

# Online game props recommendation with real assessments

De-Chuan Zhan<sup>1</sup> · Jun Tang<sup>1</sup> · Zhi-Hua Zhou<sup>1</sup>

Received: 5 August 2016 / Accepted: 31 October 2016 / Published online: 18 November 2016  
© The Author(s) 2016. This article is published with open access at Springerlink.com

**Abstract** With the rapid development of smart mobile devices, phone games become an important way of entertainments. Benefitting from sophisticated payment environments of mobile platforms, e.g., Apple APP store, the In-APP purchases which sell equipments or virtual props bring in the main profits for game carriers and developers. Although virtual props from a certain type of smart phone game are monopolized by only one seller, like other commodities, products' recommendation is able to improve the profit margins as well. One main difference between virtual props recommendation and the general good recommendations lies in that the virtual props are closely related to the game contexts, and this will lead to complicated dependencies. Therefore, general recommendation systems without consideration on game contexts cannot perform very well. Besides, multiple types of props in one game may depend on different game characters of players, thus single player trends to buy only appropriate props for improving the skills of his game characters. Moreover, the purchase intentions of players are influenced by multiple factors, and will change over time. Therefore, it is desired recommendation approach to be capable of handling the role dependencies and concept variations.

This work was supported by NSFC (61333014) and CCF-Tencent Open Research Fund (RAGR20150117).

✉ De-Chuan Zhan  
zhandc@lamda.nju.edu.cn

Jun Tang  
tangj@lamda.nju.edu.cn

Zhi-Hua Zhou  
zhouzh@lamda.nju.edu.cn

<sup>1</sup> National Key Laboratory for Novel Software Technology, Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University, Nanjing, China

In this paper, we treat the game contexts as events from game log records, and model the game props recommendation into a multi-instance multi-label learning task for utilizing the complicated dependencies and capturing the rank of purchase intentions. We proposed three variants of solutions against the concept variation problem as well. Finally, we conduct comprehensive empirical investigations on real-world data sets and a series of real online smart phone games. The positive experiment results and increasing profit margins validate the remarkable effectiveness of our solutions.

**Keywords** Virtual props recommendation · Multi-instance multi-label learning · Real assessments on phone game platforms

## Introduction

With the rapid development of smart phones and mobile Internet, phone games become a huge entertainment market. The commercial objective of game carrier and developers (including individual developers and game companies) is to obtain profits and royalties from this emerging market. There are four major types of marketing and charging for phone games: the traditional purchase of software package still occupies a certain market, e.g., Infinity Blade; Purchase of game point cards is a new payment method which is often taken by Asian players, and the points consume according to the time of use; Payment by embedded advertisements is popular in phone games, but sometimes becomes annoying. For this kind of payoff, Cost Per Click/Mille/Action and Click Through Rate are the most important indicators on which researchers focus; In-APP purchase of game props, e.g., Fruit Ninja, becomes attractive with the maturity of mobile payment platforms (e.g., Alipay and Paypal). In this

paper, we focus on investigating the last type of trade mode and aim at increasing the conversion rate of props purchase in phone games.

In a phone game with In-APP purchase, players can enjoy the game for free and make efforts for upgrading the game levels or passing the barriers in games. Yet, they can also buy props to improve the abilities of game roles or accelerate the process of game for a better experience. Props in a certain game can be only bought and applied to the same APP; therefore, game props trade is a kind of monopoly sales to some extent. Nevertheless, game carriers and developers also can offer some recommendations on the game props, usually some bundles of equipments, for squeezing on the average profits to achieve higher volumes. Thus, recommendation can be applied in phone game props sales similar to common goods transactions.

Recommendation systems can be mainly categorized as collaborative filtering (CF) [7, 10, 14, 18] and content-based recommendation (CBR) [4, 9, 15]. There has been much work done both in industry and academia on developing new approaches to recommendation systems over the last decade. CF methods build models relied only on past user behaviors, i.e., CF methods focus only on the user-item activities which can be users' previous transactions or product ratings. There are two primary types of CF approaches: the neighborhood approach [3, 24] and latent factor models [10, 13, 21]. Recently, researchers also find that CF can be implemented with low-rank matrix completion techniques [1, 16]. CBR tries to sequentially find similar items in terms of content. It assumes that similar users should be interested in similar types of items and user/item similarities can be measured by preferences, profiles, locations, and social relationships etc. [12, 25]. However, there are some prominent problems in directly applying the state-of-the-art recommendation methods to game props recommendation. We first analyze these problems in this paper and categorized the problems into four aspects.

First, *Complicated Dependency on Context (CDC)* Game props are not only dependent on the purchase activities of players but also heavily relevant to the elaborated game contexts, e.g., the prop “yellow scarves armor” can be obtained by all players from “traitors” team on location “D”, and only can be equipped as a major prop for the character “Jiao” in the game of “Three Kingdoms”. That is, props can be dependent on lots of features, and even worse, player intentions for buying certain props may depend on the game events which are not directly related to the purchase activities. These kinds of complicated contexts can be neither well captured by CF nor easily measured in CBR with ordinary similarity functions.

Second, *Long-Distance Intervention (LDI)* Player purchasing game props can be affected by events occurred long ago. For example, prop  $k$  is compounded by prop  $i$  and  $j$  with 50 points of “spar”, and only can be possessed by character

of “priest”, i.e., there is an equipment upgrade tree for prop  $k$ . Any disturbances on even leaf nodes of the upgrade tree will affect the composition of prop  $k$ . Therefore, players have to make efforts to guarantee the possessions of prop  $i$  and  $j$  all the time, after they obtain both props. Otherwise, they need to buy prop  $k$  directly to avoid repetitive game tasks. Obviously, the problem induced by long-distance intervention makes props related to each other, and existing methods may be inappropriate for this recommendation task.

Third, *Props priority-Role Dependency (P-RD)* There are lots of props in a certain game, while different types of characters, e.g., “warrior”, “wizard”, and “tactician”, have different requirements for props. It is common that players focus on limited game characters, and then only the props required by these characters could be purchased with high probability. In this case, the recommendation methods for the task need to give a rank to finger out the sequence of items according to the preferences of player roles.

At last, *Concept Variation (CV)* The players may insist to obtain new props by playing games instead of purchasing. Besides, the budget of each player is obviously limited. These factors make the actual purchase volume far below the purchase intentions, especially when the game difficulty levels and players familiarities are relatively low. However, with the player level promotion and game difficulties increase, volume of actual purchase will increase. This is different from the missing entries phenomenon in other multi-label learning system [26]. Missing entries are caused by uncooperative actions of annotators, which are with randomness, while the concept variation is related to the changing of class priors.

The first two issues are mainly about how to take advantages of the given information (inputs) in the phone game props recommendation, while the last two issues are towards the desired properties of outputs. General recommendation methods consider player profiles or preferences as feature vectors to capture user purchase intentions. Nevertheless, user profiles and preferences cannot represent the Complicated Dependencies and Long-Distance Interventions in the game props recommendation tasks. To model those complicated dependencies, and interventions between the inputs and multiple outputs, we represent the entire log data of a player as a multi-instance example and cast the recommendation task into a multi-instance multi-label learning (MIML) problem. In our MIML formulation, the complicated relationships between all kinds of events are implicitly considered and will be captured by the MIML learner. By minimizing the ranking error of multi-labels within an example (bag) in our MIML solution, the props priority-Role dependency can be easily formalized. Moreover, we will show that the concept variation issue can be solved with three variants of our solution. One minor modification of the three variants is

directly applied to a commercial corporation, and ultimately improves the profitability of that company.

The paper is organized as follows: Sect. “Game props recommendation” is directed against the four issues discussed in this section, and introduces our game props recommendation solutions under the multi-instance multi-label framework. In Sect. “Experiments and assessments on real games”, we compare our solutions with other recommendation methods and MIML methods, and perform online assessments on real games platform. Both experiment results and profit margins are reported. Section “Conclusion” concludes this paper.

## Game props recommendation

Aiming at the first two issues posted above, this section first gives our multi-instance multi-label representation for the game prop recommendation task in detail, and then formulates the game prop recommendation into a MIML learning problem together with figuring out the basic solution, which minimizes the ranking error according to props priority-role dependencies. After that we analyze the problem of concept variations and modify the basic solution into three variants.

### Multi-instance multi-label representation

The problem brought forward by complicated dependency on context and long-distance intervention is mainly related to the representation of game props recommendation task. The complicated dependencies between the props and contexts make players buying specific prop related to lots of factors, e.g., the properties of the concerned prop, the events that players encountered in game threads.

We give a typical example here: Fig. 1 gives a screen shot of a game in the background of “Three Kingdoms in ancient



**Fig. 1** Screen shot illustration of a game similar to “Three Kingdoms” showing a hidden event

China”. In Fig. 1, the player was managing the army of “Wei” with the third person perspective, and was attacked by the ballista or stone thrower. From the logs, it can be obtained that this event was turned up with ID: 187759 and was at the coordinate which is near the right-bottom of the battle zone. From the logs, we can also find after this skirmishes, the same player encountered ballista several times when he was trying to occupy castles. The destructive power of the prop ballista and similar props like crossbow impressed this player very much. He, therefore, may consider comprehensively on the properties of the props ballista or crossbow, e.g., offensiveness, portability for a single soldier, etc., and as a consequence, the probability of purchasing these props may be increased greatly.

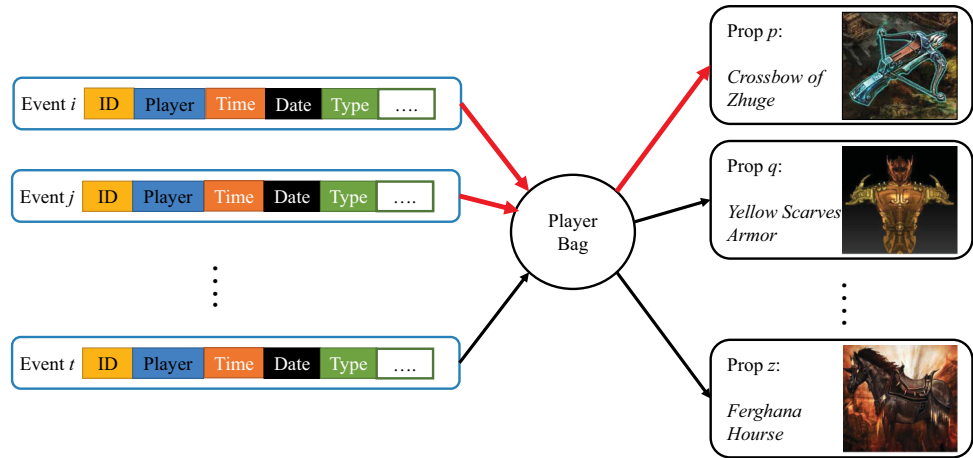
It is notable that the mentioned event is not directly related to the purchase activity, and is, therefore, denoted as a hidden event. However, a series of hidden events, indeed, influence the users purchase intentions. As a matter of fact, this mentioned player brought crossbows (a ballista style weapon for individual soldier) 2 days later according to the events log records.

Besides, the long-distance intervention bridges the indirect relationships between the acquisition behaviors and hidden events which were occurred long before, while these complicated dependencies cannot be well modeled by simply representing in feature vectors.

In this paper, we claim that the purchase activities can be triggered by a set of hidden events and the player information should be organized into multi-instance bags. In multi-instance representation, we treat each event record, like “ID: 187759, Player ID: 16749, Time 11:20 p.m., Date 12-04-2014, Location coordinate (12, 14), Type attack, etc”, as an instance, and all instances from one player are in one bag. For facilitating the description, an illustration is shown in Fig. 2. As shown in Fig. 2, each player is represented as a bag which contains large number of instances. Each instance in a bag describes an event, either hidden event or user purchase activity. The events are with feature ID, Player ID, Timestamp, etc. An example of the complicated dependency and long-distance intervention is also shown in Fig. 2, which is marked with red right-arrows. The example indicates that the purchase activity of prop  $p$  is related to two hidden events  $i$  and  $j$  which occur with different timestamp. It is obvious that we can treat the props which were (or would be potentially) bought as the labels. One player could buy more than one prop to improve the game characters abilities; therefore, one bag can map to multiple labels, i.e., Fig. 2 essentially offers a multi-instance multi-label representation. Then, the game props recommendation task can be cast into an MIML prediction problem.

Under the assumption that some events in the game log records trigger the purchase intention of a player, our task is to predict the props that will be potentially bought by the

**Fig. 2** Multi-instance multi-label representation for game prop recommendation task



**Table 1** Notations

$\mathbf{x}$	A single instance describes an event, $\mathbf{x} \in \mathbb{R}^D$
$\mathbf{x}_{ij}$	The $j$ -th instance in the $i$ -th bag, $i \in \{1, \dots, N\}$
$\mathbf{X}_i$	The $i$ -th bag, the collection of $\mathbf{x}_{ij}$ , $j = 1, 2, \dots, n_i$
$\mathbb{X}$	The example space of $\mathbf{x}$ , i.e., $\mathbf{x} \in \mathbb{X}$
$\mathbf{y}_i$	The label vector possessed by the $i$ -th bag
$y_{ij}$	The $j$ -th element of $\mathbf{y}_i$ , $y_{ij} = 1$ indicates $\mathbf{X}_i$ purchases prop $j$ in the data set, $j \in \{1, \dots, l\}$
$\mathbf{Y}$	The label matrix consisted of $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N$
$\mathbb{Y}$	The label space, i.e., $\mathbf{y}_i \in \mathbb{Y}$
$f$	A MIML learner, $f : 2^{\mathbb{X}} \rightarrow \mathbb{Y}$
$f_j$	The $j$ -th element of $f$ , $f_j(\mathbf{X}_i)$ is used to predict whether $j$ -th prop is desired by player $\mathbf{X}_i$

player, given his game records. To simplify our discussion, Table 1 lists the notations used in this paper.

We assume events triggering the purchase intentions, while the event records itself are represented with features at an abstract level. It is reasonable that some features act major roles and should be with special attentions. We, therefore, consider a derivative feature space, in which a single instance is represented by  $W_0^T \mathbf{x}$  and the example feature space comes after  $W_0^T \mathbb{X}$ , where  $W_0 \in \mathbb{R}^{D \times d}$ . Suppose that there is only one event record for a player  $i$ , i.e.,  $\mathbb{X}_i = \{\mathbf{x}\}$ , then the prediction function on whether this player has desire for the  $j$ -th prop can be in the form of

$$f_j(\{\mathbf{x}\}) = \mathbf{w}_j^T W_0^T \mathbf{x}, \tag{1}$$

where  $\mathbf{w}_j \in \mathbb{R}^d$  is the linear coefficients for  $f_j$ , and  $f_j \in \{0, 1\}$  is the  $j$ -th elements of  $f$ . Since a bag is labeled positive if and only if there is at least one positive instance in multi-instance learning, the prediction of  $f_j$  for a player with more than one event record can be

$$f_j(\mathbf{X}) = \max_{\mathbf{x} \in \mathbf{X}} f_j(\{\mathbf{x}\}). \tag{2}$$

Equation 2 indicates that event with the strongest stimulus in one bag will disclose the current players intension.

However, Eq. 1 is a linear predictor and then perhaps could not be able to capture the complicated dependencies between player intentions and event features, we follow [11] to redefine the prediction of single instance on prop  $j$  as:

$$f_j(\{\mathbf{x}\}) = \max_{l=1, \dots, K} \mathbf{w}_{j,l_j}^T W_0^T \mathbf{x},$$

where  $\mathbf{w}_{j,l_j}$  corresponds to the  $l_j$ -th sub-concept of purchase intention for prop  $j$ ,  $l_j = 1, 2, \dots, K$ , i.e., with overall considerations around the events and event features, the intension of buying prop  $j$  may be related to one particular feature combination of one event which is of paramount importance. The operator  $\max(\cdot)$  introduces non-linearities for the prediction function.

From the prediction functions listed above, we can find multiple linear coefficients  $\mathbf{w}_{j,l_j}$ ,  $j = 1, 2, \dots, l$ , corresponding to multi-labels. Therefore, by formulating game prop recommendation in MIML learning framework, the recommendation system can list multiple props for a player and the prediction values correspond to the degree of purchase intentions. Nevertheless, it is obvious that excessive and indiscriminate recommendations will degrade user experience and cause reductions of players. It is, therefore, crucial for carefully selecting a good strategy for learning with multiple labels.

### Ranking the multiple labels

According to the Props priority-role dependency (P-RD) property, one player is only intent to buy the props desired most by his characters in game. This implicitly indicates that there are some types of ranking orders among these props for a concerned player. Consequently, ranking orders among multiple labels should be emphasized during the MIML learning procedure. To tackle the issues of P-RD, we need to

minimize the ranking error for our multi-instance multi-label learner.

For a bag  $\mathbf{X}_i$  and one of its relevant labels, i.e.,  $\{y_{ij}, y_{ij} = 1\}$ , we define a ranking risk function inspired by [20]

$$R(\mathbf{X}_i, y_{ij}) = \sum_{k:y_{ik}=0} I[f_j(\mathbf{X}_i) < f_k(\mathbf{X}_i)], \tag{3}$$

where  $y_{ik} = 0$  indicates prop  $k$  is a irrelevant label and is not desired by player  $i$ .  $I[f_j(\mathbf{X}) < f_k(\mathbf{X})] = 1$  iff the condition  $f_j(\mathbf{X}) < f_k(\mathbf{X})$  is true, otherwise 0. Equation 3 counts how many undesired props are ranked before prop  $j$  desired by the  $i$ -th game player. Following the argument of [20], the ranking error of  $\mathbf{X}_i$  on label  $y_{ij}$  can be defined as:

$$\text{err}_{\text{rank}}(\mathbf{X}_i, y_{ij}) = \sum_{i=1}^{R(\mathbf{X}_i, y_{ij})} \alpha_i,$$

where  $\alpha_i \in [0, 1]$  is with positive, non-increasing values:  $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_l$ . For simplicity, we choose the mean pairwise error penalty and, therefore, can set  $\alpha_i = \frac{1}{i}$ . Then, the ranking error can be spread into all irrelevant labels as:

$$\begin{aligned} \ell^*(\mathbf{X}_i, y_{ij}) &= \mathbb{E}_{k:y_{ik}=0}[\text{err}_{\text{rank}}(\mathbf{X}_i, y_{ij})] \\ &= \sum_{k:y_{ik}=0} \frac{I[f_j(\mathbf{X}_i) < f_k(\mathbf{X}_i)]}{R(\mathbf{X}_i, y_{ij})} \text{err}_{\text{rank}}(\mathbf{X}_i, y_{ij}). \end{aligned} \tag{4}$$

Here, the ranking error is defined as the expectation of  $\text{err}_{\text{rank}}$ , which is attributed to the fact that the event of randomly choosing a irrelevant label  $k$  from the irrelevant label set  $\tilde{Y}_i$  is assumed to be uniform distributed. The probability of this event is  $\frac{1}{R(\mathbf{X}_i, y_{ij})}$ .

However, the ranking error defined with  $\ell^*$  in Eq. 4 is non-convex and discontinuous, and could be rather difficult to optimize directly. According to [5], the hinge loss is an optimal choice in all convex surrogate losses, we redefine ranking error as:

$$\ell(\mathbf{X}_i, y_{ij}) = \sum_{k:y_{ik}=0} \frac{|1 + f_k(\mathbf{X}_i) - f_j(\mathbf{X}_i)|_+}{R(\mathbf{X}_i, y_{ij})} \text{err}_{\text{rank}}(\mathbf{X}_i, y_{ij}),$$

where  $|z|_+ = \max(z, 0)$ . Since  $\ell(\cdot)$  is an upper bound for  $\ell^*(\cdot)$ , we can minimize the surrogate loss function  $\ell(\cdot)$  instead of that in Eq. 4 on whole data set as

$$\text{ranking err} = \sum_{i=1}^N \sum_{j:y_{ij}=1} \ell(\mathbf{X}_i, y_{ij}). \tag{5}$$

We can employ stochastic gradient descent (SGD) to minimize the ranking error defined in Eq. 5. We, therefore,

randomly sample a triplet  $(\mathbf{X}_i, y_{ij}, y_{ik})$ , where  $\mathbf{X}_i$  is a random multi-instance bag,  $y_{ij} = 1$  and  $y_{ik} = 0$ , i.e., the bag  $\mathbf{X}_i$  possesses prop  $j$ , and prop  $k$  is irrelevant. The triplet  $(\mathbf{X}_i, y_{ij}, y_{ik})$  introduces a loss:

$$\mathbb{L}(\mathbf{X}_i, y_{ij}, y_{ik}) = \text{err}_{\text{rank}}(\mathbf{X}_i, y_{ij})|1 + f_k(\mathbf{X}_i) - f_j(\mathbf{X}_i)|_+,$$

while the  $R(\mathbf{X}_i, y_{ij})$  in  $\text{err}_{\text{rank}}(\mathbf{X}_i, y_{ij})$  should be estimated before the optimization process of SGD. Here, we can randomly sample labels from the irrelevant label set of bag  $\mathbf{X}_i$  one by one, until a violated label  $k$ , which makes  $f_k(\mathbf{X}_i) > f_j(\mathbf{X}_i) - 1$  ( $j$  is a relevant label), is found at the  $v$ -th sampling step. Then, the approximated estimate of  $R(\mathbf{X}_i, y_{ij})$  equals to  $\lfloor |\tilde{Y}_i|/v \rfloor$ , where  $|\tilde{Y}_i|$  is the number of irrelevant labels for  $\mathbf{X}_i$ . Then, the approximation of ranking loss is for triplet  $(\mathbf{X}_i, y_{ij}, y_{ik})$  can be defined as:

$$\begin{aligned} \mathbb{L}(\mathbf{X}_i, y_{ij}, y_{ik}) &= \text{err}_{\text{rank}}(\mathbf{X}_i, y_{ij})|1 + f_k(\mathbf{X}_i) - f_j(\mathbf{X}_i)|_+ \\ &\approx \begin{cases} 0, & \text{if } k \text{ is not a violated label;} \\ S_{i,v}(1 + [\mathbf{w}_{k,l_k}^t]^\top W_0^t \mathbf{x}_k - [\mathbf{w}_{j,l_j}^t]^\top W_0^t \mathbf{x}_j), & \text{otherwise,} \end{cases} \end{aligned}$$

where  $\mathbf{x}_j$  is a key instance which achieves the maximum prediction value on the  $l_j$ -th concept of label  $j$ , and similarly,  $\mathbf{x}_k$  is a key instance which achieves the maximum prediction value on the  $l_k$ -th concept of label  $k$  and  $S_{i,v} = \sum_{i=1}^{\lfloor |\tilde{Y}_i|/v \rfloor} \frac{1}{i}$ . Consequently, the update rules for  $W_0, \mathbf{w}_{j,l_j}$  and  $\mathbf{w}_{k,l_k}$  is:

$$W_0^{t+1} = W_0^t - \gamma_t S_{i,v} \left( \mathbf{w}_{k,l_k}^t \mathbf{x}_k^\top - \mathbf{w}_{j,l_j}^t \mathbf{x}_j^\top \right), \tag{6}$$

$$\mathbf{w}_{j,l_j}^{t+1} = \mathbf{w}_{j,l_j}^t + \gamma_t S_{i,v} W_0^t \mathbf{x}_j, \tag{7}$$

$$\mathbf{w}_{k,l_k}^{t+1} = \mathbf{w}_{k,l_k}^t - \gamma_t S_{i,v} W_0^t \mathbf{x}_k, \tag{8}$$

where  $\gamma_t$  is the step size in the  $t$ -th iteration. Following this update rule, the MIML approach will minimize the ranking errors on the recommendation task.

### Concept variations

With the update rules listed in Eqs. 6, 7 and 8, it is expected that we can learn a MIML predictor in the form of Eq. 2 to forecast whether a player wishes to purchase a certain group of props, and then the recommendation system can target advertisements better to increase revenues. Although, in the test phase, one can obtain the recommendation value (real-value prediction from Eq. 2) of each prop for a player, and then the rank of all labels, the problem of choosing how many recommendations for the player is left untouched. We suggest introducing each bag a dummy label and training the MIML model to rank the dummy label before all irrelevant labels and after the relevant labels. The dummy label will somewhat change the training procedure: during triplet sampling in SGD, if  $y_{ij}$  is sampled, where  $j$  corresponds to the dummy

label, then the irrelevant label set remains unchanged, otherwise, the new irrelevant label set is consisted of the union of original irrelevant labels and the dummy label. This implies the rank of dummy label is between relevant labels and irrelevant labels. In this way, the recommendation system will be able to pick up the props before the dummy label in the rank sequence, and offer players recommendations. We consider this MIML approach as the base solution.

Nevertheless, the problem induced by concept variation increases the difficulties of accurate recommendation. We make different assumptions on the concept variation problem from three aspects, and then propose three different variational solutions in this subsection.

#### *Prior weighting solution: We-MIML*

The concept variation can be caused by the variation of class priors. The prop purchase intentions of player frequently change. This is closely related to the product life cycles (PLC). PLC is a statistical phenomenon of players behaviors. In the introductory phase of PLC, when there are only a few players, game carriers have to increase the investments on advertising, since recommendation systems, which focus on targeted ads., usually does not work. When there are a considerable amount of players, especially paying players, and the average revenue per player (ARPP) can be maintained to a certain degree, the game enters into the expansion phase. In this phase, recommendation system should be taken into consideration to reduce the cost of advertising. During this phase, the class prior, which is related to the purchase intentions for different props, frequently changes. Here, by attributing the concept variation problem to the drift of class priors, we first put forward a prior re-weighting solution here. Class prior re-weighting is widely applied in the class imbalance problem, while, in multi-label learning scenario, researchers found that the class imbalance problem is even serious and the class imbalance ratio may also change overtime [23].

In game prop recommendation task, as time goes by, players will realize that some props are essential for their game characters, and become interested in buying these props. This is the internal cause of class priors continuously changing over time. Therefore, MIML prediction functions trained months ago cannot predict purchase intentions at present accurately. Weighted MIML (We-MIML) is a variant solution of the base MIML learner, which can solve the problem of class priors variation. In We-MIML, we first calculate the most recent “class priors” in a time window, and use the temporary class priors to “weight” the base learner described in the previous section. In particular, we denote the purchase frequencies of each prop in a time windows (a short period before) as  $v_1, v_2, \dots, v_l$ , where  $v_j < 1$  and  $\sum_{j=1}^l v_j = 1$ . We consequently assume that the ground truth of class prior distribution can be approximated by  $\mathbb{D}_v = \{v_1, v_2, \dots, v_l\}$

and in the real online assessment in experiments, the prior is configured according to the frequencies a month before.

It is obvious that the ranking error function on the whole set of players should be re-written as

$$\text{ranking err} = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{j \sim \mathbb{D}_v; y_{ij}=1} [\ell(\mathbf{X}_i, y_{ij})]. \quad (9)$$

We can also employ the SGD to minimize the ranking errors on the whole set of players. However, at each iteration of SGD, we should sample the relevant labels  $y_j$  based on the distribution  $\mathbb{D}_v$  rather than sample them uniformly, since we claim the following Lemma stands.

**Lemma 1**  $\text{ranking err} = \mathbb{E}_{i, j \sim \mathbb{D}_v} [\mathbb{E}_k [\mathbb{L}(\mathbf{X}_i, y_{ij}, y_{ik})]]$ , where  $y_{ij} = 1, y_{ik} = 0, \mathbb{E}_{i, j \sim \mathbb{D}_v}(\cdot)$  denotes the expectation where bags follows uniform distribution and the relevant props  $j$  follows  $\mathbb{D}_v, \mathbb{E}_k$  indicates the expectation of choosing the irrelevant labels in the irrelevant label set  $\bar{Y}_i$ .

*Proof* This lemma directly follows these three facts that the probability of randomly choosing  $i$  based on uniform distribution; sampling the relevant props  $j$  following the distribution of  $\mathbb{D}_v$ ; and the probability of choosing irrelevant props  $k$  is  $1/R(\mathbf{x}_i, y_{ij})$  by given  $i$  and  $j$ .

The Lemma 1 simplifies the SGD sampling procedure for We-MIML. Owing to this lemma, when obtaining the triplet  $(\mathbf{X}_i, y_{ij}, y_{ik})$ , we only need to change the sampling strategy for the relevant label  $j$  to obey the distribution of  $\mathbb{D}_v$ . In practice, we duplicate those triplets containing the relevant label  $j$  according to the class prior  $v_j$  in We-MIML, to make the relevant labels obeying the class prior distribution.

#### *Sparse prediction solution: Sp-MIML*

In We-MIML, we have assumed the class prior changes yet can be approximated by the class frequencies in a time window. Here, in this part, we will first take a deep look at the relationships between matrix completion-based recommendation systems and the MIML-based solution, before introducing a new invariant assumption in concept variations for the game props recommendation task.

Recommendation approaches based on matrix completion assume similar users will have similar evaluation scores on certain items, while similar items will attract similar customer groups. This will lead to the user-item matrix with the nature of sparsity; specifically, the user-item matrix will be with low-rank. Our approach is different from general matrix completion-based recommendation approaches on two major aspects. First, although the player-prop label set, which is the desired output part of our MIML representation, acts like the user-item matrix, our MIML solution models the events

as bags in addition. Therefore, the multi-instance multi-label representation can capture more feature information. Second, matrix completion-based recommendation is substantially a transductive filling procedure. However, in our task, the players change constantly. In particular, there are new players joining in game continually. Recommendation system deals with the problem of users increasing with cold start techniques. Our MIML style approach, however, is an integrated solution which can directly predict the purchase intentions for *unseen* new players.

However, the invariant factor of similar user preferences, i.e., similar users will intent to purchase similar props, is still reasonable in our situation. Therefore, sparse assumptions can be made on the multi-label predictors. Without any loss of generality, we can suppose the set of multiple labels of all bags be denoted as  $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ ,  $\mathbf{Y} \in \mathbb{R}^{l \times N}$ . In low-rank matrix completion, researchers usually devote to solving the following problem as:

$$\begin{aligned} \arg \min_{\mathbf{F}} \quad & \text{rank}(\mathbf{F}) \\ \text{s.t.} \quad & \mathbf{F}_{i,j} = y_{ij}, \end{aligned}$$

where  $\mathbf{F}$  is a complete low-rank matrix and  $\mathbf{F}_{i,j}$  is the  $(i, j)$ -entry for  $\mathbf{F}$ . The constraints indicate the consistencies between  $\mathbf{F}_{i,j}$  and the known purchase behavior of player  $i$  on prop  $j$ . The  $\text{rank}(\cdot)$  is an operator for matrix rank calculation, and can be used to induce low-rank solution for  $\mathbf{F}$ . With the completed sparse matrix  $\mathbf{F}$ , the recommendation systems can push advertisements to existing players  $\{i\}$ ,  $i = 1, 2, \dots, N$ , yet can neither make use of the information from the events of players nor give recommendations to the new players. In our base solution, we essentially treat the  $\mathbf{F}$  as constituted by prediction values from a group of generalizable functions described in Eq. 2. As a result, for those known players, we have  $F_{i,j} = f_j(\mathbf{X}_i)$ , while for the new players, Eq. 2 will also give reasonable predictions. Then, by incorporating with the idea of sparse learning [26], we can enforce the linear coefficients in Eq. 2, i.e.,  $\mathbf{w}_{j,l_j}$ , to be sparse rather than directly regularize the prediction values  $f_j(\mathbf{X}_i)$ , and present a second variant solution Sp-MIML.

It is further notable that, in Sp-MIML,  $W_0$  should not be considered as a sparse/low-rank matrix. Since we have assumed  $W_0$  be a mapping matrix from the original feature space to the multi-label shared feature space for predicting every props,  $W_0$  should concentrate the information of all the predictors and the bags, therefore, should be reasonably configured as a “non-sparse” matrix.

In Sp-MIML, we only need to modify the update rules of Eq. 7 in base solution to the following Eq. 10 by appending with a proximal operator [22] to obtain the sparse solutions for the linear coefficients  $\mathbf{w}_{j,l_j}$  of each relevant prop  $j$  as

**Algorithm 1** Pseudo code for We/Sp-MIML training

**Require:** training bags,  $\gamma_t, \mathbb{D}_v$  (for We-MIML) or  $\eta$  (for Sp-MIML), initialized  $W_0$  and  $\mathbf{w}$ s, set  $t = 0$ ;  
 1: **while** Stop criterion doesn't meet **do**  
 2: Randomly sample a bag  $X_i$  uniformly and one of its labels  $j$  according to  $\mathbb{D}_v$  (for We-MIML) or uniformly (for Sp-MIML);  
 3: Obtain the most relevant instance  $\mathbf{x}_j$  of label  $j$  through  $\arg \max_{\mathbf{x}_j \in \mathbf{X}_{i,j}} f_j(\mathbf{x})$ ;  
 4: **for** Each  $k \in \tilde{Y}_i$  **do**  
 5: **if** The irrelevant label  $k$  is a violated label **then**  
 6: Update  $W_0^{t+1}$  according to Eq. 6;  
 Update  $\mathbf{w}_{j,l_j}^{t+1}$  according to Eq. 7 (for We-MIML) or Eq. 10 (for Sp-MIML);  
 Update  $\mathbf{w}_{k,l_k}^{t+1}$  according to Eq. 8;  
 7: Normalize  $W_0^{t+1}$ ,  $\mathbf{w}_{j,l_j}^{t+1}$ , and  $\mathbf{w}_{k,l_k}^{t+1}$ , respectively;  
 $t = t + 1$ ;  
 8: **Break**;  
 9: **end if**  
 10: **end for**  
 11: **end while**  
 12: **return**  $W_0^{t+1}$ ,  $\mathbf{w}_{j,l_j}^{t+1}$  and  $\mathbf{w}_{k,l_k}^{t+1}$ .

$$\mathbf{w}_{j,l_j}^{t+1} = \text{prox}_{\eta} \left( \mathbf{w}_{j,l_j}^t - \eta \mathbf{v}^{t+1} \right), \tag{10}$$

where

$$\mathbf{v}^{t+1} = \nabla \mathbb{L}(\mathbf{X}_i, y_{ij}, y_{ik}) \mathbf{w}_{j,l_j}^t - \nabla \mathbb{L}(\mathbf{X}_i, y_{ij}, y_{ik}) \tilde{\mathbf{w}}_{j,l_j} + \nabla F_{\tilde{\mathbf{w}}_{j,l_j}},$$

and

$$\nabla F_{\tilde{\mathbf{w}}_{j,l_j}} = \frac{1}{m} \sum_{i=1}^m \nabla \mathbb{L}(\mathbf{X}_i, y_{ij}, y_{ik}) \tilde{\mathbf{w}}_{j,l_j}. \tag{11}$$

Here,  $\nabla \mathbb{L}(\mathbf{X}_i, y_{ij}, y_{ik}) \mathbf{w}_{j,l_j}^t$  is the stochastic gradient of the ranking error function  $\mathbb{L}(\mathbf{X}_i, y_{ij}, y_{ik})$  at  $\mathbf{w}_{j,l_j}^t$ , and Eq. 11 defines the mini-batch gradient of a  $m$  size window.  $\tilde{\mathbf{w}}_{j,l_j}$  is the estimation of the optimal  $\mathbf{w}_{j,l_j}^*$  after every periodical mini-batch. The  $L_1$  proximal operator defined on vector  $\mathbf{w}$  is as follows:

$$\text{prox}_{\eta}(\mathbf{w}) = \text{sign}(\mathbf{w}) \max(\text{abs}(\mathbf{w}) - \eta, \mathbf{0}).$$

It is notable that in the Sp-MIML, the coefficients of irrelevant props should not be restricted to be sparse during the training stage, since the un-purchased props are not provided with the property of sparsity.

*Co-training style solution: Co-MIML*

The first two solutions make assumptions not only on the cause of concept variation but also on either class priors or sparseness. By noting that the fact of one player not buying prop  $j$  does not indicate his disinterest in prop  $j$ , we

can also assume that the un-purchased props should be unlabeled rather than negative labeled, i.e., the irrelevant labels in our MIML learner are substantially unlabeled instead of “irrelevant”. Then, the whole virtual props recommendation problem can be casted into a weak label problem [19].

Inspired by weak label learning and semi-supervised learning, it comes up with a third solution for concept variant problem in game props recommendation task, i.e., Co-MIML with co-training techniques. Co-training [6] is a representative paradigm of disagreement-based methods in semi-supervised learning. Co-training first learns a separate classifier for each base learner using labeled examples. The most confident predictions of each on the unlabeled data are then used to iteratively construct additional labeled training data for the opposite learner. Co-training can be beneficial if the labels are limited in semi-supervised scenario. In Co-MIML, the base learner is the MIMLFast, and there are two base learners involved.

In general, Co-training requires two independent and sufficient [6] feature sets (called views), yet for single view data sets, view splitting and training data splitting can be used for generating two classifiers, so as to achieve the performance improvements with co-training style learners.

In our recommendation task, there is only one group of features for all instances, thus we simply split the training data into two parts with bootstrap, namely  $\{\mathbf{X}_{d_1}, \mathbf{Y}_{d_1}^0\}$  and  $\{\mathbf{X}_{d_2}, \mathbf{Y}_{d_2}^0\}$ , and then train the base MIML learner on these two parts and obtain the  $f_{d_1}^0$  and  $f_{d_2}^0$ , respectively. After that we employ  $f_{d_1}^0$  and  $f_{d_2}^0$  to label the un-purchased props for each player, i.e., predict the zero elements in  $\mathbf{Y}_{d_1}^0$  and  $\mathbf{Y}_{d_2}^0$ . Then, we treat the union of the original positive labels and additional positive predictions which are with high confidences as pseudo labels  $\mathbf{Y}_{d_1}^1$  and  $\mathbf{Y}_{d_2}^1$ . We consequently re-train the base MIML learners on two parts of the training data, i.e.,  $\{\mathbf{X}_{d_1}, \mathbf{Y}_{d_1}^1\}$  and  $\{\mathbf{X}_{d_2}, \mathbf{Y}_{d_2}^1\}$ , respectively. We repeat the procedure for iterations and denote the learner after  $t$  iterations as  $f_{d_1}^t$  and  $f_{d_2}^t$ . When achieving the given maximum iteration times  $T$ , the intersection of the binary prediction value of  $f_{d_1}^T$  and  $f_{d_2}^T$  is used for decision.

We summarize the We-MIML/Sp-MIML approach in Algorithm 1 and the Co-MIML approach in Algorithm 2. The We/Sp-MIML is illustrated together, since they share a large portion of operations, and the differences between We/Sp-MIML in Algorithm 1 are marked with underlines. It is notable that Algorithm 2, i.e., Co-MIML, can be trained with *unseen* new players (with full unlabeled  $\mathbf{y}$ ) as well. To achieve this, we can simply set the multi-labels of the new players to zeros, and append them into the training set. Hence, Co-MIML is a semi-supervised version solution, and in our experiments, we compare all methods in a semi-supervised scenario.

---

### Algorithm 2 Pseudo code for Co-MIML Training

---

**Require:** training bags  $\mathbf{X}$  together with the partially annotated multi-labels  $\mathbf{Y}$ , max-iteration times  $T$ ;

- 1: Sample two different training sets  $\{\mathbf{X}_{d_1}, \mathbf{Y}_{d_1}^0\}$  and  $\{\mathbf{X}_{d_2}, \mathbf{Y}_{d_2}^0\}$  via Bootstrap;
- 2: Train two base MIML classifiers  $f_{d_1}^0$  and  $f_{d_2}^0$  with  $\{\mathbf{X}_{d_1}, \mathbf{Y}_{d_1}^0\}$  /  $\{\mathbf{X}_{d_2}, \mathbf{Y}_{d_2}^0\}$ , respectively;
- 3: **for** iter= 0 to  $T - 1$  **do**
- 4: Predict  $\mathbf{Y}_{d_1}^{\text{iter}+1,*}$  (0/1 binary prediction) for  $\mathbf{X}_{d_1}^{\text{iter}}$  with  $f_{d_1}^{\text{iter}}(\mathbf{X}_{d_1})$ , and sample a subset  $\mathbf{Y}_{d_1}^{\text{iter}+1}$  with the highest confidence in  $\mathbf{Y}_{d_1}^{\text{iter}+1,*}$ ;
- 5:  $\mathbf{Y}_{d_1}^{\text{iter}+1} = \mathbf{Y}_{d_1}^0 \cup \mathbf{Y}_{d_1}^{\text{iter}+1}$ ;
- 6: Predict  $\mathbf{Y}_{d_2}^{\text{iter}+1,*}$  (0/1 binary prediction) for  $\mathbf{X}_{d_2}^{\text{iter}}$  with  $f_{d_2}^{\text{iter}}(\mathbf{X}_{d_2})$ , and sample a subset  $\mathbf{Y}_{d_2}^{\text{iter}+1}$  with the highest confidence in  $\mathbf{Y}_{d_2}^{\text{iter}+1,*}$ ;
- 7:  $\mathbf{Y}_{d_2}^{\text{iter}+1} = \mathbf{Y}_{d_2}^0 \cup \mathbf{Y}_{d_2}^{\text{iter}+1}$ ;
- 8: Re-train the base MIML learner  $f_{d_1}^{\text{iter}+1}$  and  $f_{d_2}^{\text{iter}+1}$  on  $\{\mathbf{X}_{d_1}, \mathbf{Y}_{d_1}^{\text{iter}+1}\}$  /  $\{\mathbf{X}_{d_2}, \mathbf{Y}_{d_2}^{\text{iter}+1}\}$ , respectively;
- 9: **end for**
- 10: **return**  $f_{d_1}^T, f_{d_2}^T$ .

---

## Experiments and assessments on real games

Our empirical investigations are composed of four parts: (1) we compare the classification performance of the three variational MIML solutions with the base solution on six typical benchmark data sets [11]; (2) we then directly apply the three variational MIML solutions to three real operational data sets from a phone game corporation in Jiangsu, China. Comparisons of classification performance are made with state-of-the-art MIML methods; (3) we claim that our solutions are able to handle the problems derived from *complicated dependency on context*, *long-distance intervention*, *props priority-role dependency*, and *concept variation*, and can be used for props recommendation; therefore, we extensively compared our solutions with a series of state-of-the-art recommendation approaches; (4) we radically changed the recommendation systems of that commercial company with our solution in Dec. 2014, after that real assessments were carried out, and statistical reports on the real operational data validate that our solution has increased the profit margins of the company.

### Classification on benchmark data sets

Our proposed We/Sp/Co-MIML solutions are categorized as multi-instance multi-label learning methods, thus the effectiveness of these solutions should be first tested on MIML benchmark data sets. The detailed characteristics of the six benchmark data sets are summarized in Table 2, all data sets are as same as [11].



We split each data into ten parts without overlap and then choose nine of ten as training bags, while the remain one acts as the test set. Note that Co-MIML can be performed in a semi-supervised scenario; therefore, we use only 67%

of the training data as labeled data and the remaining 33% are unlabeled data. Co-MIML are trained with both label and unlabeled data, while other compared methods are only trained with labeled bags which at least possess one label.

For the base MIML learner, i.e., MIMLfast, the step size is  $\gamma_t = \gamma_0 / (1 + \mu\gamma_0 t)$  according to [20]. The parameters are selected by threefold CV on labeled training data with regard to ranking loss. Parameter candidates are as same as that in [11]. The parameter configurations for our variational solutions are the same to that of MIMLfast, while the maximum co-training iteration is set to 5 for Co-MIML, the maximum iterations for SGD is fixed at 10. The sparse parameter  $\eta$  for Sp-MIML simply equals  $10^{-5}$ . Since the effectiveness and efficiency of MIMLfast have been validated on these six benchmark data sets in literatures; therefore, we only com-

**Table 2** Data summarization

Data name	# Cls.	# Bags	# Lab./Bag	# Ins./Bag
Letter Frost	26	144	3.6	3.9
Letter Carroll	26	166	3.9	4.3
MSRA-v2	23	591	2.5	3
Reuters	7	2000	1.2	3.6
Bird song	13	548	2.1	18.7
Scene	5	2000	1.2	9

# Cls number of classes, # Lab./Bag number of labels per bag, # Ins./Bag number of instances per bag

**Table 3** Comparing our three variational solutions with MIMLfast (mean  $\pm$  SD) on the benchmark data sets

Data and methods	Hamming loss $\downarrow$	One error $\downarrow$	Coverage $\downarrow$	Ranking loss $\downarrow$	Avg. precision $\uparrow$
Letter Carroll					
MIMLFast	0.194 $\pm$ 0.011	0.237 $\pm$ 0.047	0.444 $\pm$ 0.030	0.179 $\pm$ 0.012	0.624 $\pm$ 0.030
We-MIML	0.135 $\pm$ 0.012	0.150 $\pm$ 0.060	0.389 $\pm$ 0.026	<b>0.133 <math>\pm</math> 0.010</b>	<b>0.708 <math>\pm</math> 0.024</b>
Sp-MIML	0.179 $\pm$ 0.014	0.148 $\pm$ 0.060	0.416 $\pm$ 0.033	0.149 $\pm$ 0.018	0.685 $\pm$ 0.037
Co-MIML	<b>0.132 <math>\pm</math> 0.006</b>	<b>0.148 <math>\pm</math> 0.049</b>	<b>0.378 <math>\pm</math> 0.030</b>	<b>0.133 <math>\pm</math> 0.016</b>	0.706 $\pm$ 0.028
Letter Frost					
MIMLFast	0.171 $\pm$ 0.012	0.247 $\pm$ 0.042	0.456 $\pm$ 0.040	0.192 $\pm$ 0.020	0.612 $\pm$ 0.034
We-MIML	0.135 $\pm$ 0.015	0.213 $\pm$ 0.051	0.414 $\pm$ 0.048	0.166 $\pm$ 0.029	0.647 $\pm$ 0.041
Sp-MIML	0.156 $\pm$ 0.018	0.210 $\pm$ 0.066	0.429 $\pm$ 0.028	0.173 $\pm$ 0.019	0.645 $\pm$ 0.037
Co-MIML	<b>0.123 <math>\pm</math> 0.012</b>	<b>0.199 <math>\pm</math> 0.040</b>	<b>0.386 <math>\pm</math> 0.041</b>	<b>0.149 <math>\pm</math> 0.017</b>	<b>0.673 <math>\pm</math> 0.035</b>
MSRC v2					
MIMLFast	0.177 $\pm$ 0.007	0.371 $\pm$ 0.022	0.302 $\pm$ 0.015	0.157 $\pm$ 0.010	0.611 $\pm$ 0.016
We-MIML	0.130 $\pm$ 0.016	0.368 $\pm$ 0.048	0.333 $\pm$ 0.025	0.179 $\pm$ 0.020	0.595 $\pm$ 0.028
Sp-MIML	0.167 $\pm$ 0.024	0.367 $\pm$ 0.031	0.303 $\pm$ 0.022	0.153 $\pm$ 0.013	0.609 $\pm$ 0.023
Co-MIML	<b>0.127 <math>\pm</math> 0.006</b>	<b>0.366 <math>\pm</math> 0.018</b>	<b>0.294 <math>\pm</math> 0.019</b>	<b>0.151 <math>\pm</math> 0.007</b>	<b>0.616 <math>\pm</math> 0.018</b>
Reuters					
MIMLFast	0.033 $\pm$ 0.005	0.058 $\pm$ 0.007	<b>0.037 <math>\pm</math> 0.005</b>	<b>0.018 <math>\pm</math> 0.005</b>	0.965 $\pm$ 0.005
We-MIML	0.036 $\pm$ 0.004	0.062 $\pm$ 0.011	0.040 $\pm$ 0.004	0.020 $\pm$ 0.004	0.960 $\pm$ 0.006
Sp-MIML	0.031 $\pm$ 0.004	0.058 $\pm$ 0.009	0.039 $\pm$ 0.004	0.019 $\pm$ 0.004	0.957 $\pm$ 0.005
Co-MIML	<b>0.030 <math>\pm</math> 0.009</b>	<b>0.055 <math>\pm</math> 0.006</b>	0.038 $\pm$ 0.010	<b>0.018 <math>\pm</math> 0.002</b>	<b>0.965 <math>\pm</math> 0.007</b>
Bird Song					
MIMLFast	0.138 $\pm$ 0.010	0.111 $\pm$ 0.015	0.194 $\pm$ 0.011	0.066 $\pm$ 0.006	0.869 $\pm$ 0.017
We-MIML	0.111 $\pm$ 0.021	0.081 $\pm$ 0.024	0.185 $\pm$ 0.029	0.060 $\pm$ 0.018	0.882 $\pm$ 0.028
Sp-MIML	0.124 $\pm$ 0.020	0.104 $\pm$ 0.028	0.171 $\pm$ 0.015	0.054 $\pm$ 0.011	0.880 $\pm$ 0.022
Co-MIML	<b>0.103 <math>\pm</math> 0.007</b>	<b>0.068 <math>\pm</math> 0.012</b>	<b>0.155 <math>\pm</math> 0.010</b>	<b>0.040 <math>\pm</math> 0.004</b>	<b>0.914 <math>\pm</math> 0.022</b>
Scene					
MIMLFast	0.199 $\pm$ 0.008	0.379 $\pm$ 0.024	0.219 $\pm$ 0.012	0.206 $\pm$ 0.013	0.753 $\pm$ 0.012
We-MIML	0.191 $\pm$ 0.015	0.361 $\pm$ 0.035	0.211 $\pm$ 0.016	0.196 $\pm$ 0.017	0.764 $\pm$ 0.019
Sp-MIML	0.195 $\pm$ 0.009	0.372 $\pm$ 0.024	0.215 $\pm$ 0.012	0.201 $\pm$ 0.010	0.757 $\pm$ 0.012
Co-MIML	<b>0.185 <math>\pm</math> 0.007</b>	<b>0.338 <math>\pm</math> 0.022</b>	<b>0.199 <math>\pm</math> 0.008</b>	<b>0.181 <math>\pm</math> 0.011</b>	<b>0.780 <math>\pm</math> 0.009</b>

The best performance is marked with bold on every data sets.  $\uparrow$  ( $\downarrow$ ) Indicates that the larger (smaller) the evaluation value is, the better the performances are

pared our three variational solutions with MIMLfast rather than all state-of-the-arts approaches on these benchmarks.

The performances are evaluated with five commonly used MIML criteria, i.e., Hamming Loss, One Error, Coverage, Ranking Loss, and Average Precision [27]. For the first four criteria, the smaller the better, while for Avg. Precision, the larger the better. Experiments are repeated for 30 times, the mean values and SD of all criteria are recorded in Table 3. From Table 3, it can be clearly found that Co-MIML is significantly better than MIMLfast on most data set (except for *Reuters*) with five evaluation criteria at significance level 95%, while We/Sp-MIML is better than MIMLfast on most data sets, except for on *MSRA-v2* and *Reuters*.

### Classification on real operational data

The real operational data for phone games can be with properties much different to that of benchmark data. Therefore, we test our solutions to the state-of-the-art MIML approaches, i.e., MIMLSVM [27] and MIMLfast, on the real operational

data. These data sets are extracted from the event log records of three different smart phone games, i.e., *Three Kingdoms*, *Rock Em Blocks*, and *Parkour*. We randomly pick up 15,000 players from the log records, each player is represented as a bag. Since the number of all events related to each player is extremely large, we only retain the latest events of the same type for a player, i.e., for *Three Kingdoms*, there are 336,000 instances in all (22.4 instances per player in average); for *Rock Em Blocks*, 264,000 instances in all (17.6 instances per player); and for *Parkour*, 157,500 instances (10.5 instances per player). The detailed characteristics of the operational data are listed in Table 4. All parameters for MIMLfast and our solutions are set as the same configurations as that in the benchmark experiments. Parameters of MIMLSVM are also selected with threefold CV.

Experiments are repeated for 30 times, the average value and SD of all five MIML evaluation criteria are recorded in Table 5. Table 5 clearly reveals that We/Sp/Co-MIML is significantly better than MIMLfast on all data set with four evaluation criteria, i.e., Hamming Loss, One Error, Ranking Loss, and Avg. Precision, at significance level 95%. While on *Rock Em Blocks*, MIMLFast is superiors to We-MIML and Co-MIML on Coverage; on *Parkour*, MIMLFast outperforms We-MIML measured by Coverage. In general, our three variational solutions perform better on these real operational data of phone games. This is primarily because these data sets are with the nature of props priority-role dependency and concept variation, and then the assumptions made

**Table 4** Data summarization

Game name	# Cls.	# Bags	# Lab./Bag	# Ins./Bag
Three Kingdoms	23	15,000	2.1	22.4
Rock Em Blocks	19	15,000	2.5	17.6
Parkour	11	15,000	1.7	10.5

# Cls number of classes, # Lab./Bag number of labels per bag, # Ins./Bag number of instances per bag

**Table 5** Comparing our solutions with state-of-the-art MIML approaches (mean  $\pm$  SD) on the real operational data from a commercial company

Games and methods	Hamming loss $\downarrow$	One error $\downarrow$	Coverage $\downarrow$	Ranking loss $\downarrow$	Avg. precision $\uparrow$
Three Kingdoms					
MIMLSVM	0.133 $\pm$ 0.016	0.245 $\pm$ 0.054	0.221 $\pm$ 0.024	0.050 $\pm$ 0.015	0.762 $\pm$ 0.032
MIMLFast	0.072 $\pm$ 0.012	0.089 $\pm$ 0.043	0.144 $\pm$ 0.015	0.021 $\pm$ 0.006	0.918 $\pm$ 0.032
We-MIML	0.062 $\pm$ 0.008	0.062 $\pm$ 0.037	<b>0.132 <math>\pm</math> 0.011</b>	<b>0.017 <math>\pm</math> 0.005</b>	<b>0.957 <math>\pm</math> 0.017</b>
Sp-MIML	0.060 $\pm$ 0.008	0.059 $\pm$ 0.035	0.136 $\pm$ 0.010	0.020 $\pm$ 0.005	0.944 $\pm$ 0.020
Co-MIML	<b>0.059 <math>\pm</math> 0.011</b>	<b>0.056 <math>\pm</math> 0.033</b>	0.133 $\pm$ 0.011	0.018 $\pm$ 0.004	0.954 $\pm$ 0.019
Rock Em Blocks					
MIMLSVM	0.154 $\pm$ 0.029	0.387 $\pm$ 0.065	0.173 $\pm$ 0.025	0.102 $\pm$ 0.025	0.702 $\pm$ 0.027
MIMLFast	0.111 $\pm$ 0.016	0.198 $\pm$ 0.043	0.093 $\pm$ 0.011	0.066 $\pm$ 0.013	0.869 $\pm$ 0.017
We-MIML	<b>0.091 <math>\pm</math> 0.024</b>	0.172 $\pm$ 0.049	0.098 $\pm$ 0.023	0.061 $\pm$ 0.017	0.885 $\pm$ 0.022
Sp-MIML	0.095 $\pm$ 0.015	0.164 $\pm$ 0.068	<b>0.084 <math>\pm</math> 0.012</b>	0.056 $\pm$ 0.011	0.884 $\pm$ 0.024
Co-MIML	0.097 $\pm$ 0.011	<b>0.157 <math>\pm</math> 0.041</b>	0.104 $\pm$ 0.015	<b>0.050 <math>\pm</math> 0.013</b>	<b>0.886 <math>\pm</math> 0.019</b>
Parkour					
MIMLSVM	0.223 $\pm$ 0.037	0.552 $\pm$ 0.047	0.392 $\pm$ 0.025	0.217 $\pm$ 0.023	0.446 $\pm$ 0.037
MIMLFast	0.160 $\pm$ 0.029	0.439 $\pm$ 0.043	0.298 $\pm$ 0.022	0.179 $\pm$ 0.025	0.551 $\pm$ 0.031
We-MIML	0.159 $\pm$ 0.034	0.415 $\pm$ 0.032	0.320 $\pm$ 0.016	<b>0.144 <math>\pm</math> 0.016</b>	0.577 $\pm$ 0.029
Sp-MIML	0.155 $\pm$ 0.027	0.390 $\pm$ 0.038	0.296 $\pm$ 0.020	0.158 $\pm$ 0.010	0.602 $\pm$ 0.038
Co-MIML	<b>0.145 <math>\pm</math> 0.029</b>	<b>0.384 <math>\pm</math> 0.039</b>	<b>0.284 <math>\pm</math> 0.021</b>	<b>0.144 <math>\pm</math> 0.022</b>	<b>0.609 <math>\pm</math> 0.024</b>

The best performance is marked with bold on every data sets.  $\uparrow$  ( $\downarrow$ ) Indicates that the larger (smaller) the evaluation value is, the better the performances are

**Table 6** Top-*k* precision values (mean  $\pm$  SD) of compared approaches

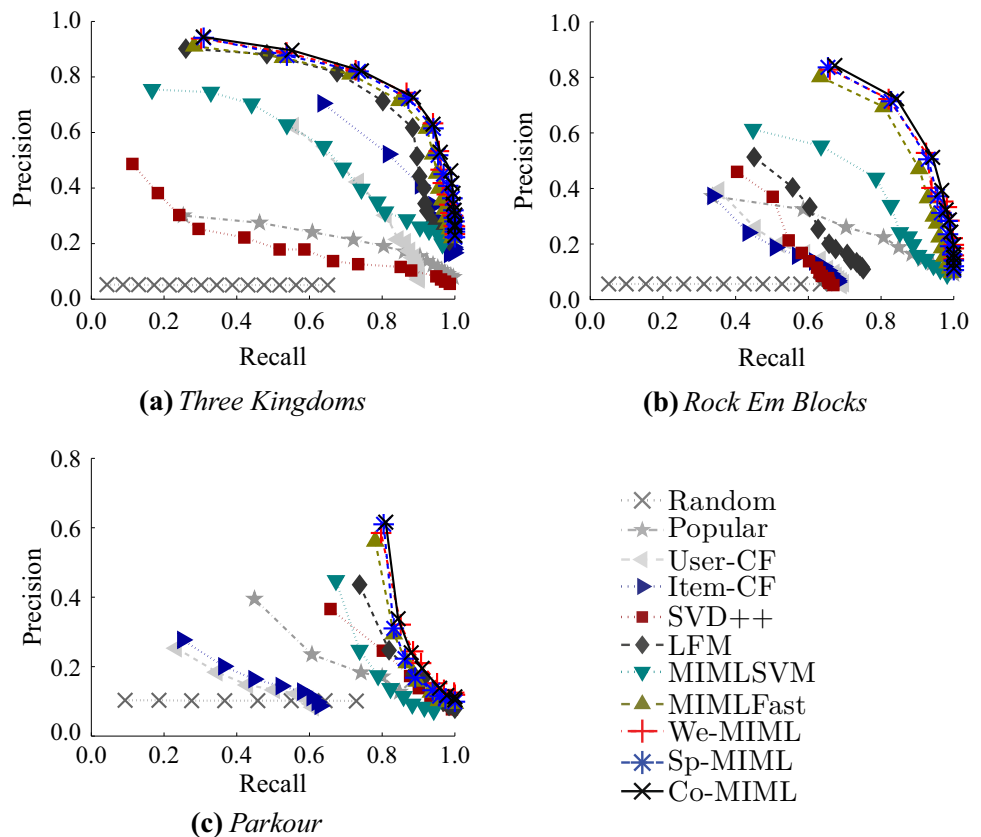
Games and methods	Top-1	Top-2	Top-3	Top-4	Top-5
<b>Three Kingdoms</b>					
Group A					
Random	0.051 $\pm$ 0.004	0.052 $\pm$ 0.003	0.052 $\pm$ 0.003	0.051 $\pm$ 0.002	0.051 $\pm$ 0.002
Popular	0.301 $\pm$ 0.009	0.275 $\pm$ 0.006	0.241 $\pm$ 0.003	0.214 $\pm$ 0.002	0.191 $\pm$ 0.001
User-CF	0.623 $\pm$ 0.010	0.422 $\pm$ 0.007	0.302 $\pm$ 0.003	0.215 $\pm$ 0.002	0.205 $\pm$ 0.002
Item-CF	0.705 $\pm$ 0.008	0.522 $\pm$ 0.006	0.410 $\pm$ 0.003	0.338 $\pm$ 0.002	0.287 $\pm$ 0.002
SVD++	0.487 $\pm$ 0.007	0.382 $\pm$ 0.007	0.303 $\pm$ 0.005	0.253 $\pm$ 0.003	0.222 $\pm$ 0.002
LFM	0.902 $\pm$ 0.015	0.882 $\pm$ 0.013	0.817 $\pm$ 0.009	0.713 $\pm$ 0.005	0.617 $\pm$ 0.003
Group B					
MIMLSVM	0.755 $\pm$ 0.035	0.745 $\pm$ 0.036	0.703 $\pm$ 0.038	0.628 $\pm$ 0.033	0.550 $\pm$ 0.032
MIMLFast	0.911 $\pm$ 0.030	0.870 $\pm$ 0.030	0.810 $\pm$ 0.033	0.715 $\pm$ 0.030	0.615 $\pm$ 0.028
We-MIML	0.938 $\pm$ 0.034	0.888 $\pm$ 0.028	<b>0.824 <math>\pm</math> 0.033</b>	<b>0.743 <math>\pm</math> 0.029</b>	<b>0.633 <math>\pm</math> 0.024</b>
Sp-MIML	0.941 $\pm$ 0.028	0.877 $\pm$ 0.029	0.820 $\pm$ 0.038	0.722 $\pm$ 0.034	0.615 $\pm$ 0.031
Co-MIML	<b>0.944 <math>\pm</math> 0.027</b>	<b>0.896 <math>\pm</math> 0.033</b>	0.822 $\pm$ 0.026	0.727 $\pm$ 0.026	0.630 $\pm$ 0.030
<b>Rock Em Blocks</b>					
Group A					
Random	0.056 $\pm$ 0.005	0.055 $\pm$ 0.001	0.056 $\pm$ 0.002	0.056 $\pm$ 0.002	0.057 $\pm$ 0.002
Popular	0.372 $\pm$ 0.009	0.325 $\pm$ 0.005	0.260 $\pm$ 0.002	0.224 $\pm$ 0.003	0.188 $\pm$ 0.001
User-CF	0.393 $\pm$ 0.011	0.257 $\pm$ 0.004	0.200 $\pm$ 0.003	0.167 $\pm$ 0.002	0.144 $\pm$ 0.001
Item-CF	0.373 $\pm$ 0.007	0.241 $\pm$ 0.004	0.188 $\pm$ 0.003	0.158 $\pm$ 0.003	0.137 $\pm$ 0.002
SVD++	0.460 $\pm$ 0.008	0.370 $\pm$ 0.006	0.213 $\pm$ 0.004	0.168 $\pm$ 0.002	0.138 $\pm$ 0.001
LFM	0.513 $\pm$ 0.017	0.404 $\pm$ 0.007	0.333 $\pm$ 0.005	0.254 $\pm$ 0.003	0.201 $\pm$ 0.002
Group B					
MIMLSVM	0.613 $\pm$ 0.038	0.553 $\pm$ 0.024	0.437 $\pm$ 0.015	0.339 $\pm$ 0.014	0.241 $\pm$ 0.008
MIMLFast	0.802 $\pm$ 0.033	0.694 $\pm$ 0.019	0.471 $\pm$ 0.015	0.366 $\pm$ 0.011	0.300 $\pm$ 0.009
We-MIML	0.828 $\pm$ 0.036	0.722 $\pm$ 0.021	<b>0.528 <math>\pm</math> 0.015</b>	<b>0.402 <math>\pm</math> 0.010</b>	<b>0.353 <math>\pm</math> 0.008</b>
Sp-MIML	0.836 $\pm$ 0.032	0.714 $\pm$ 0.019	0.505 $\pm$ 0.014	0.372 $\pm$ 0.011	0.312 $\pm$ 0.010
Co-MIML	<b>0.843 <math>\pm</math> 0.030</b>	<b>0.723 <math>\pm</math> 0.018</b>	0.512 $\pm$ 0.016	0.393 $\pm$ 0.012	0.323 $\pm$ 0.009
<b>Parkour</b>					
Group A					
Random	0.103 $\pm$ 0.006	0.103 $\pm$ 0.004	0.102 $\pm$ 0.001	0.102 $\pm$ 0.001	0.102 $\pm$ 0.001
Popular	0.395 $\pm$ 0.014	0.235 $\pm$ 0.004	0.183 $\pm$ 0.003	0.171 $\pm$ 0.001	0.127 $\pm$ 0.001
User-CF	0.253 $\pm$ 0.016	0.184 $\pm$ 0.007	0.147 $\pm$ 0.003	0.133 $\pm$ 0.001	0.122 $\pm$ 0.002
Item-CF	0.277 $\pm$ 0.014	0.201 $\pm$ 0.005	0.164 $\pm$ 0.001	0.144 $\pm$ 0.001	0.128 $\pm$ 0.001
SVD++	0.366 $\pm$ 0.012	0.246 $\pm$ 0.005	0.173 $\pm$ 0.002	0.138 $\pm$ 0.002	0.116 $\pm$ 0.001
LFM	0.436 $\pm$ 0.025	0.247 $\pm$ 0.012	0.178 $\pm$ 0.005	0.166 $\pm$ 0.004	0.126 $\pm$ 0.002
Group B					
MIMLSVM	0.448 $\pm$ 0.055	0.247 $\pm$ 0.029	0.174 $\pm$ 0.019	0.137 $\pm$ 0.013	0.114 $\pm$ 0.012
MIMLFast	0.561 $\pm$ 0.047	0.294 $\pm$ 0.027	0.211 $\pm$ 0.018	0.158 $\pm$ 0.012	0.126 $\pm$ 0.012
We-MIML	0.585 $\pm$ 0.050	0.321 $\pm$ 0.026	<b>0.244 <math>\pm</math> 0.017</b>	<b>0.210 <math>\pm</math> 0.012</b>	<b>0.152 <math>\pm</math> 0.010</b>
Sp-MIML	0.610 $\pm$ 0.046	0.310 $\pm$ 0.026	0.223 $\pm$ 0.018	0.168 $\pm$ 0.012	0.132 $\pm$ 0.011
Co-MIML	<b>0.616 <math>\pm</math> 0.048</b>	<b>0.338 <math>\pm</math> 0.022</b>	0.240 $\pm$ 0.016	0.195 $\pm$ 0.012	0.138 $\pm$ 0.012

**Table 6** continued

Games and methods	Top-1	Top-2	Top-3	Top-4	Top-5
The counts that MIML style methods are superior/inferior to all/any of Group A recommendation methods					
MIMLSVM vs. other methods in Group A	<b>2/1</b>	<b>2/1</b>	1/2	1/2	1/2
MIMLFast vs. other methods in Group A	3/0	<b>2/1</b>	<b>2/1</b>	<b>2/1</b>	1/2
We-MIML vs. other methods in Group A	3/0	3/0	3/0	3/0	3/0
Sp-MIML vs. other methods in Group A	3/0	<b>2/1</b>	3/0	<b>2/1</b>	<b>2/1</b>
Co-MIML vs. other methods in Group A	3/0	3/0	3/0	3/0	3/0

The best performance are bolded. The counts that each MIML style method in Group B is superior/inferior to all/any of Group A recommendation approaches are listed separately in the bottom of the table (a Group B method outperforming all Group A methods makes superior counts +1). Different superior/inferior count pairs are with bolditalic and italic values of table cell

**Fig. 3** Precision–recall plots of compared methods on offline data



by our solutions are satisfied on these data sets. In general, Co-MIML outperforms We/Sp-MIML in Table 5, which may be mainly caused by the limitations of labeled instances provided by the game scenarios.

**Recommendation back test on offline data**

Since our solutions will eventually be applied to recommendation, we conduct the main empirical investigations on the recommendation back tests, and compare our solutions with a series of recommendation approaches, including Random, Most-Popular (denoted as Popular in following text for short), User-CF [2], Item-CF [8], SVD++ [13], and LFM [17]. MIMLSVM [27] and MIMLFast [11] are also

compared and listed. The offline data are the log records, which are consistent with Sect. 3.2 from the same game corporation.

In particular, Random and Popular are two baseline approaches. Random picks up the props randomly according to the uniform distribution without replacement, i.e., it chooses the first recommendation with  $\frac{1}{I}$  in the whole props set, the second recommendation with  $\frac{1}{I-1}$  in the remains and so on. Popular first rank all the props in descending order on the entire props purchase logs, and recommends stationary props to all players. User-CF and Item-CF invoke the cosine similarity ( $\text{cos}_{\text{sim}}$ ) to measure the similarities among players or props, and we regularize the similarity by multiply the  $\text{cos}_{\text{sim}}$  by a variable relying on time, i.e., the similarities

used for CF recommendation are defined as  $\frac{1}{1+|t_{ij}-t_0|} \cos_{\text{sim}}$ . Here,  $t_{ij}$  is the timestamp for player  $i$  buying prop  $j$ , and  $t_0$  is the timestamp when the event log starts to record. Both SVD++ and LFM perform on the user-item purchase activities matrix. Parameters of our solutions are as same as that in Sect. 3.2. Other parameter configurations are kept identical to the default in literatures.

Top- $k$  precision is a frequently used evaluation criterion in recommendation systems [18], we consequently record the precision values (mean  $\pm$  SD) in Table 6 among the Top- $k$  ( $k = 1, 2, \dots, 5$ ) recommendations. For facilitating the discussion, we denote the conventional recommendation methods, i.e., Random, Popular, User-CF, Item-CF, SVD++, and LFM, as Group A and MIML style methods as Group B. The counts that each Group B method is better than all among the Group A approaches, and the counts that each Group B method is worse than any of the Group A methods, are listed as well. Table 6 clearly reveals the MIML style methods is superior than other recommendation methods on most situations. Moreover, We/Co-MIML is better than MIMLSVM and MIMLFast, and always outperform Group A methods on the offline data.  $t$  test at significance level 95% indicates that Co-MIML wins the all Group A methods on these data sets. Note that it is often the Top-1 and Top-2 precision values which are the most practical indicator of a recommendation system, thus We/Co-MIML are the most competitive. These phenomena occurs simply because compared with MIML-Fast, We-MIML considers the class prior; inspired by the matrix completion techniques, Co-MIML takes the property of sparseness on linearly coefficients  $W_0$  into consideration; and Co-MIML fully makes use of the information provided by the unlabeled data, i.e., these assumptions and designs in We/Sp/Co-MIML meet the natures of data in Sect. 3.2, and consequently yield the superior performances.

To extensively reveal the recommendation capability of We/Sp/Co-MIML, we record the precision-recall values on all props for all compared methods in Fig. 3. Figure 3 fingers out that our three variational solutions perform the best among all approaches, i.e., the precision-recall curves of We/Sp/Co-MIML always lie on the top-right corner of each plots. The training time costs are also recorded in Table 7, and it reveals that Co-MIML is faster than SVD++, LFM, and MIMLSVM. While Sp-MIML is the fastest among these seven methods, i.e., SVD++, LFM, MIMLSVM, MIMLFast, We/Sp/Co-MIML, and finally, the Sp-MIML reaches a good compromise between effectiveness and efficiencies.

## Real assessments online

Recommendation systems are designed for delivering the commercial values, thus after the comprehensive back tests on offline data, we eventually apply a slight minor modifi-

**Table 7** Comparison of training time costs (in seconds)

	Three Kingdoms	Rock Em Blocks	Parkour
SVD++	695.4	492.3	340.3
LFM	421.1	300.3	207.7
MIMLSVM	606.1	410.6	157.9
MIMLFast	137.2	95.5	33.6
We-MIML	691.2	563.2	412.8
Sp-MIML	136.6	92.8	33.1
Co-MIML	412.6	285.5	99.8

**Table 8** Statistics and influences on revenue benefits before/after applying the MIML recommendation system

	#A. P.	#Rcmd.	#Order
Oct. (without MIML solution)	55,735	231,848	37,939
Nov. (without MIML solution)	64,639	269,176	44,017
Dec. (with MIML solution)	75,436	341,483	54,265
Growth rates (Oct. $\rightarrow$ Nov.)	15.98%	16.10%	16.02%
Growth rates (Nov. $\rightarrow$ Dec.)	16.70%	<b>26.86%</b>	<b>23.28%</b>

Bold values clearly indicate the growth rate of orders increased from 16.02% to 23.28% while the rate of recommendations raises to 26.86% #A. P. number of active players, #Rcmd. number of recommendations, #Order number of purchases closely following the recommendation

cation of our three variational solutions to that phone game company from Jiangsu Province, China.<sup>1</sup>

It is notable that in the second half of 2014, we have noticed that the phone game “three kingdoms” operated by the company steps through the expansion phase into the stable phase, i.e., the number of players does not change drastically. Therefore, it creates a relatively reasonable environment for comparing the recommendation systems constituted from different approaches, LFM, and our system is tested then. For the We-MIML related part in our system, the time windows length is configured as 1 month, i.e., the distribution of  $v_j$  is approximated according to the class frequencies in last month.

In Table 8, we recorded the number of active players, the pop-ups of payment confirmations (the recommendations), and purchase activities closely following the pop-ups in Oct./Nov./Dec.2014, on the phone game “Three Kingdoms”. The Growth Rates on three indicators are also calculated and listed in Table 8, from which we can clearly find that the growth rate of orders is increased significantly (from 16.02% to 23.28%) even when the rate of recommendations raise from 16.10% to 26.86%, i.e., the absolute quantity of purchase has been improved to large extents after applying our recommendation solutions. After accounting, our solution brings 5.70% profit margin growth (measured by Average

<sup>1</sup> Reviewers are aware of that the minor modification containing but not confining to parameters settings, combination strategies of the proposed We/Sp/Co-MIML solutions, etc., is commercial secrets which cannot be exposed now.

Revenue Per Player, ARPP). The online A/B test comparing with the famous LFM is also performed in sandbox. The conversion efficiency, i.e.,  $\frac{\#Order}{\#Rcomds}$ , of our methods is 3.99%, which is better than LFM: 3.20%.

## Conclusion

In this paper, we systematically analyze the specific natures of game props recommendation for the first time, and attribute the problems into four aspects: the complicated dependencies on contexts, long-distance interventions, props priority-role dependencies, and concept variations. We deal with game prop recommendation under the multi-instance multi-label learning framework directly against the complicated dependencies and long-distance interventions, and minimize the ranking error to meet the requirements of the props priority-role dependencies.

To address concept variations, we proposed three variants of MIML approaches, i.e., We/Sp/Co-MIML, which consider the class prior weighting, sparse prediction, and co-training factors in MIML framework, respectively, for the first time. Experiments on benchmark and real operational data show the superior classification performance of We/Sp/Co-MIML, while the back tests and real online assessments reveal the effectiveness of our proposed methods on recommendation. In particular, the real online assessments have presented commercial values of MIML style methods and improved profitability of a game corporation. Our empirical investigations have successfully shown in commercial applications, such as game props recommendations, the natures of data, and application conform to the basic framework of MIML, and the additional assumptions made in We/Sp/Co-MIML variants.

Our practice on game props recommendation reflects the enormous potential of the multi-instance multi-label learning framework in commercial applications. Most business decisions are made after full considerations on various of dependencies, and MIML learning framework focuses on the implicit dependencies between complicated inputs and structural outputs, which can be, therefore, applied in such applications.

In future, we will keep on improving the MIML learning techniques on the ability of dealing with large-scale data, efficiency, parallel implementations, etc., and promote the effectiveness for expanding the applicability of MIML in other applications.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

1. Abernethy J, Bach F, Evgeniou T, Vert JP (2009) A new approach to collaborative filtering: operator estimation with spectral regularization. *J Mach Learn Res* 10:803–826
2. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans Knowl Data Eng* 17(6):734–749
3. Bell R, Koren Y (2007) Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In: *Proceedings of the 2007 IEEE international conference on data mining, Omaha, pp 43–52*
4. Bell R, Koren Y, Volinsky C (2007) Modeling relationships at multiple scales to improve accuracy of large recommender systems. In: *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, San Jose, pp 95–104*
5. Ben-David S, Loker D, Srebro N, Sridharan K (2012) Minimizing the misclassification error rate using a surrogate convex loss. In: *Proceedings of the 29th international conference on machine learning, Edinburgh, pp 1–8*
6. Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: *Proceedings of the 11th annual conference on computational learning theory, Madison, pp 92–100*
7. Chen WY, Zhang D, Chang EY (2008) Combinational collaborative filtering for personalized community recommendation. In: *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, Las Vegas, pp 115–123*
8. Deshpande M, Karypis G (2004) Item-based top-N recommendation algorithms. *ACM Trans Inf Syst* 22(1):143–177
9. Desrosiers C, Karypis G (2011) *A comprehensive survey of neighborhood-based recommendation methods*. Springer, New York, pp 107–144
10. Hofmann T (2004) Latent semantic models for collaborative filtering. *ACM Trans Inf Syst* 22(1):89–115
11. Huang SJ, Gao W, Zhou ZH (2014) Fast multi-instance multi-label learning. In: *Proceedings of the 28th AAAI conference on artificial intelligence, Quebec City, pp 1868–1874*
12. Jiang M, Cui P, Liu R, Yang Q, Wang F, Zhu W, Yang S (2012) Social contextual recommendation. In: *Proceedings of the 21st ACM international conference on information and knowledge management, Maui, pp 45–54*
13. Koren Y (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, Las Vegas, pp 426–434*
14. Koren Y (2010) Collaborative filtering with temporal dynamics. *Commun ACM* 53(4):89–97
15. van den Oord A, Dieleman S, Schrauwen B (2013) Deep content-based music recommendation. *Advances in neural information processing systems, vol 26*. MIT Press, Cambridge, pp 2643–2651
16. Salakhutdinov R, Srebro N (2011) Collaborative filtering in a non-uniform world: learning with the weighted trace norm. *Neural information processing systems, vol 24*. MIT Press, Cambridge, pp 1–8
17. Shen Y, Jin R (2002) Learning personal + social latent factor model for social recommendation. In: *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, Beijing, pp 1303–1311*
18. Shi Y, Larson M, Hanjalic A (2014) Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput Surv* 47(1):Article No. 3
19. Sun YY, Zhang Y, Zhou ZH (2010) Multi-label learning with weak label. In: *Proceedings of the 24th AAAI conference on artificial intelligence, Atlanta, pp 593–598*

20. Usunier N, Buffoni D, Gallinari P (2009) Ranking with ordered weighted pairwise classification. In: Proceedings of the 26th international conference on machine learning, Montreal, pp 1057–1064
21. Weimer M, Karatzoglou A, Le QV, Smola AJ (2008) CoFi<sup>rank</sup>-maximum margin matrix factorization for collaborative ranking. *Advances in neural information processing systems*, vol 20. MIT Press, Cambridge, pp 1593–1600
22. Xiao L, Zhang T (2014) A proximal stochastic gradient method with progressive variance reduction. *SIAM J Optim* 24:2057–2075
23. Xioufis ES, Spiliopoulou M, Tsoumakas G, Vlahavas I (2011) Dealing with concept drift and class imbalance in multi-label stream classification. In: Proceedings of the 22nd international joint conference on artificial intelligence, Barcelona, pp 1583–1588
24. Xue GR, Lin C, Yang Q, Xi W, Zeng HJ, Yu Y, Chen Z (2005) Scalable collaborative filtering using cluster-based smoothing. In: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval. Salvador, pp 114–121
25. Yin H, Sun Y, Cui B, Hu Z, Chen L (2013) LCARS: a location-content-aware recommender system. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, Chicago, pp 221–229
26. Yu HF, Jain P, Kar P, Dhillon IS (2014) Large-scale multi-label learning with missing labels. In: Proceedings of the 31st international conference on machine learning, Beijing, pp 1–8
27. Zhou ZH, Zhang ML, Huang SJ, Li YF (2012) Multi-instance multi-label learning. *Artif Intell* 176(1):2291–2320