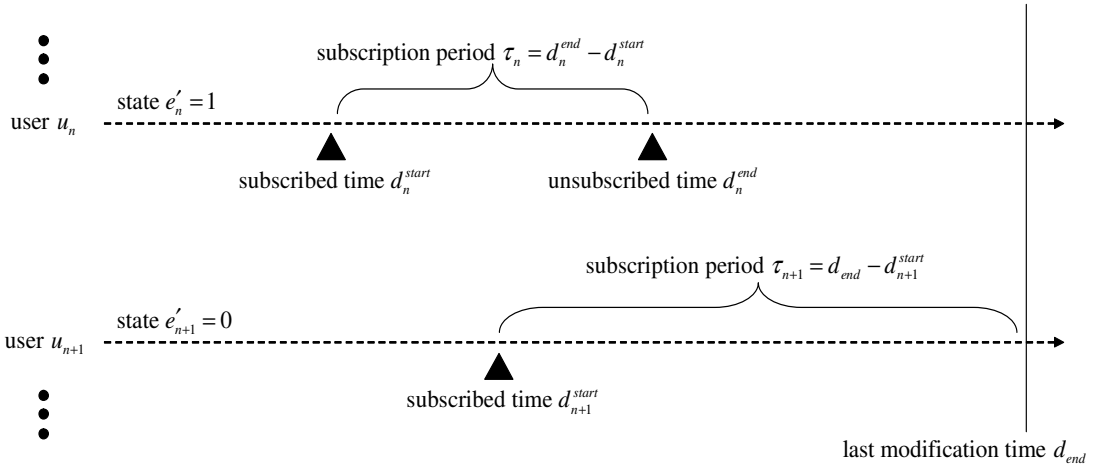


Figure 4.6: Relationships among subscription periods, subscribed times, unsubscribed times, the last modification time, and status.

Table 4.9: Example input data of the recommendation method for subscription services. The number in the parenthesis is the subscription period at which the item is purchased.

subscription period	status	purchased history (subscription period)
145	0	$s_3(0)$, $s_1(0)$, $s_6(1)$, $\cdots$
438	1	$s_2(0)$, $s_8(3)$, $s_1(5)$, $\cdots$
8	1	$s_2(0)$, $s_{13}(7)$
12	1	$s_3(0)$, $s_1(2)$, $s_2(12)$
411	0	$s_5(0)$, $s_1(0)$, $s_8(2)$, $\cdots$
$\vdots$	$\vdots$	$\vdots$

at τ, and $\boldsymbol{x}_n(\tau)$ is the feature vector of user u when the subscription period is τ. Features that have low $\beta_b \in \boldsymbol{\beta}$ (< 0) are characteristic purchase patterns for long-subscription users, and features that have high $\beta_b \in \boldsymbol{\beta}$ (> 0) are characteristic patterns for short-subscription users.

4.5.2 Probability of extending the subscription period given a purchased item

With subscription services, if the subscription period is long, the LTV increases. Therefore, $Q'(l|u,s')$ is assumed to be the probability of extending the subscription period when user u purchases item s', and is estimated from hazard function $h'(\tau|\boldsymbol{x})$ in a manner similar to that described in Section 4.3.3.

Let $\boldsymbol{x}$ be the purchase history of user u, and let $\boldsymbol{x}_{+s'}$ be the updated purchase history when item s' is purchased. For simplicity, we refer to the user when item s' is purchased as $u_{+s'}$. We assume that either u or $u_{+s'}$ unsubscribed at t while the other is still subscribing. At τ, the hazard function of u and $u_{+s'}$ are $h'(\tau|\boldsymbol{x})$ and $h'(\tau|\boldsymbol{x}_{+s'})$, respectively. The probability that user u unsubscribed at τ is equal to the probability of extending the subscription period when user u purchases item s' as follows:

$$
\begin{aligned}
Q'(l|u,s') &= Pr(\text{subscription period of } u_{+s'} \text{ is longer than that of } u) \\
&= \sum_{\tau=0}^{\infty} Pr(u \text{ or } u_{+s'} \text{ unsubscribes at } \tau) \\
&\quad \times Pr(u_{+s'} \text{ is the one who unsubscribes at } \tau) \\
&= \sum_{\tau=0}^{\infty} Pr(u \text{ or } u_{+s'} \text{ unsubscribes at } \tau) \frac{h'(\tau|\boldsymbol{x})}{h'(\tau|\boldsymbol{x}) + h'(\tau|\boldsymbol{x}_{+s'})} \\
&= \frac{1}{1+\exp\Big(-\boldsymbol{\beta}^T(\boldsymbol{x}-\boldsymbol{x}_{+s'})\Big)},
\end{aligned}
\tag{4.24}
$$

which is a sigmoid function.

The proposed method for subscription services recommends item $\hat{s}$ that maximizes $P'\big(l|u,r(s)\big)$, which is the probability of improving the LTV of user u when item s is recommended, as follows:

$$
\begin{aligned}
\hat{s} &= \arg\max_{s\in\boldsymbol{S}} P'\big(l|u,r(s)\big), \\
&= \arg\max_{s\in\boldsymbol{S}} \sum_{s'\in\boldsymbol{S}} Q'(l|u,s') R\big(s'|u,r(s)\big),
\end{aligned}
\tag{4.25}
$$

where the probability is decomposed as in (4.2).

4.6 Experimental results for a subscription service

4.6.1 Evaluation of subscription period models

The proposed recommendation method for extending subscription periods was evaluated by using the log data of an online cartoon distribution service for cell phones in Japan. With this service, users pay monthly to read cartoons on their cell phones. Some cartoons have several volumes, and some users purchased an item more than once. A cartoon that had several volumes was regarded as one item, and the unit time was set to one day. This service began on 16 August 2004, and the last modification date of the log was 28 October 2005.

In the proposed method, it is assumed that subscription periods can be estimated efficiently using purchase histories. To evaluate this assumption, the Cox proportional hazards models $h'(\tau|\boldsymbol{x}) = \beta_0(\tau)\exp(\boldsymbol{\beta}^T\boldsymbol{x})$ that use the purchase histories described in Section 4.5.1 and models that do not use purchase histories $h'(\tau)$ were compared. In addition, the following three sets of features were compared for the Cox proportional hazards models:

- **F1:** $\boldsymbol{x}_n = (x_{ni})_{i=1}^V$, in which each feature represents whether user u has purchased item s_i:

$$x_{ni} = \begin{cases} 1 & \text{if user } u_n \text{ has purchased item } s_i, \\ 0 & \text{otherwise,} \end{cases} \tag{4.26}$$

- **F2:** $\boldsymbol{x}_n = (x_{nij})_{i,j=1}^V$, in which each feature represents whether user u_n has purchased item s_i and item s_j:

$$x_{nij} = \begin{cases} 1 & \text{if user } u_n \text{ has purchased} \\ & \text{item } s_i \text{ and item } s_j, \\ 0 & \text{otherwise,} \end{cases} \tag{4.27}$$

- **F3:** $\boldsymbol{x}_n = (x_{n,i\to j})_{i,j=1}^V$, in which each feature represents whether user u_n has purchased item s_j next to item s_i:

$$x_{n,i\to j} = \begin{cases} 1 & \text{if user } u_n \text{ has purchased} \\ & \text{item } s_j \text{ next to item } s_i, \\ 0 & \text{otherwise,} \end{cases} \tag{4.28}$$

Table 4.10: Number of features for subscription period models.

	2005/06/30	2005/07/31	2005/08/31
Cox models (F1)	75	80	84
Cox models (F2)	2,671	3,159	3,485
Cox models (F3)	3,711	4,455	5,250

Table 4.11: Numbers of subscribers and unsubscribers.

	2005/06/30		2005/07/31		2005/08/31	
	learning	test	learning	test	learning	test
number of subscribers	13,284	7,221	14,669	9,608	28,409	17,028
number of unsubscribers	4,988	6,063	8,802	5,061	9,765	11,381

where features that appeared fewer than ten times in the learning data were omitted.

Three sets of learning and test samples were used. The learning samples were log data up to 30 June 2005, 31 July 2005, and 31 August 2005. The test samples were log data of subscribers on the end date of the learning samples, and the end date of the test samples was 28 October 2005. The number of features was as shown in Table 4.10, and the number of subscribers and unsubscribers were as shown in Table 4.11.

For the evaluation measurements, the following perplexity was used:

$$Perp = \exp\left(-\frac{1}{\sum_{\tau\in\boldsymbol{\Upsilon}}|\boldsymbol{D}'(\tau)|}\log\sum_{\tau\in\boldsymbol{\Upsilon}}\sum_{n\in\boldsymbol{D}'(\tau)}\frac{h'(\tau|\boldsymbol{x}_n(\tau))}{\sum_{m\in\boldsymbol{E}'(\tau)}h'(\tau|\boldsymbol{x}_m(\tau))}\right). \quad (4.29)$$

Table 4.12 shows the results. The perplexities for the test samples of the Cox proportional hazards models (especially F3) were higher than those for the model that does not use purchase histories. This result shows that Cox proportional hazards models with purchase histories as input can predict subscription periods more precisely than models without them.

4.6.2 Evaluation of choice models in a subscription service

Choice modeling for subscription services was evaluated based on the maximum entropy models described in Section 4.3.4, which estimate the probability that user u purchases item s', $R(s'|u)$ as in Section 4.4.2. First-order Markov transitions were

Table 4.12: Perplexities of subscription period models.

	2005/06/30		2005/07/31		2005/08/31	
	learning	test	learning	test	learning	test
without purchase histories	7079.8	18863.8	9556.7	12900.2	13534.5	20010.3
Cox models (F1)	6167.2	9946.7	8501.5	12644.8	12271.1	18251.5
Cox models (F2)	6057.2	9691.5	8341.5	12357.3	11214.9	18751.0
Cox models (F3)	5453.4	**9218.8**	8501.5	**11510.3**	11214.9	**17997.7**

Table 4.13: Numbers of transitions and items in the log data of a subscription service for choice model evaluation.

	2005/06/30		2005/07/31		2005/08/31	
	learning	test	learning	test	learning	test
number of transitions	300,486	122,904	382,778	171,749	459,456	197,476
number of items	75		81		86	

used as features. Three sets of learning and test samples were used. The learning samples were log data up to 30 June 2005, 31 July 2005, and 31 August 2005, from which transitions to the same item, items that appeared fewer than ten times, and users that purchased no more than one item were omitted. The test samples were log data from the end date of the learning samples to 28 October 2005, from which transitions to the same item and transitions that contained items that had not been distributed during the learning sample period were omitted. The number of transitions and items were as shown in Table 4.13. We compared maximum entropy models with uniform distributions and multinomial distributions.

The perplexity and accuracy were used for the evaluation measurements. Table 4.14 and Table 4.15 shows the results of these measurements. The maximum entropy models had higher perplexities and accuracies for the test samples than uniform and multinomial distributions. The ability to predict next purchase items using the maximum entropy models in this subscription service, as well as in the measured service, was shown.

4.6.3 Subscription periods and purchase probabilities

The relationship between the effect of extending subscription periods and the purchase probabilities was investigated, where the Cox proportional hazards model (F3) and the maximum entropy model estimated using the log data up to 31 Au-

Table 4.14: Perplexities of choice models for a subscription service.

	2005/06/30		2005/07/31		2005/08/31	
	learning	test	learning	test	learning	test
uniform distribution	75.0	75.0	81.0	81.0	86.0	86.0
multinomial distribution	48.2	71.0	51.3	107.0	53.3	86.0
maximum entropy model	35.0	**34.8**	35.9	**41.8**	36.8	**43.0**

Table 4.15: Accuracies of choice models for a subscription service.

	2005/06/30		2005/07/31		2005/08/31	
	learning	test	learning	test	learning	test
uniform distribution	1.35	1.35	1.25	1.25	1.19	1.19
multinomial distribution	7.84	4.54	7.30	4.25	5.99	11.35
maximum entropy model	14.64	**14.20**	15.02	**15.27**	15.18	**14.29**

gust 2005 were used.

The expected subscription period given the purchase history in Cox proportional hazards models is multiplied by $\exp(-\beta_{i\to j})$ with the existence of feature $x_{i\to j}$ that represents the existence of the purchase of item s_j next to item s_i. The effect on extending subscription periods of a transition was expressed by $\exp(-\beta_{i\to j})$. The probability of the transition was estimated using maximum entropy models. Note that the features of the Cox proportional hazards model (F3) and the maximum entropy model are both first-order Markov transitions. Figure 4.7 shows a scatter plot of the extending effects of transitions, $\exp(-\beta_{i\to j})$, and their transition probabilities, $R(s_j|s_i)$. The correlation coefficient was 0.159, and there was little correlation. This result implies that recommendations of high-purchase-probability items do not necessarily lead to an extended subscription period.

4.6.4 Simulation

The effectiveness of the proposed recommendation method for subscription services was examined by simulation. User behavior was simulated using the Cox proportional hazards model and the maximum entropy model that was estimated using the log data from 16 August 2004 to 28 October 2005. The log data comprised 107 items.

The function of Algorithm 2 is to generate a subscription period τ, where u is the

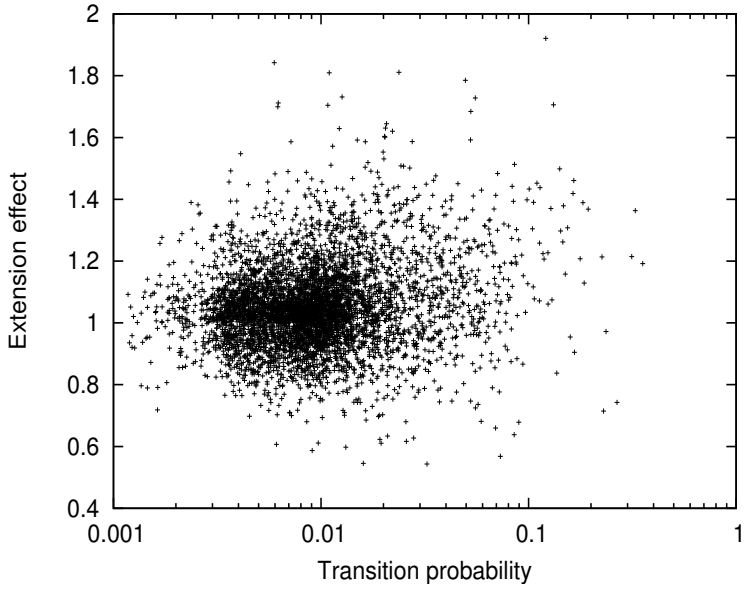


Figure 4.7: Transition probabilities vs. subscription extension effects.

sequence of purchased items, u_{+s} is the updated sequence when item s is purchased, ϕ is an empty sequence, $Bernoulli(\theta)$ is the Bernoulli distribution with success probability θ, $MaxTime$ is the time period for the simulation, and $Multinomial(\boldsymbol{\psi})$ is the multinomial distribution of one event with j's success probability ψ_j. First, from line 3 to line 4 in Algorithm 2, whether the user unsubscribes in unit time is decided using the unsubscription probability in unit time of a subscriber $h'(\tau|\boldsymbol{x})$. Second, from line 7 to line 8, whether the user purchases an item in unit time is decided using the purchase probability in unit time, g. We assumed that g is constant over subscription period τ. The first item that the user purchases is determined according to $R(s)$, where $R(s)$ is the probability of purchasing item s first (line 10). If the user has purchased some items, a recommendation is made using the proposed method (line 12), and the item that the user purchases is determined according to $R\Big(s|u, r(\hat{s})\Big)$ (line 13). Unknown parameters g, and $R(s)$ were estimated using the log data by the maximum likelihood method.

The proposed method is compared with the following recommendation methods:

- **Q Recommend** recommends an item that is most likely to extend the subscription period when the user purchases the item. Line 12 in Algorithm 2 is changed as follows:

$$\hat{s} \leftarrow \arg\max_{s} Q'(l|u, s). \tag{4.30}$$

Algorithm 2 Simulation algorithm of a user behavior in a subscription service.

1: Set $\tau \leftarrow 0$, $u \leftarrow \phi$
2: **while** $\tau \leq MaxTime$ **do**
3: Sample $r_1 \sim Bernoulli\Big(h'(\tau|\boldsymbol{x})\Big)$
4: **if** r_1 is success **then**
5: break
6: **end if**
7: Sample $r_2 \sim Bernoulli(g)$
8: **if** r_2 is success **then**
9: **if** $u = \phi$ **then**
10: Sample $s \sim Multinomial\Big(\{R(s)\}_{s\in\boldsymbol{S}}\Big)$
11: **else**
12: $\hat{s} \leftarrow \arg\max_{s\in\boldsymbol{S}} P'\big(l|u, r(s)\big)$
13: Sample $s \sim Multinomial\Big(\{R(s|u, r(\hat{s}))\}_{s\in\boldsymbol{S}}\Big)$
14: **end if**
15: Set $u \leftarrow u_{+s}$
16: **end if**
17: Set $\tau \leftarrow \tau + 1$
18: **end while**
19: Output τ

This recommendation does not take the user's interests into consideration.

- **R Recommend** recommends an item that best coincides with the user's interests. Line 12 is changed as follows:

$$\hat{s} \leftarrow \arg\max_{s\in\boldsymbol{S}} R(s|u). \tag{4.31}$$

This recommendation is the same as that of conventional methods.

- **No Recommend** does not recommend any items. The item that the user purchases is determined solely according to the user's interests. Line 12 is omitted and line 13 is changed as follows:

$$\text{Sample } s \sim Multinomial\Big(\{R(s|u)\}_{s\in\boldsymbol{S}}\Big). \tag{4.32}$$

A total of $100,000$ user subscription periods were generated through recommendations by each method, where $1 \leq \gamma \leq 10$. The maximum subscription period was set at 365 days. Figure 4.8 shows the average subscription period. The proposed

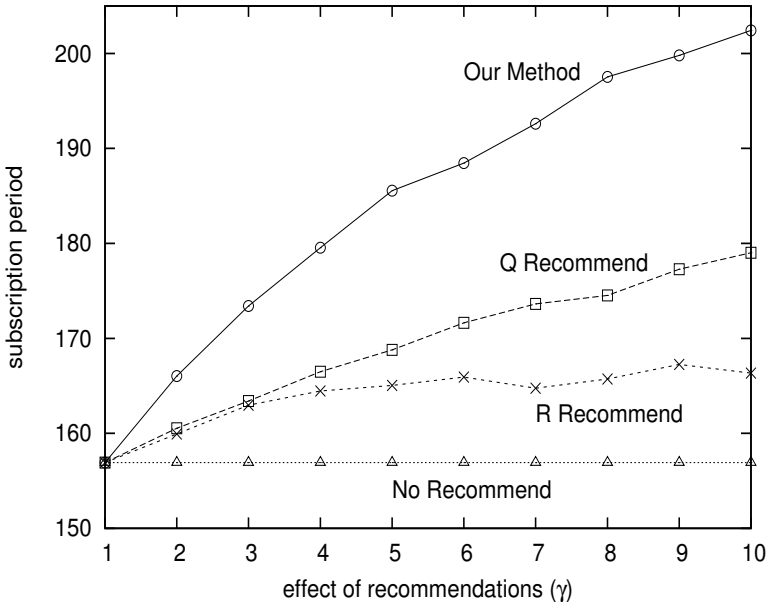


Figure 4.8: Average subscription periods in simulations.

method was more successful than the others in extending subscription periods. Since Q Recommend may recommend items that have rare probabilities of being purchased by the user, the effect of Q Recommend is smaller than that of the proposed method. R Recommend only slightly extended subscription periods because there was little correlation between the subscription periods and the purchase probability, as shown in Figure 4.7.

4.7 Summary

In this chapter, a recommendation method was proposed for improving the LTV, which encourages users to purchase more items for measured services and encourages users to extend subscription periods for subscription services. Basic features were used in the experiments to make the novelty of the proposed framework easy to understand. The proposed method can use other features such as long distance dependencies or user attributes. Since the proposed method is divided into two modules, namely the estimation of LTV and the estimation of user's interests, it can be further enhanced using survival analysis or collaborative filtering techniques. For example, the present approach can be combined with content filtering for modeling item choices. The frailty model or Cox proportional hazards model can also

be improved by including the feature selection process in order to find informative purchase patterns among high-LTV users.

Further research will be conducted in the future. First, the proposed method should be made applicable to more general business models. For example, the LTV was modeled in measured services assuming the profit generated by an item to be constant for all items. Although the profit is almost constant for all items in the online music store from which the data for the experiments were obtained, some online stores sell items of the wide price range, such as electronics and books. In these cases, the purchase frequency should be modeled considering item price information. Second, the influence of recommendations on user purchase behavior must be estimated from the log data automatically. This can be achieved by using the log data of purchase histories with and without recommendations. Finally, the proposed method should be applied to an online store, and the improvement in the LTV of real users should be examined.

Chapter 5

Conclusion and future research

In the present thesis, probabilistic user behavior models were proposed to tailor recommender systems to diverse requirements using heterogeneous information. In Chapter 2, a learning framework was proposed for obtaining a model that accurately predicts present data, in which past data are effectively used based on the similarity between the distribution at the present time and the distribution for each time. In Chapter 3, an efficient probabilistic choice model that achieves both fast parameter estimation and high predictive accuracy by combining multiple simple Markov models based on the maximum entropy principle was presented. In Chapter 4, it is shown that recommendations for improving customer lifetime values can be achieved by integrating choice models and purchase frequency or subscription period models based on a probabilistic framework. Thus, different types of behaviors can be integrated for effective recommendations by using probabilistic models.

Although encouraging results have already been obtained, the present approach must be extended before it can become a useful tool for recommendations. In the present thesis, basic probabilistic models, such as item choices, purchase frequency, and stop visiting, were proposed that use purchase log data or subscription log data. The models can be improved using other data. In the case of cartoon data, for example, the following item information can be taken into consideration for improving the predictive performance of choice models: author, publication date, and genre, such as female-oriented, hard-boiled, comedy, or sports. This item information is especially helpful for choice models for users who purchase only a few items or for items that are rarely purchased. The predictive performance of choice models can be improved using the recommendation log data that contains the information of the recommended item for each transactions because item choices are affected by the recommendations. The probability that the recommended item is purchased increases because the user has not seen the item before or the user

is newly attracted to the item by the recommendation. The access log data, by which items accessed by each use can be determined with the time stamp, even if the user does not purchase the item, are also valuable. Using the access log, the user's interests or the behavioral pattern can be estimated, such as the fact that the user often visits the store in the middle of the night or on Sunday. The ease of obtaining such information is one advantage of online stores compared with offline stores because it is difficult for offline stores to accumulate such data.

The way the presentation of the recommendation is an important factor for effective recommendations, although this was not considered in the present thesis. For example, the effectiveness of recommendation will change depending on the number of recommended items. If we recommend only one item, it is not likely to be purchased. If we recommend too many items, users may neglect all of the recommendations. By estimating the optimal number of recommendations for each user, we can improve the recommendation effect. Online stores can recommend many items to users who are easily affected by recommendations, and can avoid making recommendations to users who do not purchase recommended items. The probability that users click the recommendation also depends on whether the store provides images or abstracts of items with recommendations. The recommendations need not to be presented in the form of the item list. By using dimensionality reduction methods, we can locate items in the two-dimensional map, in which similar items are located closely. With the map, users can intuitively understand the relationships among a large number of items and browse through items. Since probabilistic dimensionality reduction methods have been proposed [20, 27], a map can be created based on the probabilistic behavior models presented in the present thesis.

Although the focus of the present thesis is modeling behaviors in online stores, it is hoped that the present research will be extended to model behaviors on the entire WWW. The log data used in the experiments were obtained from stores for specific products, such as music, movies, or cartoons. In order to model WWW behaviors, the integration of a wide variety of heterogeneous items must be considered. Furthermore, it is hoped that the present research can be extended to model behaviors in real life. To obtain personal behavioral data in real life has become easier with the rapid progress of technology. For example, daily behavioral data can be gathered using a cell phone with GPS. The framework presented in this thesis is also expected to be useful for considering behavior models on the entire WWW and in real life.

Acknowledgments

I would like to express my gratitude to Prof. Toshiyuki Tanaka for supervising this thesis. His helpful suggestion improved the quality of my research. I would also like to thank Prof. Shin Ishii and Prof. Akihiro Yamamoto for their useful comments.

The research for this thesis was conducted at Communication Science Laboratories (CS labs), Nippon Telegraph and Telephone (NTT) Corporation. I began researching machine learning when I joined NTT. Through fruitful discussions with Prof. Kazumi Saito, my supervisor at that time, I learned the essential concepts and techniques of machine learning. I am indebted to Dr. Naonori Ueda, executive manager of our laboratory, for his continuous encouragement and valuable advice from the viewpoint of a specialist of the statistical learning theory. I would like to thank to Dr. Takeshi Yamada, leader of our group, for his considerate guidance and broad support. I wish to thank to Akinori Fujino for providing valuable technical knowledge. I would also like to thank the members of the Emergent Learning and Systems Research Group at NTT CS Labs for fruitful discussions. I would like to express my appreciation to Prof. Noboru Sugamura and Prof. Shigeru Katagiri, former directors of NTT CS labs, and Dr. Yoshinobu Tonomura, the director of NTT CS labs, for providing the opportunity to undertake such interesting research.

I would like to thank to Dr. Takashi Ikegami, my master's degree adviser at the University of Tokyo, Graduate School of Arts and Sciences, and my colleagues at the Ikegami Laboratory for their insightful and exciting discussions.

I am indebted to Prof. Masaru Tomita, who was my supervisor when I was an undergraduate student at Keio University, Faculty of Environment and Information Studies, and my colleagues at the Tomita Laboratory for teaching me the basics of research and allowing me to learn the joy of research.

Finally, and most importantly, I am grateful to my parents for their love and support.

Bibliography

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward 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, 2005.

[2] Wai Ho Au, Keith C. C. Chan, and Xin Yao. A novel evolutionary data mining algorithm with applications to churn prediction. *IEEE Transactions on Evolutionary Computation*, 7(6):532–545, 2003.

[3] Michael J. A. Berry and Gordon S. Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. John Wiley, 2004.

[4] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *ICML '07: Proceedings of the 24th International Conference on Machine Learning*, pages 81–88, New York, NY, USA, 2007. ACM Press.

[5] Steffen Bickel and Tobias Scheffer. Dirichlet-enhanced spam filtering based on biased samples. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 161–168, Cambridge, MA, USA, 2007. MIT Press.

[6] David M. Blei and Michael I. Jordan. Modeling annotated data. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 127–134, New York, NY, USA, 2003. ACM.

[7] David M. Blei and John D. Lafferty. Dynamic topic models. In *ICML '06: Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120, New York, NY, USA, 2006. ACM Press.

[8] Jan Box-Steffensmeier and Brad Jones. *Event History Modeling*. Cambridge University Press, 2004.

[9] Stanley F. Chen and Ronald Rosenfeld. A Gaussian prior for smoothing maximum entropy models. Technical report, CMUCS–99–108, 1999.

[10] Mario Cleves, William W. Gould, and Roberto Gutierrez. *An Introduction to Survival Analysis Using Stata, Revised Edition*. Stata Press, 2004.

[11] David R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society, Series B*, 34(2):187–220, 1972.

[12] Hal Daume III. Frustratingly easy domain adaptation. In *ACL '07: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, 2007. Association for Computational Linguistics.

[13] Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126, 2006.

[14] Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[15] Laura Dietz, Steffen Bickel, and Tobias Scheffer. Unsupervised prediction of citation influences. In *ICML '07: Proceedings of the 24th International Conference on Machine learning*, pages 233–240, New York, NY, USA, 2007. ACM Press.

[16] Yi Ding and Xue Li. Time weight collaborative filtering. In *CIKM '05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 485–492, New York, NY, USA, 2005. ACM Press.

[17] Peter S. Fader, Bruce G. S. Hardie, and Ka Lok Lee. "Counting your customers" the easy way: An alternative to the Pareto/NBD model. *MARKETING SCIENCE*, 24(2):275–284, 2005.

[18] Akinori Fujino, Naonori Ueda, and Kazumi Saito. A hybrid generative / discriminative approach to text classification with additional information. *Information Processing and Management*, 43:379–392, 2007.

[19] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[20] Geoffrey E. Hinton and Sam T. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, pages 833–840, Cambridge, MA, USA, 2002. MIT Press.

[21] Thomas Hofmann. Probabilistic latent semantic analysis. In *UAI '99: Proceedings of 15th Conference on Uncertainty in Artificial Intelligence*, pages 289–296, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[22] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the Annual International SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57, 1999.

[23] Thomas Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 259–266, New York, NY, USA, 2003. ACM Press.

[24] Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In *IJCAI '99: Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 688–693, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[25] Jiayuan Huang, Alex Smola, Arthur Gretton, Karsten Borgwardt, and Bernhard Schoelkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems 18*, Cambridge, MA, USA, 2007. MIT Press.

[26] Joseph George Ibrahim, Ming-Hui Chen, and Debajyoti Sinha. *Bayesian Survival Analysis*. Springer, 2001.

[27] Tomoharu Iwata, Kazumi Saito, Naonori Ueda, Sean Stromsten, Thomas L. Griffiths, and Joshua B. Tenenbaum. Parametric embedding for class visualization. *Neural Computation*, 19(9):2536–2556, 2007.

[28] Tomoharu Iwata, Kazumi Saito, and Takeshi Yamada. Recommendation methods for extending subscription periods. In *KDD '06: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Miniining*, pages 574–579, 2006.

[29] Frederick Jelinek and Robert L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, 1980.

[30] Lichung Jen, Chien-Heng Chou, and Greg M. Allenby. A Bayesian approach to modeling purchase frequency. *Marketing Letters*, 14(1):5–20, 2003.

[31] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *ACL '07: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic, 2007. Association for Computational Linguistics.

[32] Xin Jin, Bamshad Mobasher, and Yanzan Zhou. A web recommendation system based on maximum entropy. In *ITCC '05: Proceedings of the International Conference on Information Technology: Coding and Computing - Volume I*, pages 213–218, Washington, DC, USA, 2005. IEEE Computer Society.

[33] Xin Jin, Yanzan Zhou, and Bamshad Mobasher. A maximum entropy web recommendation system: combining collaborative and content features. In *KDD '05: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2005.

[34] Elisa T. Lee and John Wenyu Wang. *Statistical Methods for Survival Data Analysis*. WWiley-Interscience, 2003.

[35] Yi Lin, Yoonkyung Lee, and Grace Wahba. Support vector machines for classification in nonstandard situations. *Machine Learning*, 46(1-3):191–202, 2002.

[36] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 07(1):76–80, 2003.

[37] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(3):503–528, 1989.

[38] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *COLING '02: Proceeding of the 6th Conference on Natural language learning*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[39] D. R. Mani, James Drew, Andrew Betz, and Piew Datta. Statistics and data mining techniques for lifetime value modeling. In *KDD '99: Proceedings of the 5th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 94–103, 1999.

[40] Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the 5th ACM Conference on Digital Libraries*, pages 195–204, 2000.

[41] Michael C. Mozer, Richard Wolniewicz, David B. Grimes, Eric Johnson, and Howard Kaushansky. Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. *IEEE Transactions on Neural Networks*, 11(3):690–696, 2000.

[42] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[43] Dmitry Y. Pavlov and David M. Pennock. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. In *Advances in Neural Information Processing Systems*, pages 1441–1448, 2002.

[44] Gregory Piatetsky-Shapiro and Brij Masand. Estimating campaign benefits and modeling lift. In *KDD '99: Proceedings of the 5th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 185–193, 1999.

[45] Alexandrin Popescul, Lyle Ungar, David Pennock, and Steve Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *UAI '01: Proceedings of 17th Conference on Uncertainty in Artificial Intelligence*, pages 437–444, 2001.

[46] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1988.

[47] Adrian E. Raftery. A model for high-order Markov chains. *Journal of the Royal Statistical Society B*, 47(3):528–539, 1985.

[48] Rajat Raina, Yirong Shen, Andrew Y. Ng, and Andrew McCallum. Classification with hybrid generative / discriminative models. In *Advances in Neural Information Processing Systems*, 2004.

[49] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language*, pages 133–142, 1996.

[50] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 175–186, 1994.

[51] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *UAI '04: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

[52] Saharon Rosset, Einat Neumann, Uri Eick, and Nurit Vatnik. Customer lifetime value models for decision support. *Data Mining and Knowledge Discovery*, 7:321–339, 2003.

[53] Badrul Sarwar, George Karypis, Joseph Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th International Conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM Press.

[54] Lawrence K. Saul and Michael I. Jordan. Mixed memory Markov models: Decomposing complex stochastic processes as mixtures of simpler ones. *Machine Learning*, 37(1):75–87, 1999.

[55] J. Ben Schafer, Joseph A. Konstan, and John Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5:115–153, 2001.

[56] Guy Shani, David Heckerman, and Ronen I. Brafman. An MDP-based recommender system. *Journal of Machine Learning Research*, 6:1265–1295, 2005.

[57] Upendra Shardanand and Patti Maes. Social information filtering: Algorithms for automating "word of mouth". In *CHI '95: Proceedings of ACM Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.

[58] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[59] Yuji Shono, Yohei Takada, Norihisa Komoda, Hriaki Oiso, Ayako Hiramatsu, and Kiyoyuki Fukuda. Customer analysis of monthly-charged mobile content aiming at prolonging subscription period. In *Proceedings of IEEE Conference on Computational Cybernetics*, pages 279–284, 2004.

[60] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. Technical Report TR06-0007, Department of Computer Science, Tokyo Institute of Technology, 2006.

[61] Xuerui Wang and Andrew McCallum. Topics over time: a non-Markov continuous-time model of topical trends. In *KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge discovery and data mining*, pages 424–433, New York, NY, USA, 2006. ACM Press.

[62] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007.

[63] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *ICML '04: Proceedings of the 21st International Conference on Machine Learning*, New York, NY, USA, 2004. ACM Press.

[64] Lawrence Zitnick and Takeo Kanade. Maximum entropy for collaborative filtering. In *UAI '04: Proceedings of 20th Conference on Uncertainty in Artificial Intelligence*, pages 636–643, 2004.

Appendix A

A.1 Survival analysis

Survival analysis is used for modeling time to a certain event, such as death or failure of machines. The hazard function $h(t)$ is the instantaneous rate of death at time t, and is defined as follows:

$$h(t) = \lim_{\Delta t \to 0} \frac{Pr(t < T < t + \Delta t | T > t)}{\Delta t} = \frac{f(t)}{S(t)}, \tag{A.1}$$

where T is a random variable that represents the time of death, and $Pr(t < T < t + \Delta t | T > t)$ is the probability of death between t and $t + \Delta t$ conditioned on the survival until t. $S(t)$ is a survival function that represents the probability of survival until t as follows:

$$S(t) = Pr(T > t) = 1 - \int_0^t f(\tau) d\tau, \tag{A.2}$$

where $f(t)$ is a density function. If the hazard function $h(t)$, survival function $S(t)$, or density function $f(t)$ is specified, the others are fully determined as follows:

$$h(t) = \frac{-d \log S(t)}{S(t)} = \frac{f(t)}{1 - \int_0^t f(\tau) d\tau}, \tag{A.3}$$

$$S(t) = \exp(-\int_0^t h(\tau) d\tau) = 1 - \int_0^t f(\tau) d\tau, \tag{A.4}$$

$$f(t) = h(t) \exp(-\int_0^t h(\tau) d\tau) = \frac{-dS(t)}{dt}. \tag{A.5}$$

Using survival analysis techniques, we can handle data including censored samples. For example, if users are still subscribing, we cannot know their true subscription period. These samples are called censored samples.

A.2 Log-linear models

All frailty models for purchase frequency modeling, Cox proportional hazards models for subscription period modeling, and maximum entropy models for choice modeling are categorized as log-linear models, in which the probability of class y given feature vector $\boldsymbol{x} = (x_j)_{j=1}^V$ is defined as follows:

$$P(y|x) = \frac{\exp(\boldsymbol{\lambda}_y^T \boldsymbol{x})}{\sum_{y' \in \boldsymbol{Y}} \exp(\boldsymbol{\lambda}_{y'}^T \boldsymbol{x})}, \tag{A.6}$$

where $\boldsymbol{\lambda}_y = (\lambda_{yj})_{j=1}^V$ is an unknown parameter vector for y to be estimated, and $\boldsymbol{Y}$ is a set of classes. Given a set of training samples $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, unknown parameters $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}_y)_{y \in \boldsymbol{Y}}$ can be estimated by maximizing the following log likelihood with a Gaussian prior with mean $\boldsymbol{0}$ and variance $\beta^{-1}\boldsymbol{I}$ for $\boldsymbol{\Lambda}$:

$$\begin{aligned} L(\boldsymbol{\Lambda}) &= \sum_{n=1}^N \log P(y_n|x_n) + \log P(\boldsymbol{\Lambda}) \\ &= \sum_{n=1}^N \Big(\boldsymbol{\lambda}_{y_n}^T \boldsymbol{x}_n - \log \sum_{y \in \boldsymbol{Y}} \exp(\boldsymbol{\lambda}_y^T \boldsymbol{x}_n) \Big) - \frac{\beta}{2} \sum_{y \in \boldsymbol{Y}} \parallel \boldsymbol{\lambda}_y \parallel^2 . \end{aligned} \tag{A.7}$$

For the maximization, the limited memory BFGS quasi-Newton method, which is a gradient-based optimization algorithm that has been shown to be superior for learning large-scale log-linear models [38], can be used. The gradient of the log likelihood with regard to $\boldsymbol{\lambda}_y$ is as follows:

$$\frac{\partial L(\boldsymbol{\Lambda})}{\partial \boldsymbol{\lambda}_y} = \sum_{n=1}^N I(y_n = y)\boldsymbol{x}_n - \sum_{n=1}^N P(y|\boldsymbol{x}_n)\boldsymbol{x}_n - \beta \boldsymbol{\lambda}_{\boldsymbol{y}}. \tag{A.8}$$

The Hessian of the log likelihood is negative definite and the global optimum of the estimation is guaranteed.

A.3 Hyper-parameter estimation for multinomial distribution

We estimate hyper-parameters using leave-one-out cross-validation of the multinomial distribution. In multinomial distributions, the probability of feature vector $\boldsymbol{x} = (x)_{j=1}^V$ is described as follows:

$$P(\boldsymbol{x}; \boldsymbol{\theta}) \propto \sum_{j=1}^V \theta_j^{x_j}, \tag{A.9}$$

where $\boldsymbol{\theta} = (\theta_j)_{j=1}^V$ is an unknown parameter vector to be estimated. Given a set of training samples $\{\boldsymbol{x}_n\}_{n=1}^N$, the MAP estimation with the Dirichlet prior is as follows:

$$\hat{\theta}_j = \frac{m_j + \alpha}{m + \alpha V}, \tag{A.10}$$

where $m_j = \sum_{n=1}^N x_{nj}$, $m = \sum_{k=1}^V \sum_{n=1}^N x_{nk}$, and α is a hyper-parameter to be estimated. The above MAP estimation can be rewritten as a linear combination of the maximum likelihood estimation and the uniform distribution as follows:

$$\hat{\theta}_j = \beta \frac{m_j}{m} + (1 - \beta)\frac{1}{V}, \tag{A.11}$$

where $\beta = \frac{N}{N+\alpha V}$. We can estimate hyper-parameter β with leave-one-out cross-validation by the following log likelihood using the Newton method:

$$\begin{aligned} L(\beta) &= \sum_{n=1}^N \log P(\boldsymbol{x}_n; \hat{\boldsymbol{\theta}}_{-n}) \\ &= \sum_{n=1}^N \sum_{j=1}^V x_{nj} \log\Big(\beta \frac{m_j - x_{nj}}{m - \sum_{k=1}^V x_{nk}} + (1 - \beta)\frac{1}{V}\Big), \end{aligned} \tag{A.12}$$

where $\hat{\boldsymbol{\theta}}_{-n} = (\hat{\theta}_{-n,j})_{j=1}^V$ is the MAP estimation for the training data without the nth sample. The gradient of the log likelihood is as follows:

$$\frac{\partial L(\beta)}{\partial \beta} = \sum_{n=1}^N \sum_{j=1}^V x_{nj} \frac{\tilde{\theta}_{-n,j} - \frac{1}{V}}{\hat{\theta}_{-n,j}}, \tag{A.13}$$

where

$$\tilde{\theta}_{-n,j} = \frac{m_j - x_{nj}}{m - \sum_{k=1}^V x_{nk}}, \tag{A.14}$$

is the maximum likelihood estimation without the nth sample, and

$$\hat{\theta}_{-n,j} = \beta \tilde{\theta} + (1 - \beta)\frac{1}{V}, \tag{A.15}$$

is the MAP estimation without the nth sample. The second-order differential of the log likelihood is as follows:

$$\frac{\partial^2 L(\beta)}{\partial \beta^2} = -\sum_{n=1}^N \sum_{j=1}^V x_{nj} \frac{\Big(\tilde{\theta}_{-n,j} - \frac{1}{V}\Big)^2}{\hat{\theta}_{-n,j}^2}, \tag{A.16}$$

which is always negative. Therefore, and the global optimum of the estimation is guaranteed. The hyper-parameter estimation with leave-one-out cross-validation for gapped Markov models follows the same procedure as that for multinomial distributions.

List of publications by the author

Journal papers

1. Tomoharu Iwata, Kazumi Saito, Takeshi Yamada, "Recommendation Method for Improving Customer Lifetime Value," IEEE Transactions on Knowledge and Data Engineering, in press

2. Tomoharu Iwata, Takeshi Yamada, Naonori Ueda, "Collaborative Filtering Efficiently using Purchase Orders," Information Processing Society of Japan Transactions on Mathematical Modeling and its Applications (TOM 20), 2008

3. Tomoharu Iwata, Kazumi Saito, "Visual Classifier Analysis using Parametric Embedding," Information Processing Society of Japan Journal, Vol.48, No.12, 2007

4. Tomoharu Iwata, Kazumi Saito, Takeshi Yamada, "Recommendation Method for Extending Subscription Periods," Information Processing Society of Japan Transactions on Mathematical Modeling and its Applications, Vol.48, No.SIG 6 (TOM 17), 65–74, 2007

5. Tomoharu Iwata, Kazumi Saito, Naonori Ueda, Sean Stromsten, Thomas L. Griffiths, Joshua B. Tenenbaum, "Parametric Embedding for Class Visualization," Neural Computation, Vol.19, No.9, 2536–2556, 2007

6. Tomoharu Iwata, Kazumi Saito, "Visualization of Anomalies using Mixture Models," Journal of Intelligent Manufacturing, Vol.16, 635–643, 2005

7. Tomoharu Iwata, Kazumi Saito, Naonori Ueda, "Class Structure Visualization by Parametric Embedding," Information Processing Society of Japan Journal, Vol.46, 2337–2346, 2005

Letters

1. Tomoharu Iwata, Kazumi Saito, Takeshi Yamada, "Recommendation to Extend Subscription Periods," Information Technology Letters, Vol.5, 109–112, 2006 (received the FIT young researcher award)

2. Tomoharu Iwata, Kazumi Saito, Naonori Ueda, "Visualization via Posterior Preserving Embedding," Information Technology Letters, Vol.3, 119–120, 2004 (received the Funai best paper award)

International conferences

1. Tomoharu Iwata, Kazumi Saito, Takeshi Yamada, "Modeling User Behavior in Recommender Systems based on Maximum Entropy," Proc. of 16th International World Wide Web Conference (WWW2007), 1281–1282, 2007

2. Masahide Kakehi, Tomoko Kojiri, Toyohide Watanabe, Takeshi Yamada, Tomoharu Iwata: "Organization of Discussion Knowledge Graph from Collaborative Learning Record", Proc. of International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES2007), LNAI 4694, Part 3, 600–607, 2007

3. Tomoharu Iwata, Kazumi Saito, Takeshi Yamada, "Recommendation Method for Extending Subscription Periods," Proc. of 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2006), 574–579, 2006

4. Tomoharu Iwata, Kazumi Saito, Naonori Ueda, "Visual Nonlinear Discriminant Analysis for Classifier Design," Proc. of 14th European Symposium on Artificial Neural Networks (ESANN2006), 283–288, 2006

5. Tomoharu Iwata, Kazumi Saito, Naonori Ueda, Sean Stromsten, Thomas L. Griffiths, Joshua B. Tenenbaum, "Parametric Embedding for Class Visualization," Advances in Neural Information Processing Systems 17 (NIPS2004), 617–624, 2005

6. Tomoharu Iwata, Kazumi Saito, "Visualization of Anomaly using Mixture Model," Proc. of 8th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES2004), LNAI 3214, Part 2, 624–631, 2004

Domestic workshops (refereed)

1. Tomoharu Iwata, Toshiyuki Tanaka, Takeshi Yamada, Naonori Ueda, "Learning the latest data when distributions differ over time," Proc. of 10th Workshop on Information-Based Induction Sciences (IBIS2007), 170–175, 2007

2. Tomoharu Iwata, Kazumi Saito, Takeshi Yamada, "Purchase Pattern Analysis of Loyal Customers in Subscription Services," Proc. of Data Engineering Workshop (DEWS2007), 2007

Domestic workshops (not refereed)

1. Tomoharu Iwata, Kazumi Saito, Takeshi Yamada, "Modeling User Behavior in Recommender Systems based on Maximum Entropy," Forum on Information Technology (FIT2007), 2007

2. Tomoharu Iwata, Takeshi Yamada, Naonori Ueda, "Recommendation Method for Improving Customer Lifetime Value," Technical Report of Institute of Electronics, Information and Communication Engineers, Vol.107, No.78, AI2007-3, 13–18, 2007

3. Tomoharu Iwata, Takeshi Yamada, Naonori Ueda, "Collaborative Filtering using Purchase Sequences," Technical Report of Institute of Electronics, Information and Communication Engineers, Vol.107, No.114, DE2007-11, 57–62, 2007

4. Tomoharu Iwata, Kazumi Saito, Takeshi Yamada, "Recommendation Method for Extending Subscription Periods," Technical Report of Information Processing Society of Japan, MPS-61, 5–8, 2007

5. Tomoharu Iwata, Kazumi Saito, Naonori Ueda, "Topic Visualization of Web Search Results using Parametric Embedding," Technical Report of Japan Society for Artificial Intelligence, SIG-KBS-A405, 51–56, 2005

6. Tomoharu Iwata, Kazumi Saito, Naonori Ueda, "Parametric Embedding for Class Visualization," Proc. of Japan Neural Network Society (JNNS2004), 20–21, 2004

7. Tomoharu Iwata, Kazumi Saito, Naonori Ueda, "Visualization of Bipartite Graph by Spherical Embedding," Proc. of Japan Neural Network Society (JNNS2004), 66–67, 2004

8. Tomoharu Iwata, Kazumi Saito, Naonori Ueda, "Visualization of Documents for Evaluation of Classification and Detection of Unique Documents," Technical Report of Information Processing Society of Japan, NL-161-4, 25–32, 2004

9. Tomoharu Iwata, Kazumi Saito, "Visualization of Anomaly using Mixture Model," Technical Report of Institute of Electronics, Information and Communication Engineers, Vol.103 No.734, NC2003-214, 121–126, 2004

Awards

1. FIT young researcher award in 2007

2. Funai best paper award in 2004