Factorizing Historical User Actions for Next-Day Purchase Prediction

BANG LIU, University of Alberta, Canada
HANLIN ZHANG, University of Alberta, Canada
LINGLONG KONG, University of Alberta, Canada
DI NIU, University of Alberta, Canada

It is common practice for many large e-commerce operators to analyze daily logged transaction data to predict customer purchase behavior, which may potentially lead to more effective recommendations and increased sales. Traditional recommendation techniques based on collaborative filtering (CF), although having gained success in video and music recommendation, are not sufficient to fully leverage the diverse information contained in the implicit user behavior on e-commerce platforms. In this paper, we analyze user action records in the Alibaba Mobile Recommendation dataset from the Alibaba Tianchi Data Lab, as well as the Retailrocket recommender system dataset from the Retail Rocket website. To estimate the probability that a user will purchase a certain item tomorrow, we propose a new model called Time-decayed Multifaceted Factorizing Personalized Markov Chains (Time-decayed Multifaceted-FPMC), taking into account multiple types of user historical actions not only limited to past purchases but also including various behaviors such as clicks, collects and add-to-carts. Our model also considers the time-decay effect of the influence of past actions. To learn the parameters in the proposed model, we further propose a unified framework named Bayesian Sparse Factorization Machines (BSFM). It generalizes the theory of traditional Factorization Machines to a more flexible learning structure and trains the Time-decayed Multifaceted-FPMC with the Markov Chain Monte Carlo (MCMC) method. Extensive evaluations based on multiple real-world datasets demonstrate that our proposed approaches significantly outperform various existing purchase recommendation algorithms.

CCS Concepts: • **Applied computing** → **Online shopping**; • **Computing methodologies** → *Model development and analysis*; • **Information systems** → *Data mining*.

Additional Key Words and Phrases: Online purchase prediction, recommendation systems, matrix factorization, factorizing personalized Markov chains, factorization machine, Markov chain Monte Carlo.

ACM Reference Format:
Bang Liu, Hanlin Zhang, Linglong Kong, and Di Niu. 2021. Factorizing Historical User Actions for Next-Day Purchase Prediction. *ACM Trans. Web* 1, 1, Article 1 (January 2021), 26 pages. https://doi.org/10.1145/3468227

1 INTRODUCTION

Online business-to-consumer (B2C) commerce platforms (e.g., Amazon, Alibaba, eBay) can now easily track and monitor the actions of their users, forming a large amount of implicit user behavior data. While the amount of data collected is staggering, effective approaches to extract useful information from such diverse data are still under development. In the e-commerce context, predicting

Authors' addresses: Bang Liu, University of Alberta, Electrical and Computer Engineering, Edmonton, Canada, bang3@ualberta.ca; Hanlin Zhang, University of Alberta, Electrical and Computer Engineering, Edmonton, Canada, hanlin3@ualberta.ca; Linglong Kong, University of Alberta, Mathematical and Statistical Sciences, Edmonton, Canada, lkong@ualberta.ca; Di Niu, University of Alberta, Electrical and Computer Engineering, Edmonton, Canada, dniu@ualberta.ca.

1559-1131/2021/1-ART1 $15.00
https://doi.org/10.1145/3468227

customer purchase behavior can bring about multiple benefits, such as more accurate recommendations, increased sales, higher customer acquisition rate, and enhanced competitiveness [27]. For example, one specific problem proposed in the Alibaba Mobile Recommendation Competition [1] is to predict the next-day purchases of users given the historical records of various user actions, including the *click*, *collect* (adding to favorites), *add-to-cart* and *payment*. It is highly valuable to learn from such historical implicit feedback and recommend appropriate items to interested users at the right time.

Traditional approaches for recommendation include content filtering [25], collaborative filtering (CF) [22], and matrix factorization (MF) [18], etc. However, these methods are not sufficient for next-day purchase prediction. *First,* without explicit rating scores, it is challenging to measure the similarities between users or between items just from historical actions, whereas finding user/item similarities is a key idea for both content filtering and collaborative filtering to work. *Second,* sometimes users might prefer to buy items that are different from what they already purchased. For example, a person who has already purchased an iPhone is unlikely to buy another smartphone. However, based on what users have bought, CF and MF usually tend to recommend similar items to users. *Third,* unlike explicit product ratings, the main information we can use in next-day purchase prediction contains the timestamped action sequences of each user, representing implicit feedback in nature. New approaches need to be developed to discover insights from historical actions and make recommendations based on such implicit feedback instead of explicit ratings.

In this article, we conduct an in-depth analysis of user action records in two datasets: the Alibaba Mobile Recommendation dataset [1] and the Retailrocket recommender system dataset[1]. The Alibaba Mobile Recommendation dataset is provided by the Alibaba group via Tianchi Data Lab. Alibaba is one of the largest e-commerce company in the world and is currently using big data to understand customer behavior and increase sales. The Retailrocket recommender system dataset is collected from a real-world ecommerce website, Retail Rocket, which helps web shoppers make better shopping decisions by providing personalized real-time recommendations to large amount of users and retail partners over the world. Based on our analysis to the two datasets, we propose multiple models in a unified framework to predict the next-day purchase behavior of users based on their historical action records. The main contributions of this paper are summarized as follows:

First, we analyze the Alibaba Mobile Recommendation dataset and the Retailrocket recommender system dataset. We argue that all types of historical user actions may influence their future purchases. Besides, users' interests for online shopping may vary with the day of week. Based on these observations, we propose a novel model, named *Multifaceted Factorizing Personalized Markov Chains* (Multifaceted-FPMC), to predict the probability that a user will buy a particular item on the next day. Our model takes into account multiple types of historical user actions, as well as the day-of-week information.

Second, we further observe that the influence of a user's historical actions on her future purchase decays as the time interval between the historical action and the purchase action increases. The influence decay phenomenon approximately follows a power-law distribution. We find that this phenomenon plays an important role in predicting the future behavior of users. Accordingly, we further propose our *Time-decayed Multifaceted Factorizing Personalized Markov Chains* (Time-decayed Multifaceted-FPMC) model, which incorporates the temporal decay phenomenon of the past influence into purchase prediction.

[1] https://www.kaggle.com/retailrocket/ecommerce-dataset

Finally, we propose a unified feature factorization framework, which we call Bayesian Sparse Factorization Machine (BSFM), as a solution technique to our new models, and train model parameters by Gibbs sampling. The proposed BSFM is similar to Factorization Machine [28, 29], yet only keeping sparse interactions between features and disabling undesired interactions.

We have conducted extensive evaluations based on both the Alibaba E-commerce Recommendation dataset and the Retailrocket recommender system dataset, and show that our proposed approaches significantly outperform other existing recommendation algorithms, including matrix factorization (MF) and Factorizing Personalized Markov Chains (FPMC).

The remainder of this paper is organized as follows. Sec. 2 reviews the related literatures and shows their insufficiencies in our case. We describe the Alibaba Mobile Recommendation dataset and existing approaches for online purchase prediction in Sec. 3. In Sec. 4, we conduct an in-depth analysis to the Alibaba Mobile Recommendation dataset and the Retailrocket recommender system dataset. We then present our mathematical models, including Multifaceted-FPMC and Time-decayed Multifaceted-FPMC, for next-day purchase prediction. Furthermore, we propose the Bayesian Sparse Factorization Machine (BSFM) framework as a unified form of our new models in Sec. 5, and elaborate its relationship to and differences from the original factorization machine (FM) model. We describe the Markov Chain Monte Carlo (MCMC) learning algorithm for model training in Sec. 6. In Sec. 7, we conduct extensive evaluations based on the Alibaba dataset and the Retailrocket recommender system dataset, and compare the performance of our proposed approaches with other baseline algorithms. We conclude this article in Sec. 8.

2 RELATED WORK

2.1 Traditional Recommender Systems

Online purchase prediction is related to the large amount of prior works done for recommendation systems. Content filtering [25, 40] methods create a profile for each user or item to characterize its nature and associate users with matched items. Content-based strategies are easy to express and implement. However, they require gathering external information that might be unavailable or hard to collect to create the profiles. Another problem of applying content filtering to purchase prediction is that content filtering can only recommend similar items. Different from movie or music recommendation, for online purchase, a user usually would not buy a similar item again once they already own one.

Collaborative filtering (CF) is the most popular method in recommendation systems [22, 34]. It has been widely used since 1990s and promoted the prosperity of recommendation systems [10]. Compared to content-based approaches, collaborative filtering does not require the creation of explicit profiles, but relies only on past user actions such as previous transactions or explicit item ratings. Collaborative filtering can further be divided into two primary classes: the neighborhood methods and latent factor models (matrix factorization) [13]. The neighborhood methods include user-based collaborative filtering and item-based collaborative filtering. The user-based collaborative filtering measures the similarity of two customers in various ways to identify each user's neighbors and generates recommendations based on the past behavior of a few customers who are most similar to the user, while the item-based collaborative filtering tracks user preferences by identifying similar items. Collaborative filtering has also been tailored for implicit feedback datasets in a scalable optimization procedure [14]. By optimizing the reconstruction loss with confidence variables and regularization terms, the raw observations of user behaviors on items can be factorized and utilized for recommendation.

Latent factor models [6, 16, 36, 38] try to explain the ratings by characterizing both items and users with fixed-length vectors, or latent factors, inferred from the rating patterns. Some of the most successful realizations of latent factor models are based on matrix factorization. It characterizes both items and users by vectors of factors inferred from item rating patterns. High correspondence between item and user factors leads to a recommendation. However, both neighborhood methods and latent factor models still suffer from the problem of recommending similar items. Besides, it has been mentioned in [20] that e-commerce recommender is different from movie or music recommendation systems. They should take into account the utility and utility plus of items, rather than recommending similar items based on users' historical action records.

Many works focus on tackling challenges in modeling interactions between a user, his or her previously visited item(s), and the next item to consume for better next-day purchase prediction performance. Trans4Rec[11] models such third-order relationships for large-scale sequential prediction. Specifically, Trans4Rec embeds items into a transition space where users are modeled as translation vectors operating on item sequences. GRU4Rec[12] proposes an RNN-based model for the whole session so that effective recommendations can be provided.

A variety of ranking models in recommendation systems are aiming to learn a user-specific rankings of items. The ranking scores of the items to a user are learned from the user's past interactions with the system. Bayesian Personalized Ranking (BPR) [30] is a pair-wise ranking approach for recommendations, which takes as input either implicit feedbacks or explicit ratings and outputs an ordered item list for each user. In BPR, item pairs are utilized as training data instead of a single item. By optimizing the BPR-OPT criterion through maximum posterior estimation (MAP), BPR is able to perform personalized ranking for users with factorized user-specific matrices.

2.2 Temporal Recommender Systems

In order to avoid the similar content recommendation problem and recommend more personalized items to different users, some research works have taken the sequential and temporal pattern of user historical actions into account. [5, 17, 19, 26, 31, 41] consider the temporal dynamics in collaborative filtering model to learn the dynamic characteristics of users and items. [46] investigates how to extract sequential patterns for next-state prediction, and describes a sequential recommender based on Markov chains. [23, 43] discover sequential patterns by pattern mining methods. [35] also develops a Markov-chain-based recommendation system using Markov decision processes (MDP). [31] uses personalized transition graphs to combine the benefits of sequential Markov chains with time-invariant user tastes. In particular, it proposes a model named Factorizing Personalized Markov Chains (FPMC) that combines the latent factor model and Markov chains to predict what products users will purchase on the next time. [39] proposes an opportunity model to estimate the probability that a user will buy an item at specific time interval. [42] exploits both temporal and social factors for B2B marketing campaign recommender system. [44, 45] exploit the time intervals between purchase behaviors for next-item recommendation.

Our proposed recommendation algorithms also consider the sequential and temporal patterns to predict next-purchase. Compared with previous works, we jointly model the sequential patterns with different historical actions and the time intervals. The FPMC algorithm only utilizes the order of purchased items. More information, such as the time gaps between different purchases, can be utilized to improve the temporal diversity in recommendation systems. In addition to historical purchase actions, other types of actions such as *click*, *collect* and *add-to-cart* can also be leveraged in the recommendation. Our Multifaceted-FPMC model incorporates all these kinds of information and jointly factorizes latent feature vectors for different observable features. Besides, our proposed Time-decayed Multifaceted-FPMC model further considers the temporal influence decay phenomenon of historical user actions to improve the accuracy of next-purchase prediction.

Factorization models have attracted a lot of attentions with their excellent prediction capabilities shown in several applications. Factorization machines (FM) [28] have been proposed to combine

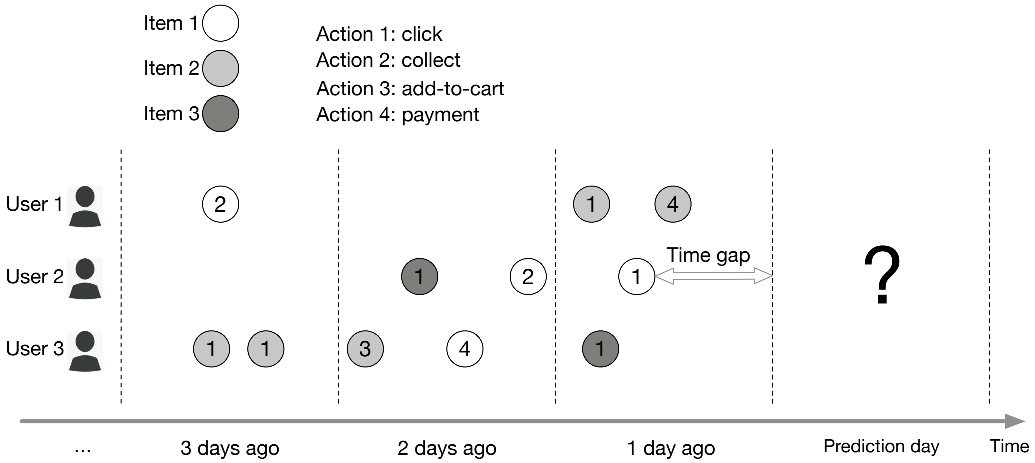


Fig. 1. Users' historical actions sequence on different items.

Table 1. The table fields of historical action records

COLUMN	DESCRIPTION
user id	Identity of users
item id	Identity of items
action type	The type of user actions, including *click, collect, add-to-cart* and *payment.*
time	The time of the action to the nearest hours.

the advantages of general machine learning classifiers, such as the Support Vector Machine (SVM), with factorization models. Factorization machines model the pairwise interactions between all features via a real-valued input feature vector. However, when considering tens of context features, the input feature vectors will be quite long and not all pairwise feature interactions are meaningful. We propose Bayesian Sparse Factorization Machine (BSFM) as a unified form to express our newly proposed models for online purchase prediction. Compared with traditional factorization machines, it enables *sparse* feature interactions through a feature interaction matrix to avoid the unnecessary interactions between irrelevant features.

3 NEXT DAY ONLINE PURCHASE PREDICTION

In this section, we first describe and analysis the user action records in the Alibaba Mobile Recommendation dataset (we briefly call it Alibaba dataset) retrieved from the Alibaba Tianchi Data Lab, as well as the Retailrocket recommender system dataset (we briefly call it Retailrocket dataset) collected from the Retail Rocket website. We then formally define our problem and present the key notations used in this paper. Finally, we briefly describe the existing Matrix Factorization (MF) approach and the existing Factorizing Personalized Markov Chains (FPMC) method for modeling user interests on items, and show their limitations in utilizing the context information contained in the historical user action data.

3.1 Data Analysis

The Alibaba dataset [1] contains the complete historical action records of 10, 000 users on 2, 876, 947 items. There are four types of actions: *click, collect, add-to-cart,* and *payment.* The time span of

Table 2. Notations used in the article

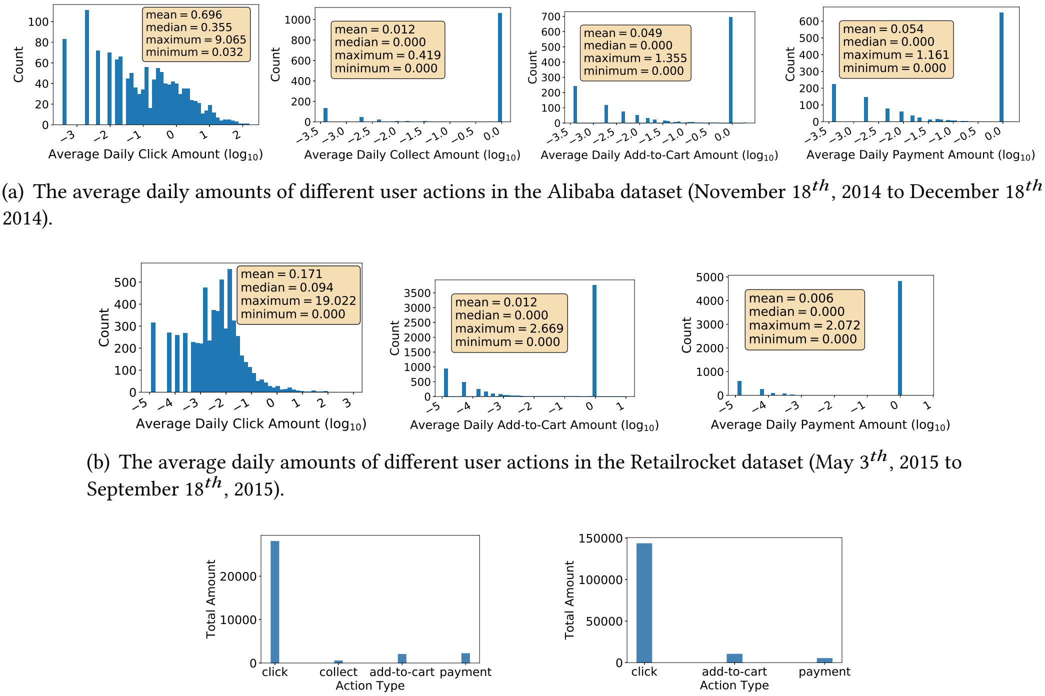
SYMBOL	DESCRIPTION
$\mathcal{U}, \mathcal{I}, \mathcal{R}$	set of users, items, historical action records
u, i, a, t, T	a user, an item, action, time, history length by day
d	day-of-week
$p(u,i)$	the probability user u will buy item i
$p(u,i\|j)$	the conditional probability user u will buy item i given that he/she already purchased item j
$p(u,i\|j,a,d)$	the conditional probability user u will buy item i on predicted day d given that he/she already performed action a on item j
v	latent feature vector
$\mathcal{B}_u$	the item set user u previously purchased
$\mathcal{B}_{u,a}$	the item set user u previously performed action a on
$C_{u,j,a}$	the total times user u performed action a on item j
$t_{u,j,a,c}$	the time interval length between the time user u perform the c-th action a on item j and the predicted day
$\mathbf{x}, y$	feature vector, target
$w_0, \mathbf{w}, V, \Phi$	model parameters of BSFM
$\mu^0, \lambda^0, \mu_\pi^w, \lambda_\pi^w, \mu_{k,\pi}^v, \lambda_{k,\pi}^v$	Regularization parameters of BSFM
$\alpha_0, \beta_0, \alpha_\lambda, \beta_\lambda, \mu_0, \gamma_0$	hyperparameters of BSFM

the historical records is from November 18^{th}, 2014 to December 18^{th}, 2014. For the Retailrocket dataset, it contains three kinds of action records: *click* (named as *view* in the dataset), *add-to-cart*, and *payment* (named as *transactions* in the dataset). The user behavior records are collected within a period of 4.5 months, from May 3th, 2015 to September 18th, 2015. There are 2,756,101 records in total, including 2,664,312 click, 69,332 add-to-cart, and 22,457 payment produced by 1,407,580 unique users.

Fig. 1 illustrates the concept of online shopping records. For each user, we record his/her various kinds of actions on different items, together with the timestamps he/she performed that action. That is, for each historical action, the *user id*, *item id*, *action type* and *time* information is recorded. Table 1 describes the different table fields for recording each historical action.

In this paper, we conduct our evaluations on a subset of each dataset to filter out the influence of web crawlers and outliers. Specifically, for the Alibaba dataset, we only keep the users who have more than 20 purchase (or *payment*) actions during November 18^{th}, 2014 to December 18^{th}, 2014 and the items that have been bought by at least one user. After filtered by this criteria, the dataset contains the historical action records of 1299 users and 1445 items. Likewise, for the Retailrocket dataset, we only retain the users and items with more than 20 records of any type. After filtering, there are 158,722 records between 6021 users and 3939 items.

Fig. 2 shows the distributions of the average daily amounts and the total amounts of different type of actions in the Alibaba dataset and the Retailrocket dataset. The text boxes in Fig. 2 shows the statistics of different types of actions (without logarithm). We can make the following observations from Fig. 2. First, as shown in the figure, user action records are highly sparse in both datasets. The average daily amount of click is 0.696 and 0.171 in the Alibaba dataset and the Retailrocket dataset, respectively. The amounts of other actions are even smaller. Second, we can see that the amount of *click* is much more than other actions. This is reasonable, as customers will browse lots of similar items before they perform any further actions on an item. Third, the amount of collect actions are



(a) The average daily amounts of different user actions in the Alibaba dataset (November 18th, 2014 to December 18th, 2014).

(b) The average daily amounts of different user actions in the Retailrocket dataset (May 3th, 2015 to September 18th, 2015).

(c) The total amounts of different user actions in the Alibaba dataset (November 18th, 2014 to December 18th, 2014).

(d) The total amounts of different user actions in the Retailrocket dataset (May 3th, 2015 to September 18th, 2015).

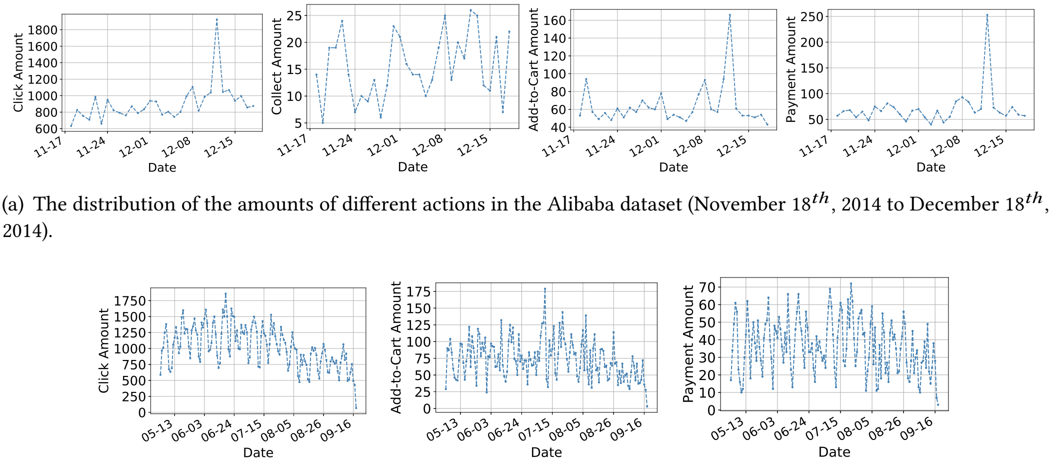
Fig. 2. The average daily amounts and the total amounts of different user actions in the Alibaba dataset and the Retailrocket dataset.

smaller than that of add-to-cart and payment in the Alibaba dataset. This makes empirical sense as users usually pay an item immediately instead of adding it as a collection if they plan to buy it.

Fig. 3 shows the distribution of the amounts of different type of actions in the Alibaba dataset and the Retailrocket dataset. For the Alibaba dataset, we can note that the amounts of different actions around December 18th, 2014 is much higher than other days. A potential explanation is that the Taobao Marketplace, under the Alibaba Group Holding Ltd, offers a variety of discounts to customers on December 12th. This is so-called Double-12 festival in China, similar to Black Friday. Therefore, the users are more active near December 12th. To normalize the bias caused by this abnormal date, a small weight can be assigned to the amounts of actions in that date [21]. The amounts of different actions over other dates are more smooth.

Fig. 4 compares the amounts of total user actions on different days of a week. As we can see, customers are more active during weekdays than during weekends. This is a common phenomenon since people usually prefer to participate in other activities or go shopping in physical stores rather than shopping online during weekends.

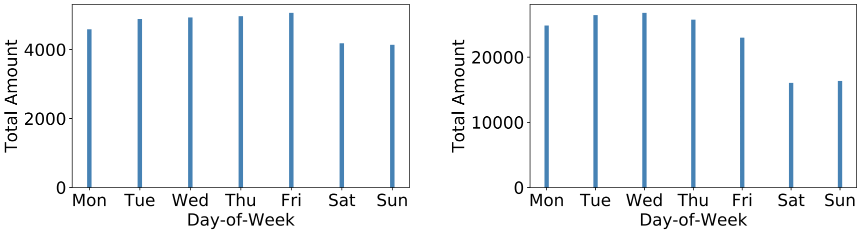
We now study how a user's previous action will affect his/her next purchase actions, taking into account the time intervals between the two actions. For each user-item pair, we extract the time interval between each purchase action of the user and his preceding action. Fig. 5 shows the



(a) The distribution of the amounts of different actions in the Alibaba dataset (November 18^{th}, 2014 to December 18^{th}, 2014).

(b) The distribution of the amounts of different actions in the Retailrocket dataset (May 3^{th}, 2015 to September 18^{th}, 2015).

Fig. 3. The distribution of the amounts of different actions in the Alibaba dataset and the Retailrocket dataset.

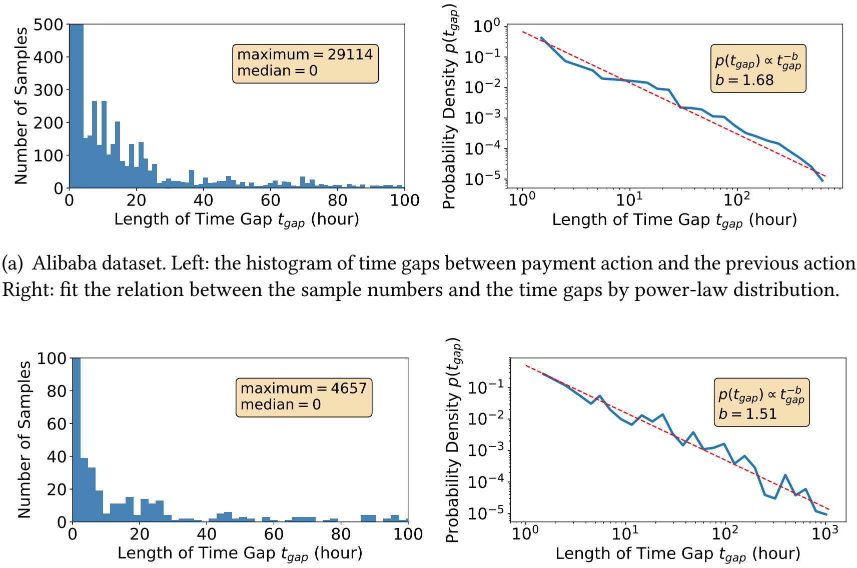


(a) Alibaba dataset.

(b) Retailrocket dataset.

Fig. 4. The amounts of all actions on different day-of-week in the Alibaba dataset (left) and the Retailrocket dataset.

histogram of the lengths of such extracted time intervals. As shown in the histogram, many of the intervals between payment actions and their preceding actions are within 1 hour, while the median interval is 0, which means that more than a half of all such intervals are less than 1 hour. From Fig. 5, we conclude that most users usually make decisions to buy an item within 1 hour after he/she performed the previous action on that item. The longer the time interval is, the less probable that the user will finally purchase the item. Therefore, the influence of previous actions on a user's purchase decision shall decay as time goes. More specifically, we use the power-law distribution [2] to model the temporal influence decay phenomenon, as shown in the right part of Fig. 5. Fig. 5 shows that the likelihood that a user will buy an item in t_{gap} hours after he/she has performed an action on it fits the power-law distribution very well; the Probability Density Function (PDF) of t_{gap} is approximately proportional to $t_{gap}^{-1.68}$ or $t_{gap}^{-1.51}$, where t_{gap} represents the time intervals between any purchase action and its previous action.



(a) Alibaba dataset. Left: the histogram of time gaps between payment action and the previous action. Right: fit the relation between the sample numbers and the time gaps by power-law distribution.

(b) Retailrocket dataset. Left: the histogram of time gaps between payment action and the previous action. Right: fit the relation between the sample numbers and the time gaps by power-law distribution.

Fig. 5. The power-law distribution of the time gaps between users' payment action and the previous action.

3.2 Problem Formalization

We now formally present our next-day-purchase prediction problem. For ease of reference, we define and list the frequently used notations in Table 2.

Definition 3.1 (Historical User Action Records). Denote a set of users by $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$ and a set of items by $\mathcal{I} = \{i_1, i_2, ..., i_{|\mathcal{I}|}\}$. A historical user action record is a tuple $(u, i, a, t) \in \mathcal{R}$, where $u \in \mathcal{U}$, $i \in \mathcal{I}$, $a \in \{1, 2, 3, 4\}$ (representing actions *click*, *collect*, *add-to-cart*, and *payment*, respectively), and t (in hours) represents the time when user u performs action a on item i. $\mathcal{R}$ is the set of all historical action records.

Given the definition of historical user action records, our problem is defined as follows:

Definition 3.2 (Problem Definition). Given a set of users $\mathcal{U}$, a set of items $\mathcal{I}$, and the historical user action records $\mathcal{R}$ between $\mathcal{U}$ and $\mathcal{I}$ during the last T days, the task is to estimate the probability $p(u, i|d, \mathcal{R})$ that a user $u \in \mathcal{U}$ will purchase an item $i \in \mathcal{I}$ during the next-day d.

3.3 From Matrix Factorization to Factorizing Personalized Markov Chains

Matrix Factorization (MF) is used widely for purchase prediction and recommendation. The traditional matrix factorization algorithm characterizes users and items by latent feature vectors inferred from user-item ratings. Fig. 6 illustrates how the matrix factorization approach works by assuming a low-rank structure for the rating matrix to be factorized. The interest of user u to item i is estimated to be proportional to the corresponding rating score. Specifically, assuming the rating of user u on item i is $r_{u,i}$, matrix factorization estimates the interest of user u on item i by

$$p(u, i) \propto r_{u,i} = \langle v_u, v_i \rangle, \tag{1}$$

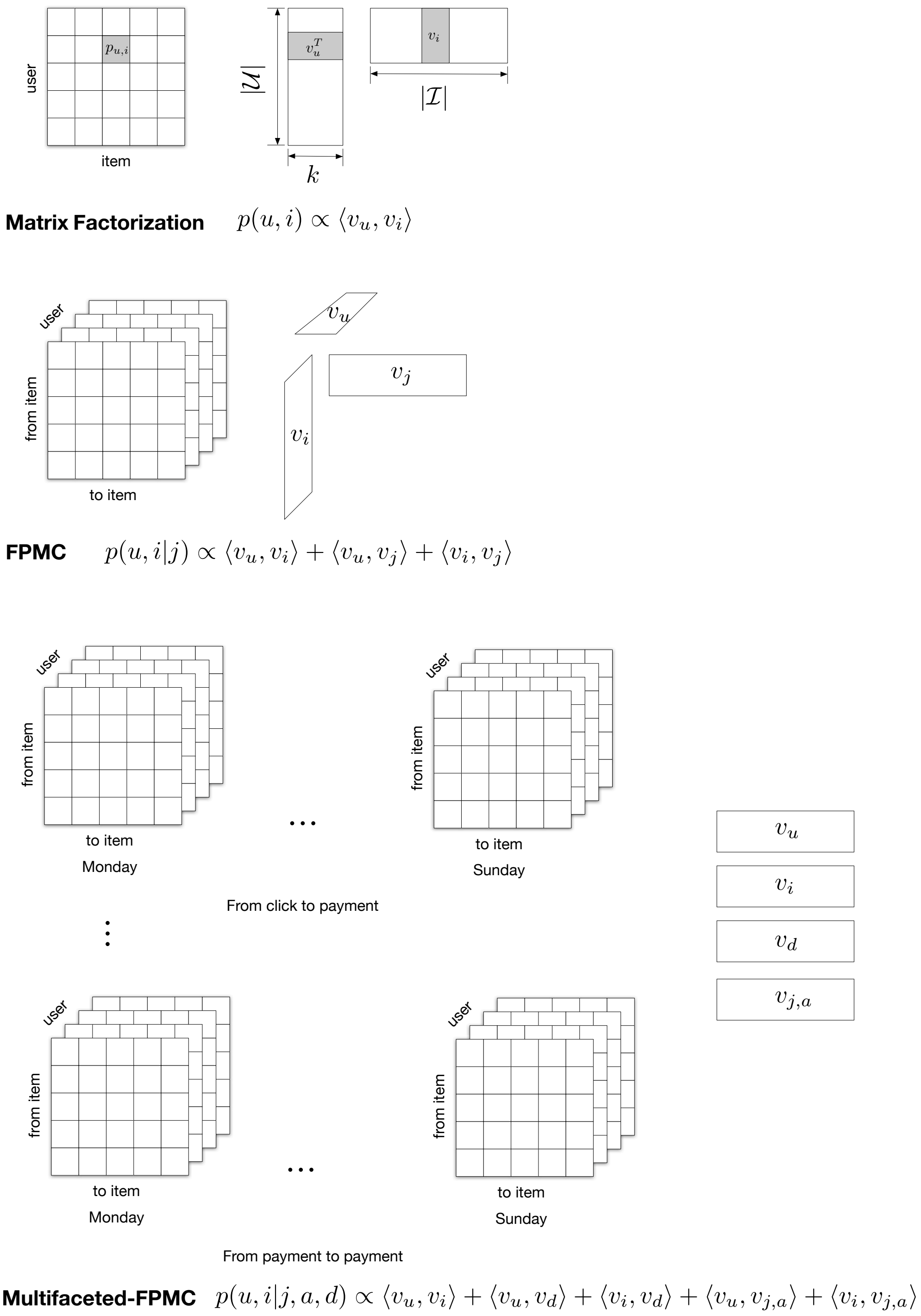

user
item
$p_{u,i}$
$|\mathcal{U}|$
v_u^T
k
v_i
$|\mathcal{I}|$
Matrix Factorization $p(u,i) \propto \langle v_u, v_i \rangle$
user
from item
to item
v_u
v_j
v_i
FPMC $p(u,i|j) \propto \langle v_u, v_i \rangle + \langle v_u, v_j \rangle + \langle v_i, v_j \rangle$
user
from item
to item
Monday
...
user
from item
to item
Sunday
From click to payment
v_u
v_i
v_d
$v_{j,a}$
user
from item
to item
Monday
...
user
from item
to item
Sunday
From payment to payment
Multifaceted-FPMC $p(u,i|j,a,d) \propto \langle v_u, v_i \rangle + \langle v_u, v_d \rangle + \langle v_i, v_d \rangle + \langle v_u, v_{j,a} \rangle + \langle v_i, v_{j,a} \rangle$

Fig. 6. Compare different models. The more information utilized, the more latent feature vectors extracted.

where $v_u \in \mathbb{R}^k$ is the latent vector that describes user u and $v_i \in \mathbb{R}^k$ is the latent vector of item i. More details about matrix factorization for recommender systems can be found in [18]. Tensor factorization models, such as Tucker Decomposition [15], are an extension for relations over several categorical variables. However, matrix/tensor factorization models are not applicable to standard prediction data that usually be described by a real-valued feature vector [28]. Besides, specialized models using factorized parameters [16, 31, 32] are usually derived individually for different specific tasks, and require effort to design specific learning algorithms.

The Factorizing Personalized Markov Chains (FPMC) model proposed in [31] further considers *the most recent* shopping basket of a user and factorizes a transition cube that contains the transition matrix of each user. Assuming C is a transition cube, then $C_{u,i,j}$ denotes the probability user u will buy item i if he/she has already bought item j. The original FPMC model utilizes a special case of Tucker Decomposition. Its form is further simplified when implemented by factorization machines [4, 29]. Assuming $\mathcal{B}_u$ is the item set of user u's last shopping basket, $p(u, i|j)$ gives the probability that user u will buy item i when he/she has already bought item j in the most recent shopping basket. Then, the FPMC model approximates this probability by

$$p(u, i|j) \propto \langle v_u, v_i \rangle + \langle v_u, v_j \rangle + \langle v_i, v_j \rangle, \tag{2}$$

where $v_u \in \mathbb{R}^k$, $v_i \in \mathbb{R}^k$ and $v_j \in \mathbb{R}^k$ are the latent feature vectors of the user, the target item for prediction, and the item the user has purchased in the most recent shopping basket. The overall probability that user u will buy item i in the next shopping basket is given by

$$p(u, i) = \sum_{j \in \mathcal{B}_u} p(u, i|j) p(j) \propto \langle v_u, v_i \rangle + \sum_{j \in \mathcal{B}_u} \frac{1}{|\mathcal{B}_u|} (\langle v_u, v_j \rangle + \langle v_i, v_j \rangle), \tag{3}$$

By jointly training these latent feature vectors, the FPMC model combines the advantages of both matrix factorization and Markov chains. Please refer to [31] for more details regarding the FPMC model.

However, given historical user action records, more context information may yet to be leveraged for next-day-purchase prediction. *First,* the FPMC model only considers the item set a user bought in his/her *last purchases.* However, rather than only considering the last purchases of a user, we can also take other types of actions into account such as click and collect. *Second,* while the FPMC model utilizes the most recent shopping basket of a user, the time gap between each purchase action and the target day for prediction is not considered. In fact, the *outdatedness* of the past actions will have an important influence on a user's future decisions and can thus be utilized to improve prediction accuracy. *Third,* when summarizing the conditional probabilities $p(u, i|j)$, the same prior probability $p(j) = \frac{1}{|\mathcal{B}_u|}$ is assigned to different items $j \in \mathcal{B}_u$ that have already been bought. However, it is clearly an over-simplified assumption that different items purchased in the past have the same impact on future purchases, without considering the quantity purchased and the time when they were purchased.

4 MODELING NEXT-DAY-PURCHASE PROBABILITY FOR E-COMMENCE

In this section, we present our newly proposed models for online purchase prediction. We argue that all different types of historical actions are informative for predicting future purchase behavior. Based on this insight, we propose our Multifaceted-FPMC model. Furthermore, we utilize the temporal decay phenomenon of the influence of historical actions based on real data, and propose the Time-decayed Multifaceted-FPMC model that takes this phenomenon into account to estimate next-day purchase probabilities.

4.1 Multifaceted Factorizing Personalized Markov Chain

Our models are motivated by the various observations from the analysis of the Alibaba dataset in Sec. 3. Apparently, if we only consider the previous payment actions, the utilized information is just a tiny portion of all the available information. By taking all kinds of historical actions into account, the information contained in the dataset will be fully utilized.

Based on the above assumption, we propose our Multifaceted-FPMC model to estimate a user's next-day purchase probability for each specific item. Denote the item set on which user u ever performed action a within the last T days as $\mathcal{B}_{u,a}$. Assume $d \in \{1, 2, ..., 7\}$ represents the day-of-week of the target day for prediction, where values $1 \sim 7$ represent Monday~Sunday respectively. The probability that user u will buy item i on the predicted day d, given that he/she performed historical action a on item j during the last T days, is given by

$$p(u, i|j, a, d) \propto \langle v_u, v_i \rangle + \langle v_u, v_d \rangle + \langle v_i, v_d \rangle + \langle v_u, v_{j,a} \rangle + \langle v_i, v_{j,a} \rangle, \tag{4}$$

where $v_d \in \mathbb{R}^k$ is the latent feature vector for different day-of-week d, and $v_{j,a}$ is the latent feature vector of item $j \in \mathcal{B}_{u,a}$.

Our Multifaceted-FPMC model estimates the probability that user u will purchase item i on the next day d by

$$\begin{aligned} p(u, i|d) &= \sum_{a=1}^{4} \sum_{j \in \mathcal{B}_{u,a}} p(u, i|j, a, d) p(j, a|d) \\ &\propto \langle v_u, v_i \rangle + \langle v_u, v_d \rangle + \langle v_i, v_d \rangle + \sum_{a=1}^{4} \sum_{j \in \mathcal{B}_{u,a}} \frac{1}{|\mathcal{B}_{u,a}|} (\langle v_u, v_{j,a} \rangle + \langle v_i, v_{j,a} \rangle), \end{aligned} \tag{5}$$

where $p(j, a|d) = \frac{1}{|\mathcal{B}_{u,a}|}$ is the prior probability for item $j \in \mathcal{B}_{u,a}$.

Our Multifaceted-FPMC model is different from the conventional FPMC model in two aspects. *First,* rather than only considering the historical *payment* actions, it also takes other three types of actions into account, which constitute a large part of utilizable information. *Second,* it also learns a latent feature vector v_d for each day-of-week to model the fact that user interests for online purchase may vary on different days of a week, as has been shown in Fig. 4. Fig. 6 shows the idea of our Multifaceted-FPMC model intuitively, in comparison with MF and FPMC. In our model, more latent feature vectors are learned so that a larger portion of historical records such as *click*, *collect* and *add-to-cart*, as well as more context information can be utilized in prediction.

4.2 Time-decayed Multifaceted FPMC

Although the model described above generalizes the FPMC model in prior literature by considering all action types and the day-of-week effect, however, both Multifaceted-FPMC and FPMC share a same weakness. That is, when predicting the next-day purchase probability, they only consider *what* actions users have performed in the past, but ignore *when* exactly these historical actions happened. In other words, if user u performed action a on both item j_1 and j_2 in the last T days, both items will have the same impact in the next-day purchase prediction, even if the action for one of the items happened more recently. Besides, the above models also ignore how many times user u has performed action a on item j. For example, if user u has clicked item $j1$ for 10 times, but has only clicked item $j2$ for one time, it is highly possible that u is more interested in item $j1$ than item $j2$. In a nutshell, in previous models, the same *priori* probabilities are assigned to different historical actions.

The temporal influence decay phenomenon of historical actions, as we have shown in Fig. 5, indicates that the influence of different historical actions shall be differentiated by the *outdatedness.*

To incorporate the temporal influence decay into our online purchase prediction model, we further revise the *priori* probability $p(j, a)$ to take into account the time and frequency of each historical action.

Suppose user u has performed action a on item j for a total number of $C_{u,j,a}$ times. The time interval between the c-th action and the predicted day d is $t_{u,j,a,c}$. We assign an *priori* probability $p(j, a|d)$ to item $j \in \mathcal{B}_{u,a}$ according to the following equation

$$p(j,a|d) \propto \frac{\sum_{c=1}^{C_{u,j,a}} t_{u,j,a,c}^{-b}}{|\mathcal{B}_{u,a}|}. \tag{6}$$

According to the power-law fitting result in Fig. 5, we set $b = 1.68$ for the Alibaba dataset and $b = 1.51$ for the Retailrocket dataset. Based on this *priori* distribution for different historical items j, we propose the Time-decayed Multifaceted Factorizing Personalized Markov Chains (Time-decayed Multifaceted-FPMC) model:

$$\begin{aligned} p(u,i|d) &= \sum_{a=1}^{4} \sum_{j \in \mathcal{B}_{u,a}} p(u,i|j,a,d)p(j,a|d) \\ &\propto \langle v_u, v_i \rangle + \langle v_u, v_d \rangle + \langle v_i, v_d \rangle + \sum_{a=1}^{4} \sum_{j \in \mathcal{B}_{u,a}} \frac{\sum_{c=1}^{C_{u,j,a}} t_{u,j,a,c}^{-b}}{|\mathcal{B}_{u,a}|} (\langle v_u, v_{j,a} \rangle + \langle v_i, v_{j,a} \rangle). \end{aligned} \tag{7}$$

5 BAYESIAN SPARSE FACTORIZATION MACHINES

In this section, we introduce a unified framework named Bayesian Sparse Factorization Machines (BSFM) which generalizes the existing Factorization Machines (FM) theory by only considering sparse interactions between latent feature vectors. We show that BSFM can be used to express both traditional models such as MF and FPMC, as well as our newly proposed models. In the next section, we will describe the MCMC inference technique to lean the BSFM model.

5.1 Bayesian Sparse Factorization Machines

Given a prediction problem, we assume it is described by a design matrix $X \in \mathbb{R}^{n \times p}$, where the i-th row $\mathbf{x}_i \in \mathbb{R}^p$ of X describes one case with p real-valued prediction variables. The prediction target of the i-th case is y_i. BSFM models the interactions between variables using factorized parameters. The model equation for a 2-order BSFM is defined as:

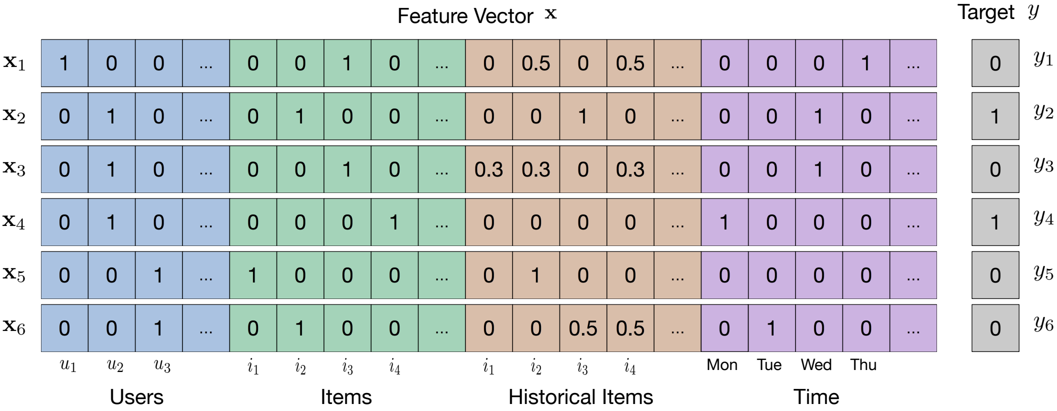
$$\hat{y}(\mathbf{x}) := w_0 + \sum_{j=1}^{p} w_j x_j + \sum_{j=1}^{p} \sum_{j'=j+1}^{p} \Phi_{j,j'} \langle \mathbf{v}_j, \mathbf{v}_{j'} \rangle x_j x_{j'}, \tag{8}$$

where $\mathbf{v}_j$ is the latent feature vector of length k for prediction variable x_j. The model parameters $\Theta = \{w_0, w_1, ..., w_p, v_{1,1}, ..., v_{p,k}\}$ are

$$w_0 \in \mathbb{R}, \ \mathbf{w} \in \mathbb{R}^p, \ V \in \mathbb{R}^{p \times k}. \tag{9}$$

Compared with existing Factorization Machines (FM), BSFM explicitly introduces a sparse matrix $\Phi \in \mathbb{R}^{p,p}$ to indicate whether x_j and $x_{j'}$ have interaction with each other. In particular, Φ is defined as:

$$\Phi_{j,j'} = \begin{cases} 1 & \text{if } x_j \text{ interacts with } x_{j'} \\ 0 & \text{if } x_j \text{ doesn't interacts with } x_{j'} \end{cases} \tag{10}$$



	Feature Vector $\mathbf{x}$																				Target y	
	Users				Items					Historical Items					Time							
	u_1	u_2	u_3		i_1	i_2	i_3	i_4		i_1	i_2	i_3	i_4		Mon	Tue	Wed	Thu				
$\mathbf{x}_1$	1	0	0	...	0	0	1	0	...	0	0.5	0	0.5	...	0	0	0	1	...		0	y_1
$\mathbf{x}_2$	0	1	0	...	0	1	0	0	...	0	0	1	0	...	0	0	1	0	...		1	y_2
$\mathbf{x}_3$	0	1	0	...	0	0	1	0	...	0.3	0.3	0	0.3	...	0	0	1	0	...		0	y_3
$\mathbf{x}_4$	0	1	0	...	0	0	0	1	...	0	0	0	0	...	1	0	0	0	...		1	y_4
$\mathbf{x}_5$	0	0	1	...	1	0	0	0	...	0	1	0	0	...	0	0	0	0	...		0	y_5
$\mathbf{x}_6$	0	0	1	...	0	1	0	0	...	0	0	0.5	0.5	...	0	1	0	0	...		0	y_6

Fig. 7. An example to show how to represent a recommender problem with real valued feature vectors $\mathbf{x}$. Each row represents a feature vector $\mathbf{x}_i$ with its corresponding target y_i. For easier interpretation, we use different colors to group feature variables into indicators for different types of information, such as user id, item id, historically purchased items and day-of-week.

The first part of BSFM is a linear regression model that contains the unary interactions of every x_j with the target. The second part contains the pairwise interactions of input variables. Compared with standard polynomial regression model, the key difference is that the interaction of x_j and $x_{j'}$ is not modeled by an independent parameter $w_{j,j'}$ but with a factorized parameterization $w_{j,j'} \approx \langle \mathbf{v_j}, \mathbf{v_{j'}} \rangle = \sum_{f=1}^{k} v_{j,f} v_{j',f}$ based on the assumption that the effect of pairwise interactions has a low rank [29]. Compared with traditional factorization machines, the key distinction of BSFM is that it introduces a matrix Φ to control sparse interactions between features, avoiding unnecessary interactions, whereas the existing FM model can only express full interactions between all pairs of features. It is easy to see that the BSFM will degenerate into the FM if we set all the elements of Φ to be 1.

We show that both our newly proposed models and several existing models can be represented in the form of BSFM by defining appropriate feature vector $\mathbf{x}$ and interaction matrix Φ for each model. For different models, the corresponding feature vector $\mathbf{x}$ is defined as follows:

- **MF**: we can exactly approximate the matrix factorization (MF) algorithm by defining the feature vector $\mathbf{x}$ using two categorical variables $\mathbf{x}_u \in \mathbb{R}^{|\mathcal{U}|}$ and $\mathbf{x}_i \in \mathbb{R}^{|\mathcal{I}|}$, that is,

$$\mathbf{x}_u = (\underbrace{0, ..., 0, 1, 0, ..., 0}_{|\mathcal{U}|}), \tag{11}$$

$$\mathbf{x}_i = (\underbrace{0, ..., 0, 1, 0, ..., 0}_{|\mathcal{I}|}), \tag{12}$$

where each variable in $\mathbf{x}_u$ denotes a user, and each variable in $\mathbf{x}_i$ denotes an item. For a user-item pair (u, i), the u-th entry in $\mathbf{x}_u$ is 1, and similarly the i-th entry in $\mathbf{x}_i$ is 1, and the rest is 0 (e.g., see the first two groups of Fig. 7). Using a feature vector $\mathbf{x} \in \mathbb{R}^{|\mathcal{U}|+|\mathcal{I}|}$ with binary indicator variables as the input of BSFM,

$$(u, i) \longrightarrow \mathbf{x} = (\mathbf{x}_u, \mathbf{x}_i), \tag{13}$$

the BSFM will be exactly the same as a biased matrix factorization model [24, 37]:

$$\hat{y}(\mathbf{x}) = \hat{y}(u,i) = w_0 + w_u + w_i + \sum_{f=1}^{k} v_{u,f} v_{i,f}. \tag{14}$$

where w_0 is the global bias. w_u, w_i refer to three categorical variables for user and item, respectively [28].

- **FPMC**: the FPMC algorithm incorporates the historical purchased item set, factorizing an MF model and Markov chain model jointly. BSFM can also mimic the FPMC algorithm by appending a third part, $\mathbf{x}_4$ (with 4 representing the action *payment*), to the feature vector $\mathbf{x}$ [29]. $\mathbf{x}_4$ is a set-categorical variable to represent the items that have been purchased by a user in the past T days, i.e.,

$$\mathbf{x}_4 = (\underbrace{0, ..., 1/|\mathcal{B}_u|, 0, ..., 1/|\mathcal{B}_u|, 0, ..., 0}_{|\mathcal{I}|,\ \text{historically purchased item set}}) \tag{15}$$

where the $|\mathcal{B}_u|$ non-zero elements in it represent the items purchased by user u. The feature vector representation $\mathbf{x} \in \mathbb{R}^{|\mathcal{U}|+2|\mathcal{I}|}$ will be

$$(u,i) \longrightarrow \mathbf{x} = (\mathbf{x}_u, \mathbf{x}_i, \mathbf{x}_4). \tag{16}$$

- **Multifaceted-FPMC**: we can incorporate more context information by adding more feature sections into $\mathbf{x}$. Similar to $\mathbf{x}_4$ in FPMC, we add three more set-categorical variables, $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ that represent the item sets a user has clicked, collected or added-to-cart in the past T days, respectively. Besides, considering that user interests for online shopping may change as day-of-week varies, we further add a categorical variable $\mathbf{x}_d$ to indicate which day-of-week the predicted day is, i.e.,

$$\mathbf{x}_d = (\underbrace{0, ...0, 1, 0, ..., 0}_{7,\ \text{day-of-week}}). \tag{17}$$

In this case, the feature vector $\mathbf{x} \in \mathbb{R}^{|\mathcal{U}|+5|\mathcal{I}|+7}$ will be

$$(u,i) \longrightarrow \mathbf{x} = (\mathbf{x}_u, \mathbf{x}_i, \mathbf{x}_d, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4). \tag{18}$$

- **Time-decayed Multifaceted-FPMC**: the Time-decayed Multifaceted-FPMC model can be represented in the form of BSFM similar to the Multifaceted-FPMC model. Assume that the set-categorical variable x_a^t ($a = 1, 2, 3, 4$) represents the items that user u has performed the action a on in the past T days, i.e.,

$$\mathbf{x}_a^t = (\underbrace{0, ..., \sum_{c=1}^{C_{u,j1,a}} t_{u,j1,a,c}^{-b}/|\mathcal{B}_{u,a}|, 0, ..., \sum_{c=1}^{C_{u,j2,a}} t_{u,j2,a,c}^{-b}/|\mathcal{B}_{u,a}|, 0, ..., 0}_{|\mathcal{I}|,\ \text{historically purchased item set}}). \tag{19}$$

In this case, the feature vector $\mathbf{x} \in \mathbb{R}^{|\mathcal{U}|+5|\mathcal{I}|+7}$ for the Time-decayed Multifaceted-FPMC model will be

$$(u,i) \longrightarrow \mathbf{x} = (\mathbf{x}_u, \mathbf{x}_i, \mathbf{x}_d, \mathbf{x}_1^t, \mathbf{x}_2^t, \mathbf{x}_3^t, \mathbf{x}_4^t). \tag{20}$$

By defining the feature vectors as above, we are able to represent our models in the general form of BSFM, with the corresponding interaction matrix Φ set according to (10).

6 MARKOV CHAIN MONTE CARLO INFERENCE OF MODEL PARAMETERS

In this section, we introduce the Markov chain Monte Carlo (MCMC) [3, 8] inference for learning model parameters in the form of BSFM. Compared with algorithms such as stochastic gradient descent (SGD) or alternative least square (ALS), the MCMC algorithm is able to do automatic hyperparameter learning or no learning hyperparameter. MCMC usually gives better accuracy with structured Bayesian inference [8]. Here we present 2-order BSFM without loss of generality.

6.1 Model Structure

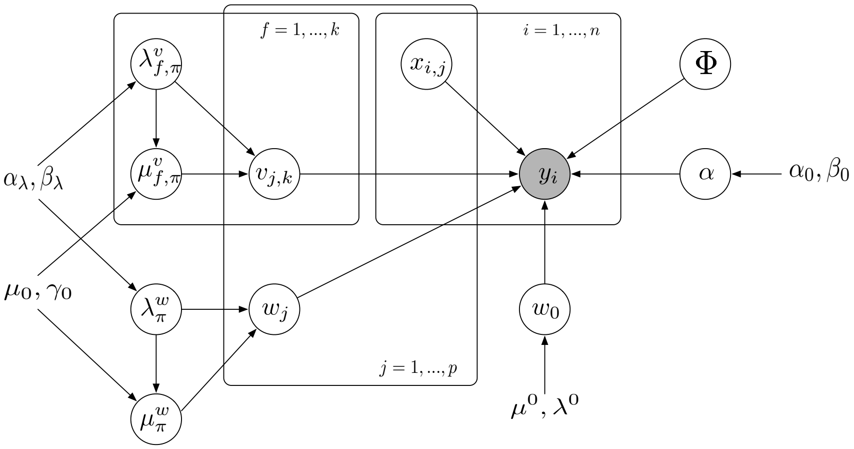


Fig. 8. Graphical representation of BSFM.

Fig. 8 depicts the graphical representation of our BSFM framework. According to the functionalities of different parameters, we divide them into three categories.

- Model parameters $\Theta := \{w_0, \mathbf{w}, V, \Phi\}$. The interaction matrix Φ is predefined according to the feature interactions in a specific model. Other parameters will be sampled by MCMC inference using Gibbs sampling.
- Regularization parameters $\Theta_H := \{\mu^0, \lambda^0, \mu^w_\pi, \lambda^w_\pi, \mu^v_{f,\pi}, \lambda^v_{f,\pi}\}$. These parameters regularize the model parameters to prevent overfitting.
- Hyperparameters $\Theta_0 := \{\alpha_0, \beta_0, \alpha_\lambda, \beta_\lambda, \mu_0, \gamma_0\}$. These parameters are introduced in BSFM so that the regularization values Θ_H can be automatically determined, which is a major advantage of MCMC. The number of hyperpriors $|\Theta_0|$ is smaller than the number of regularization parameters $|\Theta_H|$. More importantly, MCMC is typically insensitive to the choice of Θ_0 [29].

6.2 Inference: Efficient Gibbs Sampling

We use Gibbs sampling to draw from the posterior of BSFM since posterior inference is analytically intractable. Standard Gibbs sampling divides all inferred variables Θ and Θ_H into disjoint blocks and every block contains a subset of the parameters. However, it will leads to high time complexity in the final Gibbs sampler. Therefore, we use single parameter Gibbs sampling instead of standard block Gibbs sampling [29]. We exploit the multi-linear nature of BSFM for notational readability.

Definition 6.1 (Multi-linear Nature of BSFM). For each model parameter $\theta \in \Theta$, the BSFM is a linear combination of two functions g_θ and h_θ that are independent of the value θ. Therefore, (8) can be rewritten as [28]:

$$\hat{y}(\mathbf{x}|\Theta) := g_\theta(\mathbf{x}) + \theta h_\theta(\mathbf{x}), \forall\theta \in \Theta \tag{21}$$

where

$$h_\theta(\mathbf{x}) = \frac{\partial \hat{y}(\mathbf{x}|\Theta)}{\partial\theta} = \begin{cases} 1 & \text{if } \theta \text{ is } w_0 \\ x_l & \text{if } \theta \text{ is } w_l \\ x_l \sum_{j\neq l} \Phi_{l,j} v_{j,f} x_j & \text{if } \theta \text{ is } v_{j,f} \end{cases} \tag{22}$$

The value of g_θ will be computed by $g_\theta(\mathbf{x}) = \hat{y}(\mathbf{x}|\Theta) - \theta h_\theta(\mathbf{x})$ instead of computing directly, therefore its definition is omitted here.

We now describe the MCMC inference for BSFM. MCMC samples from the posterior distributions of parameters rather than learn an optimal value for each of them. For hyperparameters, as MCMC is typically insensitive to the choice of Θ_0, we introduce priors for the hyperparameters and choose $\alpha_0 = \beta_0 = \alpha_\lambda = \beta_\lambda = \gamma_0 = 1$, which allows model complexity to be controlled automatically basd on the training data.

For regularization parameters, MCMC places distributions on the priors for integration of Θ_H. By integrating regularization parameters into the model, it avoids a time-consuming search for these parameters. Specifically, for each pair $(\mu_\theta, \lambda_\theta) \in \Theta_H$ of prior parameters, we assume a Gamma distribution for each prior precision λ_θ and α except λ^0, and a normal distribution for each mean μ_θ of all model parameters $\theta \in \Theta$ but μ^0:

$$\lambda_\pi^w \sim \Gamma(\alpha_\lambda, \beta_\lambda),\ \mu_\pi^w \sim \mathcal{N}(\mu_0, \gamma_0\lambda_\pi^w), \tag{23}$$

$$\lambda_{f,\pi}^v \sim \Gamma(\alpha_\lambda, \beta_\lambda),\ \mu_{f,\pi}^v \sim \mathcal{N}(\mu_0, \gamma_0\lambda_{f,\pi}^v), \tag{24}$$

$$\alpha \sim \Gamma(\alpha_0, \beta_0). \tag{25}$$

Given n observed samples $(y_i, \mathbf{x}_i) \in \mathbb{R}^{p+1}$, the corresponding conditional posterior distributions for Θ_H are [29]:

$$\alpha|y, X, \Theta_0, \Theta \sim \Gamma\left(\frac{\alpha_0 + n}{2}, \frac{1}{2}\left[\sum_{i=1}^{n}\left(y_i - \hat{y}\left(\mathbf{x}_i|\Theta\right)\right)^2 + \beta_0\right]\right), \tag{26}$$

$$\lambda_\pi|\Theta_0, \Theta_H \setminus \{\lambda_\pi\}, \Theta \sim \Gamma\left(\frac{\alpha_\lambda + p^\pi + 1}{2}, \frac{1}{2}\left[\sum_{j=1}^{p}\delta\left(\pi(j) = \pi\right)\left(\theta_j - \mu_\theta\right)^2 + \gamma_0\left(\mu_\pi - \mu_0\right)^2 + \beta_\lambda\right]\right), \tag{27}$$

$$\mu_\pi|\Theta_0, \Theta_H \setminus \{\lambda_\pi\}, \Theta \sim \mathcal{N}\left(\frac{1}{p^\pi + \gamma_0}\left[\sum_{j=1}^{p}\delta\left(\pi(j) = \pi\right)\theta_j + \gamma_0\mu_0\right], \frac{1}{\left(p^\pi + \gamma_0\right)\lambda_\pi}\right), \tag{28}$$

where

$$p^\pi := \sum_{j=1}^{p}\delta\left(\pi(j) = \pi\right), \tag{29}$$

and δ is the indicator function

$$\delta(b) := \begin{cases} 1 & \text{if } b \text{ is true} \\ 0 & \text{if } b \text{ is false} \end{cases} \tag{30}$$

For model parameters, we assume normal distribution. With n observed samples $(y_i, \mathbf{x}_i)$, the corresponding conditional posterior distributions for Θ satisfy:

$$p(\Theta|\mathbf{y}, X, \Theta_H) \propto \prod_{i=1}^{n}\sqrt{\alpha}e^{-\frac{\alpha}{2}(y_i - y(\mathbf{x}_i,\Theta))^2}\prod_{\theta\in\Theta}\sqrt{\lambda_\theta}e^{-\frac{\lambda_\theta}{2}(\theta-\mu_\theta)^2}. \tag{31}$$

Using the multi-linear representation form of BSFM, we can infer that the conditional posterior distribution for each model parameter $\theta \in \Theta$ is:

$$\theta | X, y, \Theta \setminus \{\theta\}, \Theta_H \sim \mathcal{N}(\tilde{\mu}_\theta, \tilde{\sigma}^2_\theta), \tag{32}$$

where

$$\tilde{\sigma}^2_\theta := \left(\alpha \sum_{i=1}^{n} h^2_\theta (\mathbf{x}_i) + \lambda_\theta \right)^{-1}, \tag{33}$$

$$\tilde{\mu}_\theta := \tilde{\sigma}^2_\theta \left(\alpha \theta \sum_{i=1}^{n} h^2_\theta (\mathbf{x}_i) + \alpha \sum_{i=1}^{n} h_\theta (\mathbf{x}_i)\, e_i + \mu_\theta \lambda_\theta \right), \tag{34}$$

e_i is the prediction error of the i-th sample:

$$e_i := y_i - \hat{y}(\mathbf{x}_i | \Theta). \tag{35}$$

6.3 Learning Procedures

Algorithm 1 depicts the learning procedures of Gibbs sampling for BSFM. First, we initialize the model parameters to be zero or random values. For each sampling iteration, we sample the regularization parameters and model parameters in sequence. Before sampling the next parameter, the depending variables and parameters must be updated using the sampled new parameters.

We need to make two changes for binary classification task. First, after we get the probabilities, we need to map the normal distributed $\hat{y}$ to a probability $P(\hat{y}) \in [0, 1]$ that defines the Bernoulli distribution for binary classification [9]. Here we use the CDF function of a normal distribution for mapping:

$$P(\hat{y}) := \Phi(\hat{y}). \tag{36}$$

Second, in algorithm 1, instead of regressing to y, we sample it in each iteration from its posterior that has a truncated normal distribution

$$y'_i | \mathbf{x}_i, y_i, \Theta \sim \begin{cases} \mathcal{N}(\hat{y}(\mathbf{x}_i, \Theta), 1)\delta(y'_i < 0) & \text{if } y_i \text{ belongs to negative class} \\ \mathcal{N}(\hat{y}(\mathbf{x}_i, \Theta), 1)\delta(y'_i \geqslant 0) & \text{if } y_i \text{ belongs to positive class} \end{cases}. \tag{37}$$

Sampling from this distribution is efficient [33].

7 EXPERIMENTS

In this section, we compare our approaches with multiple state-of-the-art algorithms and show the benefits of incorporating various context information and the temporal influence decay phenomenon of historical actions.

7.1 Experimental Setup and Metrics

Our performance evaluation is conducted on the subset of the Alibaba and Retailrocket e-commerce Recommendation dataset, as described in Sec. 3. Our objective is to predict the user-item pairs that will have purchase actions on a prediction date based on previous action records, which is a binary classification problem. We have run experiments on different prediction dates with different historical record lengths (number of days) and observed similar results. Thus, without loss of generality, our model takes as input the records of first T days in each dataset as training data, and predict the purchase decisions in the next day. For example, the time span of the historical records is from November 18th, 2014 to December 18th, 2014 in the Alibaba dataset. For a historical length of $T = 7$, we take the records from November 18th, 2014 to November 25th, 2014 as our training data, and evaluate the performance of different models on November 26th, 2014. Similarly

ALGORITHM 1: Markov Chain Monte Carlo Inference (MCMC) for BSFM

Input: Training data S_{train}, test data S_{test}, initialization σ.
Output: Prediction $\hat{y}_{test}$ for test cases.

Initialization Step:

```
$w_0 \leftarrow 0$; $\mathbf{w} \leftarrow (0, ..., 0)$; $\mathbf{V} \sim \mathcal{N}(0, \sigma)$;
$\#_{samples} \leftarrow 0$;
```

Gibbs Sampling Step:

```
repeat
    $\hat{\mathbf{y}} \leftarrow$ predict all cases $S_{train}$;
    $\mathbf{e} \leftarrow \mathbf{y} - \hat{\mathbf{y}}$;
    //update the regularization parameters
    sample $\alpha$ using (26);
    for $(\mu_\pi, \lambda_\pi) \in \Theta_H$ do
        sample $\lambda_\pi$ using (27);
        sample $\mu_\pi$ using (28);
    end for
    //update the model parameters
    sample $w_0$ from $\mathcal{N}(\tilde{\mu}_{w_0}, \tilde{\sigma}^2_{w_0})$ (32);
    for $l \in \{1, ..., p\}$ do
        sample $w_l$ from $\mathcal{N}(\tilde{\mu}_{w_l}, \tilde{\sigma}^2_{w_l})$ (32);
        update $\mathbf{e}$;
    end for
    for $f \in \{1, ..., k\}$ do
        for $l \in \{1, ..., p\}$ do
            sample $v_{l,f}$ from $\mathcal{N}(\tilde{\mu}_{v_{l,f}}, \tilde{\sigma}^2_{v_{l,f}})$ (32);
            update $\mathbf{e}$;
        end for
    end for
    $\#_{samples} \leftarrow \#_{samples} + 1$;
    $\hat{\mathbf{y}}^*_{test} \leftarrow$ predict all cases $\mathrm{S}_{test}$ (37);
    $\hat{\mathbf{y}}_{test} \leftarrow \hat{\mathbf{y}}_{test} + \hat{\mathbf{y}}^*_{test}$;
until stopping criterion is met
$\hat{\mathbf{y}}_{test} \leftarrow \frac{1}{\#_{samples}} \hat{\mathbf{y}}_{test}$;
```

for $T \in \{1, 7, 14, 28\}$. We report our experimental results on the two datasets to compare the performance of different methods.

To evaluate the performance of different approaches, we are interested in finding out how many user-item pairs that have purchase actions on the predicted day can be correctly predicted by different methods. Denote N_{TP} as the number of correctly predicted user-item purchase pairs, N_{FP} as the number of incorrectly predicted user-item implicit action pairs, N_{TN} as the number of correctly predicted negative user-item purchase actions, and N_{FN} the number of incorrectly predicted negative user-item purchase actions. We compute the following metrics for evaluation:

- *Precision*: the ratio of correctly predicted purchase actions to total $N_{TP} + N_{FP}$ pairs predicted to have purchase actions on the predicted day.

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}}, \tag{38}$$

- *Recall*: the ratio of correctly predicted purchase actions to the $N_{TP} + N_{FN}$ pairs that really have purchase actions on the predicted day.

$$Recall = \frac{N_{TP}}{N_{TP} + N_{FN}}, \tag{39}$$

- *Precision-Recall curve (PR curve)*: The Precision-Recall curve shows the trade-off between precision and recall for different thresholds. Compared with ROC curve, Precision-Recall curve is preferable in highly skewed datasets such as e-commerce purchase datasets where the number of negative instances is far more than positive instances since it gives more informative reflections about the performance of a classifier [7].
- *Receiver Operating Characteristic curve (ROC curve)*: the ROC curve is created by plotting the Recall against the false positive rate (FPR) at various threshold settings, where FPR is defined as:

$$FPR = \frac{N_{FP}}{N_{FP} + N_{TN}}. \tag{40}$$

- *Area Under the Curve (AUC)*: the area under an ROC curve, which is insensitive to sample imbalance. The AUC is larger AUC when the performance of a classifier is better.

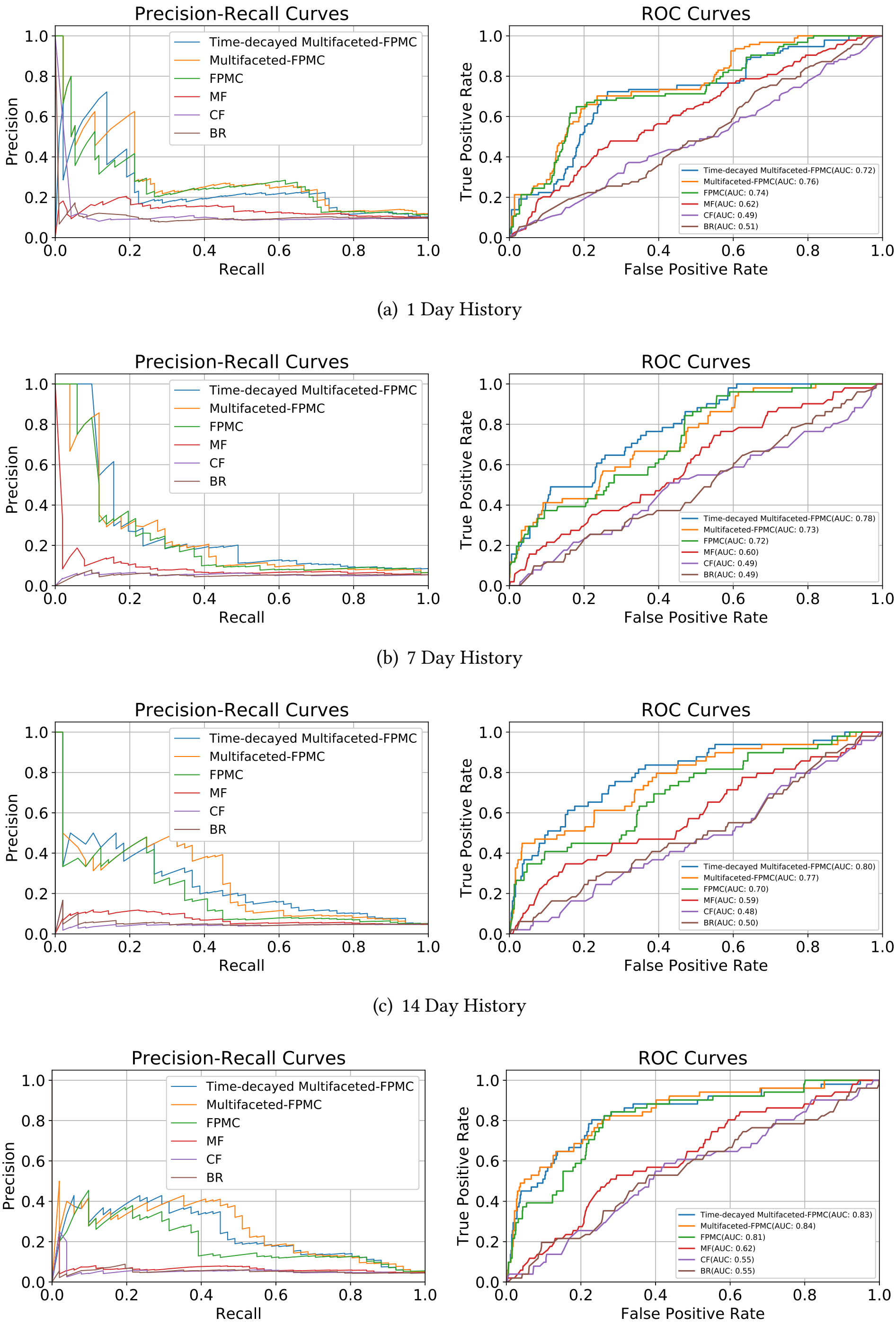
7.2 Evaluated Approaches

We compare our approaches with multiple existing algorithms. By incorporating different features into the feature vector that describes a user-item pair, we are able to utilize different context information and factorize multifaceted latent factors for prediction and recommendation. Specifically, we compare the performance of the following methods: MF, FPMC, Multifaceted-FPMC and Time-decayed Multifaceted-FPMC that described in Sec. 5, as well as Bayesian Personalized Ranking (BPR) [30] and Collaborative Filtering (CF) for implicit feedback datasets [14]. Considering historical information in our datasets, we also compare our model with two strong baselines: GRU4Rec[12] and TransRec[11].

We compare the performance of different approaches over the Alibaba and Retailrocket data subsets described in Sec. 3.1. For hyper-parameters, we use the the history records of length {1, 7, 14, 28} for training, the next day of historical ends for validation (for choosing hyper-parameters such as lengths of latent vectors) and the next following day of history records for testing in the Alibaba dataset and the Retailrocket dataset. As MCMC is typically insensitive to the choice of Θ_0 due to the huge amount of explanatory variables [8], we introduce priors for the hyper-parameters and initially set $\alpha_0 = \beta_0 = \alpha_\lambda = \beta_\lambda = \gamma_0 = 1$. We set the time span of historical records to be $T \in \{1\ day, 7\ days, 14\ days, 28\ days\}$, as reported in Fig. 9 and Fig. 10. For ranking-based models such as BPR, we specify a list of items with length N $\in \{10, 20, \ldots, 100\}$, regularization factor $\in \{0.001, 0.1, 10\}$ and use grid search to fine-tune the models. In particular, for every user-item feature vector in the records, we generate an item list with length N for that user, if this item is in the list, the model will predict that the user will purchase this item.

7.3 Performance Analysis

Fig. 9 and Fig. 10 demonstrates the effectiveness of our new proposed models. The Multifaceted-FPMC model outperforms the FPMC model and MF model significantly in most cases, which proves the importance of taking all types of historical actions into account. When history length is not long (such as 1 day or 7 days), incorporating actions other than *payment* is quite necessary. The reason is that it is highly possible that users may have only a few purchase actions or even no purchase action during the last few days. However, the times of other actions such as *click* are usually much



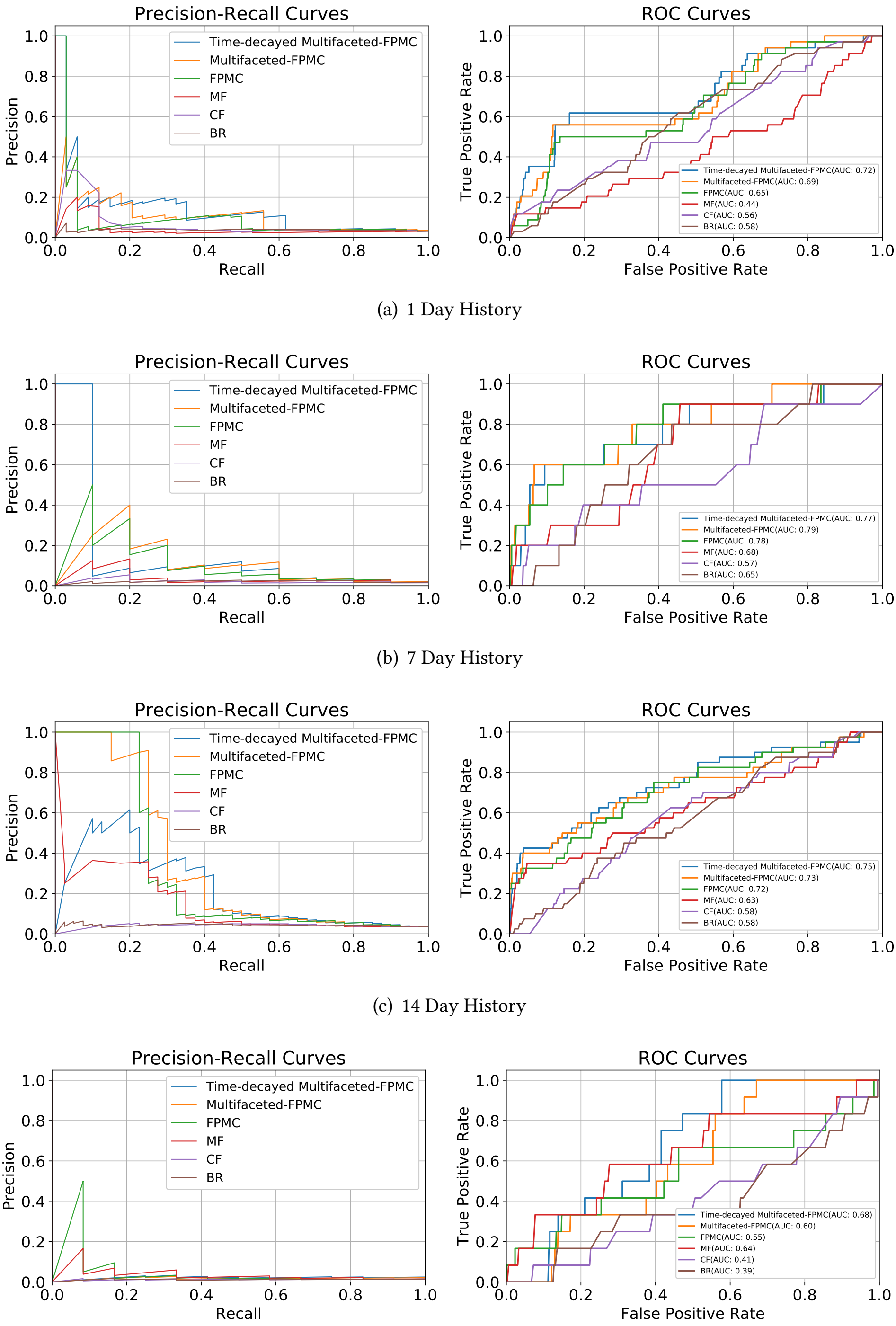
(a) 1 Day History

(b) 7 Day History

(c) 14 Day History

(d) 28 Day History

Fig. 9. Compare different algorithms with various history lengths based on the Alibaba dataset.



(a) 1 Day History

(b) 7 Day History

(c) 14 Day History

(d) 28 Day History

Fig. 10. Compare different algorithms with various history lengths based on the Retailrocket dataset.

more than *payment* action. In this case, the Multifaceted-FPMC model achieves benefits by utilizing the information of other actions for future purchase prediction.

Compared with other approaches, the performance of Time-decayed Multifaceted-FPMC keeps being the best under different experiment settings. As we can see, after taking the time intervals of historical actions into account and model the temporal influence decay phenomenon by power-law distribution, the Time-decayed Multifaceted-FPMC model further improves the prediction AUC and outperforms all other approaches. This demonstrates the key role of time intervals for purchase prediction, where other approaches such as MF, FPMC, and Multifaceted-FPMC fail to take into account.

BPR generally performs worst under a sparse and long time span setting. BPR fails to utilize the supervised labels for training. Instead, it learns in an unsupervised manner by incorporating all kinds of user actions. The primary objective of BPR is providing a user with a ranked list of items of interest based on the BPR-OPT criterion, which is a differentiable maximum posterior estimator composing the sum of logarithmic predicted preferences and the regularization term [30]. The BPR-OPT criterion is preferable for a ranking task where we are classifying the difference of two predictions ($\hat{x}_{ui} - \hat{x}_{uj}$) between user u and two items i and j to create user specific pairwise preferences using rank-based metrics. However, for purchase prediction, our task is not to predict a list of items that the target user is most likely to access in his next visit. Instead, we treat the next-item recommendation task as a binary classification problem, where we takes as input the feature vector of every record and output whether the corresponding user would purchase the item or not. In this case, BPR and CF yield bad results as they treat the items that a user has interacted with homogeneously. We can also see that most of the methods' performance improves with the growth of the history length. However, both the performance of CF and BPR remain constant in the Alibaba dataset. This clearly demonstrates the effectiveness of exploiting sequential strategies, i.e., users' historical sequential preferences, to improve the recommendation performances.

In our additional comparison with two strong baselines incorporating the whole historical records, our model performs better than Trans4Rec[11] and GRU4Rec[12] in Precision@N, where $N \in \{300, 600, 1000\}$ as shown in the table 3. We perform a paired t-test at significance level 10% to show that our model is comparable to Trans4Rec and GRU4Rec. Although Trans4Rec and GRU4Rec both capture personalized sequential behavior and model third-order interactions between users, their previously consumed item, and their next item, they fail at incorporating time-decaying effects of user historical behaviors, which leads to inferior performance as demonstrated in our experiments.

In our experiments, we consistently compare the performance of proposed models and baselines on the two datasets with different lengths of history. It is shown that choosing a history length of 7 days or 14 days performs the best in general. It makes empirical sense since people are more likely to buy products they recently browsed. The performance decreases significantly with 4-week's history length, which proves that choosing an appropriate length of history is of great importance

Table 3. Performance comparison with Trans4Rec[11] and GRU4Rec[12]

	Tianchi			Retailrocket		
Precision@N	300	600	1000	300	600	1000
Ours	0.025	0.023	0.018	0.035	0.032	0.031
Trans4Rec	0.021	0.020	0.015	0.027	0.026	0.023
GRU4Rec	0.023	0.021	0.017	0.029	0.025	0.023

to the performance of different approaches. Long historical records are usually too noisy and redundant for the next-day purchase prediction task.

8 CONCLUSIONS

To summarize, traditional approaches for recommendation and future purchase prediction, such as Matrix Factorization and Factorizing Personalized Markov Chains, are insufficient to fully utilize the various context information contained in users' historical records data. In this article, based on the two historical user action records datasets from Alibaba group and Retail Rocket website, we investigate the characteristics of real users' actions and get some insights. First, we show that different types of actions user performed previously are informative and helpful for users' future purchase prediction. Based on this discovery, we propose our Multifaceted-FPMC model that utilizes all different kinds of actions. Second, we further observe that users' historical actions' influence on their future purchase actions decays with the time intervals between historical actions and purchase actions. The decay speed is approximately following a power law distribution. Based on this temporal influence decay phenomenon, we further propose our Time-decayed Multifaceted-FPMC model for future purchase probability estimation. Finally, we show that our models can be represented in a unified manner and propose the Bayesian Sparse Factorization Machines framework. Extensive evaluations show that the proposed models can achieve better performance than previous approaches such as Matrix Factorization and Factorizing Personalized Markov Chains.

ACKNOWLEDGMENTS

This project was supported by NSERC Discovery Grant RGPIN-2019-04998. The authors would like to thank Dr. De-Chuan Zhan for his helpful advice and comments on various technical issues investigated in the paper.

REFERENCES

[1] Alibaba. 2015. Ali-Mobile-Rec. http://tianchi.aliyun.com/datalab/dataSet.htm?id=4.

[2] Jeff Alstott, Ed Bullmore, and Dietmar Plenz. 2014. powerlaw: a Python package for analysis of heavy-tailed distributions. (2014).

[3] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. 2003. An introduction to MCMC for machine learning. *Machine learning* 50, 1-2 (2003), 5–43.

[4] Immanuel Bayer and Steffen Rendle. 2013. Factor models for recommending given names. *ECML PKDD Discovery Challenge* (2013), 81.

[5] Wei Chen, Wynne Hsu, and Mong Li Lee. 2013. Modeling user's receptiveness over time for recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval.* ACM, 373–382.

[6] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems.* ACM, 39–46.

[7] Jesse Davis and Mark Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning.* ACM, 233–240.

[8] Christoph Freudenthaler, Lars Schmidt-Thieme, and Steffen Rendle. 2011. Bayesian factorization machines. (2011).

[9] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. 2014. *Bayesian data analysis.* Vol. 2. Taylor & Francis.

[10] Wanrong Gu, Shoubin Dong, and Zhizhao Zeng. 2014. Increasing recommended effectiveness with markov chains and purchase intervals. *Neural Computing and Applications* 25, 5 (2014), 1153–1162.

[11] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems.* 161–169.

[12] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[13] Thomas Hofmann. 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 89–115.

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 263–272.

[15] Yong-Deok Kim and Seungjin Choi. 2007. Nonnegative tucker decomposition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

[16] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.

[17] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53, 4 (2010), 89–97.

[18] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[19] Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. 2010. Temporal diversity in recommender systems. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 210–217.

[20] Beibei Li, Anindya Ghose, and Panagiotis G Ipeirotis. 2011. Towards a theory model for product search. In *Proceedings of the 20th international conference on World wide web*. ACM, 327–336.

[21] Qiang Li, Maojie Gu, Keren Zhou, and Xiaoming Sun. 2015. Multi-Classes Feature Engineering with Sliding Window for Purchase Prediction in Mobile Commerce. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 1048–1054.

[22] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE* 7, 1 (2003), 76–80.

[23] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. 2002. Using sequential and non-sequential patterns in predictive web usage mining tasks. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE, 669–672.

[24] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, Vol. 2007. 5–8.

[25] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*. Springer, 325–341.

[26] H Pragarauskas and Oliver Gross. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. (2010).

[27] Jiangtao Qiu, Zhangxi Lin, and Yinghong Li. 2015. Predicting customer purchase behavior in the e-commerce context. *Electronic Commerce Research* 15, 4 (2015), 427–452. https://doi.org/10.1007/s10660-015-9191-6

[28] Steffen Rendle. 2010. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 995–1000.

[29] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 57.

[30] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[31] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 811–820.

[32] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 81–90.

[33] Christian P Robert. 1995. Simulation of truncated normal variables. *Statistics and computing* 5, 2 (1995), 121–125.

[34] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.

[35] Guy Shani, Ronen I Brafman, and David Heckerman. 2002. An MDP-based recommender system. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 453–460.

[36] Erez Shmueli, Amit Kagian, Yehuda Koren, and Ronny Lempel. 2012. Care to comment?: recommendations for commenting on news stories. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 429–438.

[37] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. 2004. Maximum-margin matrix factorization. In *Advances in neural information processing systems*. 1329–1336.

[38] Jian Wang and Yi Zhang. 2011. Utilizing marginal net utility for recommendation in e-commerce. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 1003–1012.

[39] Jian Wang and Yi Zhang. 2013. Opportunity model for e-commerce recommendation: right product; right time. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 303–312.

[40] Tim Westergren. 2007. The music genome project. *Online: http://pandora. com/mgp* (2007).

[41] Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun. 2010. Temporal recommendation on graphs via long-and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 723–732.

[42] Jingyuan Yang, Chuanren Liu, Mingfei Teng, Hui Xiong, March Liao, and Vivian Zhu. 2015. Exploiting Temporal and Social Factors for B2B Marketing Campaign Recommendations. In *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 499–508.

[43] Ghim-Eng Yap, Xiao-Li Li, and S Yu Philip. 2012. Effective next-items recommendation via personalized sequential pattern mining. In *International Conference on Database Systems for Advanced Applications*. Springer, 48–64.

[44] Gang Zhao, Mong Li Lee, Wynne Hsu, and Wei Chen. 2012. Increasing temporal diversity with purchase intervals. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 165–174.

[45] Gang Zhao, Mong LI Lee, and Hsu Wynne. 2014. Utilizing purchase intervals in latent clusters for product recommendation. In *Proceedings of the 8th Workshop on Social Network Mining and Analysis*. ACM, 4.

[46] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. 2001. Using temporal data for making recommendations. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 580–588.