MultiRec: A Multi-Relational Approach for Unique Item Recommendation in Auction Systems

Ahmed Rashed*
ahmedrashed@ismll.uni-hildesheim.de
Information Systems and Machine Learning Lab,
University of Hildesheim
Germany

Shayan Jawed*
shayan@ismll.uni-hildesheim.de
Information Systems and Machine Learning Lab,
University of Hildesheim
Germany

Lars Schmidt-Thieme
schmidt-thieme@ismll.uni-hildesheim.de
Information Systems and Machine Learning Lab,
University of Hildesheim
Germany

Andre Hintsches
Andre.Hintsches@vwfs.io
Volkswagen Financial Services AG, Braunschweig
Germany

ABSTRACT

In auction-based systems such as in used car centers and online auction websites, users usually bid on items, and those items get sold to their highest bidders. In these settings, every item is unique and can be sold only once, which means users' purchase histories will be unique, and no common items will exist across them. On the other hand, items will not have any historical sales at all. Such extreme settings pose a significant challenge to the current recommender systems models that rely on historical user-item interactions. While some of those models will not be applicable altogether, such as the matrix factorization models, neighborhood models, and even the naive most-popular model, the rest will need to rely only on items' attributes. In this paper, we address the challenges of auction-based item recommendation by proposing a simple multi-relational recommender model (MultiRec) that can seamlessly leverage user and item attributes along with auxiliary relational information such as the user's bidding history. Experiments on one proprietary dataset from Volkswagen Financial Services used-cars center, and on a real-world publicly available eBay dataset show that the proposed model significantly outperforms multiple state-of-art models in the task of auction-based unique item recommendation.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Learning from implicit feedback**; • **Information systems** → **Recommender systems**.

*Both authors contributed equally to this research.

RecSys '20, September 22–26, 2020, Virtual Event, Brazil

ACM ISBN 978-1-4503-7583-2/20/09…$15.00
https://doi.org/10.1145/3383313.3412242

KEYWORDS

Multi-Relational Learning, Collaborative Filtering, Auction Systems, Unique Item Recommendation, Attribute-Aware Recommender Systems

ACM Reference Format:
Ahmed Rashed, Shayan Jawed, Lars Schmidt-Thieme, and Andre Hintsches. 2020. MultiRec: A Multi-Relational Approach for Unique Item Recommendation in Auction Systems. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20), September 22–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3383313.3412242

1 INTRODUCTION

Recommender systems are currently employed in various applications, and they play vital roles in our daily lives, such as in online market places, social networks, and online media websites. Nowadays, there is also an increasing demand in employing recommender systems in other applications such as large auction-based systems, including but not limited to online auction markets and used cars centers. In such systems, users usually get presented with hundreds to thousands of items in a short period of time, and they have to filter and bid on a shortlist of those items to buy them. Given the time constraints, users will have little to no time to go through all items, and eventually miss most of the items that might have been interesting. On the other hand, some items will have few bids because the best-matched users could not find them on time. Providing users with a short personalized list of recommended items in such scenarios will significantly improve customer satisfaction and the overall distribution of bids across items. Unfortunately, though, there has been limited progress in designing recommender systems targeted explicitly to the auctions domain as it poses a significant challenge to the current models. In such settings, every item is unique and can only be bought once; hence, users' historical purchase records will be unique, and no common items will exist across them. Since items are unique, collaborative filtering models that rely heavily on mining similar patterns from the user-item interaction graph without using attributes [10, 14, 17, 21] will be rendered inapplicable. Some of those models, however, can be extended into hybrid models that take into account user and item attributes. However, recent studies have shown that even though such models show some lift over the basic versions, most of their performance lift is still coming from the patterns mined from the historical user-item interactions and not from the attributes [18]. These studies also showed that carefully tuned models that do not leverage attributes could also outperform state-of-art hybrid models.

In this paper, we try to tackle this problem by proposing a simple non-linear multi-relational model (MultiRec). The proposed model utilizes user and item attributes along with the auxiliary relational information that generally exists in most auction settings such as the bidding history of users and the final sale prices of items. The model is divided into two main components that are trained in an end to end fashion. First, the model co-embeds the user and item attributes using non-linear embedding networks similar to the model proposed by Rashed et al. [18]. It then constructs separate scoring functions for predicting the likelihood of purchasing, bidding, and the expected sale price for which a user will purchase the target item. Our research thus fills an essential gap in the recommender systems literature and shows promising results in fueling the growth of these important but neglected online auctions. The contributions of this paper can be summarized as follows :

- To tackle the challenges of unique item recommendation in auction systems, we introduce a simple multi-relational non-linear co-embedding model (MultiRec). MultiRec leverages users and items attributes along with auxiliary relational information that generally exists in auction settings such as user's purchase and bidding histories, and final sale prices of items. To the best of our knowledge, this is the first model that is designed specifically for such problem settings.
- We propose an end-to-end learning procedure for MultiRec that trains and optimizes all model components seamlessly for the multiple auxiliary relations that exist in auction systems.
- We conduct multiple experiments on a proprietary dataset from Volkswagen Financial Services used cars center and on a publicly available eBay auction dataset. Results show that the proposed MultiRec model outperforms state-of-the-art hybrid models for personalized item recommendation and achieves improvements of up to 9.12% and 135.7% on the two datasets. It also shows that the auxiliary relational information can provide a significant lift in model performance in such settings.

2 RELATED WORK

On a fundamental level, recommender systems can be categorized into two main groups, collaborative filtering models, which also include the attribute-aware subgroup of models and content-based models. The working principle behind collaborative filtering approaches is that if users have a similar purchase history, then future recommendations could be crafted by exploiting these similar purchases. On the other hand, content-based methods are more user-centric, where recommendations are followed by comparing item similarities between the user's past liking and contender items to be recommended. The building block for the content-based methods is hence the user/item features that are exploited for learning methods. Building on the same principle, researchers proposed to bridge the two approaches in the form of hybrid methods that utilized user and item features in the collaborative filtering process.

There have been a plethora of recommender system approaches that have attempted to incorporate user and item attributes. Of specific interest are the deep neural network-based methods, that are most related to our approach. We first note works that fall under the umbrella of explicit feedback-based approaches where the task is to predict ratings that indicate users' preferences on items. We note the work by Li et al. [11] where a hybrid model was designed via capturing user and item attributes latent representations via an autoencoder and then regularizing using a contrastive loss between these learned autoencoder representations and matrix factorization latent representations. Kim et al. [9] proposed to learn item latent features by applying convolutions to documents and then to multiply these with user's latent features from matrix factorization with overall learning proceeding in a coordinate descent framework. Additional models that exploited attributes were proposed by Zhang et al. [30], where autoencoders again were used to learn items attributes which are then subsequently normalized and added to the item latent features from matrix factorization. This approach differs from the one proposed by Li et al. [11] as the optimization was done sequentially in the sense that training of the autoencoder was disjoint from matrix factorization. Another approach was recently proposed by Rashed et al. [18], where a deep neural network was devised that, in contrast to prior approaches, noted so far directly embedded user and item features into a joint latent space together with their attributes. Moreover, Laplacian features were exploited from the user-item interaction graph that leads to a significant lift in modeling accuracy. This model is also used as a baseline in this paper, and the details are further elaborated in the experiments section.

Implicit feedback, on the other hand, has recently garnered specific attention in comparison to explicit feedback works. The task in implicit feedback works is to learn on only binary indicators of user preferences and output a probability for the latest items that can then be ranked to build a recommendation list. We note autoencoder based methods that proposed to model collaborative filtering jointly with side information [25, 28, 29]. Wang et al. [25] proposed a Bayesian learning method designed for jointly learning item's attributes and collaborative filtering. The method was based on an autoencoder that was trained on the item's attributes, and the bottleneck then used to draw latent features for items for collaborative filtering bridging the two tasks. Building upon the popularity of the autoencoder works, Wu et al. [28] proposed to reconstruct the users' feedback vectors jointly together with users-specific attributes. Through an ablation study, it was also shown that these attributes were vital to the success of the model. A significant finding was reported by Zhang et al. [29], where item latent representations are learned from different heterogeneous sources such as images and text and simultaneously integrated together with the latent offset vector from past user-item interactions. A different approach to auto-encoders that can be extended easily to attribute-aware settings has been proposed by He et al.[7], where instead of modeling the user and item input vectors with deep learning methods they proposed resorting to matrix-factorization for collaborative filtering by exploiting deep non-linear feedforward network for modeling the user-item interaction matrix. An extended attribute-aware version of this model is also used as a baseline in this paper, and further details are elaborated in the experiments section. Moreover, we also note works where the attention mechanism was used to specifically cater to the attributes to learn rich features for collaborative filtering [1, 24].

Attribute-aware and content-based models have been shown to be effective in settings where no prior information exists for the item to be recommended [3, 22]. Given that in our problem formulation, all used vehicles are by virtue unique, we propose to exploit the user's and item's attributes directly. Wang et al. [27] proposed a unified model that optimized for content features and parameters guiding user preferences. The content features were learned from a deep Bayesian network, while the recommendations are generated by an inner product between the learned features and user preferences. We note the work by Huang et al. [8], where interestingly, a latent semantic model, was proposed to learn the distance, a cosine similarity explicitly between queries and documents for modeling click-through data. We further shed light on this model in the experiments section, as we propose to use this model as a baseline. Also notably, this model was further on expanded to a multi-view setting catering for cross-domain features from multiple sources by Elkahky [2] where pairwise similarities were learned between the user and each of the different item views. Additionally, we note the approach by Suglia et al. [23], where deep recurrent methods were utilized to model textual features of item descriptions. These learned representations were concatenated in contrast to previous noted work and then fed to logistic regression layers. Moreover, Lian et al. [12] proposed a hybrid model that augments item latent features from a deep neural network-based collaborative filtering with separately embedded latent content features.

Another sub-class of attribute-aware approaches that merits a discussion here are the factorization machine [20] and methods based on it [5, 6, 13]. Factorization Machines (FM) have shown to learn feature interactions automatically, leading to significant gains in performance. Successful extensions include the work by Guo et al. [5], where the objective was laid to learn low and high order feature interactions with FM and deep feedforward networks, respectively. We further shed light on this model in the experiments section, as we proposed to use it as a baseline. Since the learning was done end-to-end, the model was able to generalize well to both sorts of interactions. On the other hand, a parallel effort was made by He et al. [6] to design a hierarchical structure with FM as the building block and the structure advancing with a deep feedforward network to learn non-linear high order feature interactions from this representation. More recently, Lian et al. [13] focused on learning feature interactions explicitly like similar to the work of Wang et al. [26] but in a vector setting like in FM in contrast to the feedforward networks which model at the feature level.

Additionally, we take note of the multi-relational approaches in the literature. Broadly speaking, the approaches can be divided into either models that focus on item-item relations [16], user social relations [4], and a combination of explicit and implicit user-item relations [10]. The recurring theme in these works is to exploit rich latent information from multiple sources as input that can effectively augment the generalizability of the model in sparse settings. Pioneering work from Koren et al. [10] exploited users' implicit interactions and their explicit feedback jointly in a matrix factorization model. Guo et al. [4] proposed to leverage implicit and explicit trust information for modeling users. From a methodological perspective, the model is built as an extension to the model proposed by Koren et al. [10] and hence captures multiple relations via low-rank matrix factorization. On the other hand, the item-item relational information was modeled in a content-based setting to recommend users with similar items that they preferred historically in the work by McAuley et al. [16]. More specifically, since the method focused on learning visual similarity, it utilized Convolutional Neural Networks that learned distances between features of items that were brought together or served as substitutes. We note the work's similarity to the work by Huang et al.'s and Elkahky et al.'s [2, 8], which learn similar distance-based functions to model items' similarities. A general relational model was most recently proposed by Rashed et al. [19], for multi-relational classification. The model comprehensively learned structured data for different entities in the relational graph and modeled the auxiliary relations between entities as auxiliary tasks while classifying the target relation jointly. The model additionally demonstrated the effect of non-linear modeling in learning rich latent features in multi-relational settings. Moreover, the auxiliary tasks also relate to our multi-objective optimization as opposed to input-only multi-relational information. Nevertheless, a major limitation of this model is that it cannot model and differentiate between multiple relations that have the same interacting entity types because it uses a simple non-parametric dot-product scoring function for all relations.

Finally, we note the works concerning auctions for online advertisement, where the focus has been on either estimating click-through rate [5, 6, 13] or bid estimation [5, 6].

With motivation set forth from the literature review, we draw the following insight: a plethora of works where attributes from both users and items have been actively used to model the user preferences in either explicit or implicit settings. However, for an auction-based recommender system, our work is the first to tackle a multi-objective end-to-end optimization for implicit purchases, implicit bidding behavior, and regressing the final price of all purchases all the while leveraging user and item attributes.

3 PROBLEM DEFINITION

In item recommendation tasks, there exists a set of users $U := \{u_1, u_2, ..., u_N\}$ with attributes $A_u \in \mathbb{R}^{N \times J}$, a set of items $I := \{i_1, i_2, ..., i_M\}$ with attributes $A_i \in \mathbb{R}^{M \times O}$, and a sparse binary interaction matrix $R \in \mathbb{R}^{N \times M}$ that indicate user's preferences on items through historical implicit feedback interactions such as purchase records R_P.

$$R_{P_{(u,i)}} = \begin{cases} 1, & \text{if } (u,i) \text{ is observed in } R_P \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The main goal of the recommendation task is to estimate the missing scores in R that will indicate the likelihood of an item being purchased or consumed by users. These scores are then used to provide a ranked personalized short-list of items to users. In traditional settings, we generally have one main interaction matrix R available; e.g., in online markets, this matrix usually represents

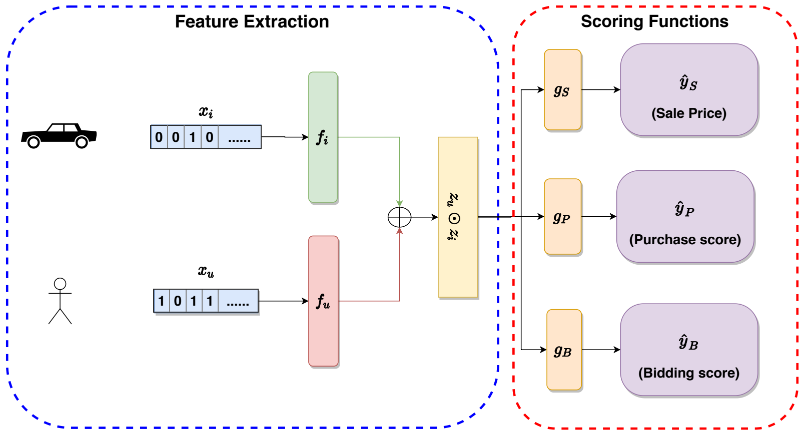


Figure 1: Illustration of the MultiRec model, which is composed of two main modules, namely feature extraction and scoring functionality for respective tasks. Interacting entities are embedded separately with non-linear full connected layers and subsequently multiplied element-wise. Scoring functions tune the common representations to task-specific representations and output predictions.

purchase interactions between users and items, while in online media websites, it represents items consumption by users.

On the other hand, auction-based settings have slightly different characteristics. In such settings, users tend to bid on items first, and then the items get sold to the highest bidder. Having such a dual-phase process means we have two implicit feedback relations between users and items, which are the bidding and purchase relations, respectively. The bidding relation with its interaction matrix $R_B \in \mathbb{R}^{N \times M}$ mostly provides a weak signal that indicates users' initial interests in items, while the purchase relation $R_P \in \mathbb{R}^{N \times M}$ provides a stronger signal that indicates users' willingness to buy the items for the highest price. Another dimension of the purchase relation is the final sale price; as a higher sales price indicates a more substantial interest in buying the item than a lower sales price. Therefore, the purchase relation can be further divided into two sub-relations, a binary implicit feedback relation for purchase records R_P and a positive real-value implicit feedback relation for the final sale prices $R_S \in \mathbb{R}^{N \times M}$. More formally, we can define all the auxiliary relations that will be considered in this work along with the main purchase relation R_P as follows:

$$R_{B_{(u,i)}} = \begin{cases} 1, & \text{if } (u,i) \text{ is observed in } R_B \\ 0, & \text{otherwise} \end{cases}$$
$$, R_{S_{(u,i)}} = \begin{cases} \text{price}(u,i), & \text{if } (u,i) \text{ is observed in } \in R_P \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

In this work, we limit our focus on the binary implicit bidding relation; however, we note that following the same notation, the bidding price relation could also be subsequently derived and exploited.

It is also worthy to note that since every item can only be sold once, the dimension M in all relations matrices will increase with every new item being introduced, and it cannot be fixed.

Finally, since the purchase relation provides the ultimate signal that indicates the user's maximum willingness to buy an item, the goal of auction-based unique item recommendation task will be to estimate the missing likelihood scores in the main purchasing matrix R_P by utilizing the previously observed interaction in all available relations $\mathcal{R} := \{R_P, R_B, R_S\}$ to present users with a ranked short-list of items.

4 METHOD

The proposed MultiRec model mainly consists of a feature extraction component that relies on non-linear co-embedding layers inspired by the work of Rashed et al. [18] and a set of scoring functions g_r for the available relations $\mathcal{R} := \{R_P, R_B, R_S\}$. We will discuss each component in detail in the following subsections. The full architecture of MultiRec is illustrated in Figure 1.

4.1 Feature Extraction

Given the set of all users $U := \{u_1, u_2, ..., u_N\}$ and items $I := \{i_1, i_2, ..., i_M\}$ with their attributes a_u and a_i, we define two non-linear co-embedding functions $f_U : \mathbb{R}^{N \times (N+J)} \longrightarrow \mathbb{R}^{N \times D}$ and $f_I : \mathbb{R}^{M \times O} \longrightarrow \mathbb{R}^{M \times D}$ to extract their latent feature vectors z_u and z_i as follows:

$$z_u = f_U(x_u; \theta_U), \qquad z_i = f_I(x_i; \theta_I) \tag{3}$$
$$x_u = [v_u, a_u], \qquad x_i = a_i \tag{4}$$

where x_u represents the user's personalized input feature vector, which is a concatenation between the one-hot encoding vector v_u of the user id and the user's attributes a_u. Item's input vector is denoted as x_i, which consists only of the item attributes a_i as we explicitly indicated that items are unique. We denote the latent embedding vectors of user and item input vectors with z_u and z_i, respectively. Finally, f_U and f_I are a series of non-linear fully connected layers with parameters θ_U and θ_I, respectively.

4.2 Scoring Functions

After extracting the users and items latent non-linear embeddings, we define an independent scoring function g_R for every relation in

$\mathcal{R} := \{R_P, R_B, R_S\}$ that utilizes the learned embedding to predict the target values $\hat{y}_P$,$\hat{y}_B$ and $\hat{y}_S$ for any user-item pair. Notice that g_R can be a series of non-linear fully connected layers with parameters θ_R or a simpler non-parametric function such as a dot-product operation. The scoring functions are defined as follows:

$$\hat{y}_B = g_B(z_u \odot z_i; \theta_B) \tag{5}$$

$$\hat{y}_S = g_S(z_u \odot z_i; \theta_S) \tag{6}$$

$$\hat{y}_P = g_P(z_u \odot z_i; \theta_P) \tag{7}$$

where g_B, g_S, and g_P are the scoring functions for the bidding, sale price and purchase relations respectively. θ_B, θ_S, and θ_P are the networks weights for the scoring functions g_B, g_S, and g_P. The element-wise product of two vectors is denoted by $\odot$. Notice that the input vectors to the scoring functions are constructed by the element-wise product of the user's and item's latent embeddings instead of concatenating them. The main goal for the element-wise product is to maintain the alignment of the features between the latent vectors and to capture the multiplicative interaction between them similar to matrix factorization.

4.3 Optimizing MultiRec

In this section we discuss the end-to-end learning approach for optimizing the MultiRec model. Given the scores $\hat{y}_B$, $\hat{y}_S$ and $\hat{y}_P$ for all available relations $\mathcal{R} := \{R_P, R_B, R_S\}$, we define three separate loss functions as follows:

$$\mathcal{L}_P = -\sum_{(u,i)\in\mathcal{Y}_P^+\cup\mathcal{Y}_P^-} y_P \log(\hat{y}_P) + (1-y_P)\log(1-\hat{y}_P) \tag{8}$$

$$\mathcal{L}_B = -\sum_{(u,i)\in\mathcal{Y}_B^+\cup\mathcal{Y}_B^-} y_B \log(\hat{y}_B) + (1-y_B)\log(1-\hat{y}_B) \tag{9}$$

$$\mathcal{L}_S = -\sum_{(u,i)\in\mathcal{Y}_S^+} (y_S - \hat{y}_S)^2 \tag{10}$$

where $\mathcal{Y}_P^+$, $\mathcal{Y}_B^+$ and $\mathcal{Y}_S^+$ represent the observed positive interactions in R_P, R_B and R_S. On the other hand, $\mathcal{Y}_P^-$ and $\mathcal{Y}_B^-$ represent the unobserved negative interactions in R_P and R_B. We have employed binary-cross entropy loss for optimizing the purchase and bidding scoring functions, which are generally optimized by sampling positive and negative pairs from the observed and unobserved interactions uniformly across users. For the sale price scoring function, we employed a means squared error loss, which is trained on only the observed interactions $\mathcal{Y}_S^+$ in R_S. We used an equal ratio between positive and negative samples during the training phase because it provided the best prediction performance.

It is also important to note that the sampling is done uniformly across users to avoid having a bias toward users who have more interactions and to match the user sampling strategy of the leave-one-out evaluation protocol [7] that will be used in the experiments section.

Finally, the MutliRec model is then optimized by minimizing the overall objective function $\mathcal{L}(\Theta)$, which is the weighted sum of three loss functions as follows:

$$\mathcal{L}(\Theta) = \sum_{R\in\mathcal{R}} \alpha_R \mathcal{L}_R \tag{11}$$

where α_R is the loss weight of each relation in $\mathcal{R}$ and Θ is a set of all model parameters. The full pseudo-code of MultiRec is described in Algorithm 1.

Algorithm 1: MultiRec $(U, I, B, E, \mathcal{R})$

input : The set of users U, set of items I, batch size B, number of epochs E, and all relations matrices $\mathcal{R}$
output: The predicted values $\hat{y}_B$, $\hat{y}_S$ and $\hat{y}_P$ for the missing entries in all relations in $\mathcal{R}$

Initialize model parameters Θ
for *E epochs* **do**
    for $R \in \mathcal{R}$ **do**
        Draw $B/2$ users
        for *each user* **do**
            if $R == R_S$ **then**
                Draw (u,i) pair from $\mathcal{Y}_R^+$
            else
                Draw (u,i) pair from $\mathcal{Y}_R^+$ and another (u,i) pair from $\mathcal{Y}_R^-$
            end
        end
        $z_u \leftarrow f_U(x_u; \theta_U)$
        $z_i \leftarrow f_I(x_i; \theta_I)$
        $\hat{y}_R \leftarrow g_R(z_u \odot z_i; \theta_R)$
        Update θ_U, θ_I and θ_R by minimizing $\mathcal{L}(\Theta)$
    end
end

5 EXPERIMENTS

In this section, multiple experiments were conducted to evaluate the performance of MultiRec and to answer the following research questions.

RQ1 How well does MultiRec perform compared to the state-of-the-art recommender system models on auction-based settings?
RQ2 What is the impact of adding the auxiliary relational information?
RQ3 What is the effect of using different relations loss weights on MultiRec performance?

5.1 Datasets

In order to evaluate the performance of MultiRec against the state-of-art models on auction-based settings, we used the following datasets.

(1) VWFS: This Business-to-business (B2B) proprietary dataset was collected by Volkswagen Financial Services used-cars center during the period between 2015 and 2019. The dataset contains historical used-cars purchase and bidding records done by multiple brands and independent dealers. Usually, in such settings, dealers bid and purchase multiple used-cars

directly from Volkswagen Financial Services. Then they offer them again to their private customers with some added profit margin.

(2) eBay [1]: A publicly available auction-based dataset collected from eBay. The dataset has similar characteristics to the VWFS dataset and contains auction information on Cartier wristwatches, Palm Pilot M515 PDAs, Xbox game consoles, and Swarowski beads.

Our main focus throughout the experiments will be on the VWFS datasets. However, to promote reproducibility, we utilized the eBay dataset in the comparative study to demonstrate the generalizability of the proposed model across other datasets. Table 1 presents a summary of the most important statistics of the datasets used for validating the experimental performance of MultiRec. These real world auction datasets differ considerably across dimensions and serve to demonstrate the model's unbiased performance against different data generating processes in the area.

5.2 Evaluation Protocol

To evaluate the proposed MultiRec model on the item recommendation task, we employed the widely used leave-one-out evaluation protocol [7]. For each user, we held out his latest two interactions for validation and testing, while the rest are used for training. In the evaluation phase, the test item is ranked among 99 randomly sampled items that are not interacted by the user. To avoid violating the chronological order of instances in the VWFS dataset, the sampled 99 negative instances are composed of vehicles that were available at auctions in the same month. Finally, the ranked list is truncated at different threshold values K and its quality is judged by the Hit-Ratio (HR) and the Normalized Discounted Cumulative Gain (NDCG) [7] metrics. Both metrics are calculated separately for each test user, and we use the average NDCG and HR as the final performance metrics for the model.

To measure the statistical significance of the reported results, we repeat this process five times with different random weights initialization, and we report the average metrics across all runs along with the p-values of a paired t-test.

The optimal hyper-parameters for MultiRec and other baselines have been estimated via grid search on the validation set. More details about the hyper-parameters and the reproducibility of the experiments will be discussed in detail in section 5.6.

5.3 Performance comparison with state-of-the-art item recommendation models (RQ1)

In this experiment, we compare the performance of MultiRec against multiple state-of-the-art models for implicit feedback item recommendation. Traditional recommender system baselines such as auto-encoder models [14], matrix factorization models [10, 17, 21], and neighborhood-based models were not included in this experiment because they were not applicable to the auction-based settings as they rely on the similarity between user's purchase records which are inherently unique in such settings.

[1]https://www.kaggle.com/onlineauctions/online-auctions-dataset/data#auction.csv

5.3.1 Baselines.

(1) **Random**: A simple baseline model that ranks items randomly.
(2) **TopPopular**: A naive baseline model that ranks used-cars based on the number of sales records of their model type. This baseline is only applied to the VWFS dataset because there was no similar feature that can be used for indicating popularity in the eBay dataset.
(3) **DSSM** [8]: An attribute-aware deep learning recommender model that relies on optimizing the cosine similarly between user and item latent features extracted by deep feedforward networks. To use DSSM in auction-based settings, we used the concatenation between the user's one-hot encoded vector and the user's attributes vector as one input feature vector for the user. On the other hand, we only used the item's attributes vector as an input feature vector for the item.
(4) **DeepFM** [5]: State-of-the-art model for click-through rate prediction that relies on learning low and high order feature interactions using an ensemble of Factorization Machines (FM) and deep feedforward networks. To use DeepFM, we concatenated the user's one-hot encoded vector, the user's attributes vector, and the item's attributes vector into one big input vector.
(5) **AANeuMF**: This is an extended attribute-aware version of the state-of-the-art NeuMF model [7] that merges matrix factorization and deep neural networks for item recommendation in implicit feedback settings. Given that the original NeuMF architecture relied mainly on user's, and item's ids, we modified it such that it takes into account the users' attributes vectors along with their original one-hot encoded input vectors, and we completely replaced the items' one-hot encoded input vectors with their attributes vectors.
(6) **Graph2R**: This is an extended version of the state-of-the-art attribute-aware GraphRec [18] model for item recommendation in implicit feedback settings. GraphRec relies on co-embedding user and item attributes through non-linear embedding layers with CReLU activation. To extend the model for ranking items in implicit feedback auction settings, we replaced the means squared error (MSE) loss function with a logistic loss function, and we omitted the items' one-hot encoded input vectors. One important aspect of the extended GraphRec is that it could not use the Graph-Features because the adjacency matrix rows will be unique across items and users.

We emphasize that for all models except TopPopular and Random, we used the same training protocol and sampling strategy.

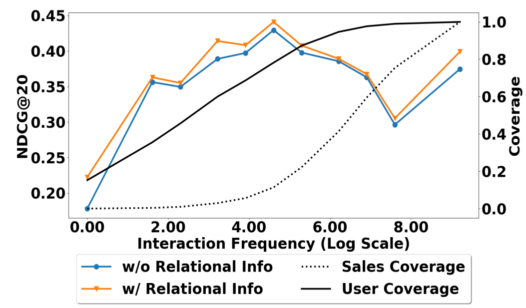
5.3.2 Results. Tables 2 and 3 show that training MultiRec on auxiliary relational information such as the sale price and previous bidding interactions can improve the prediction performance significantly across different threshold values K. MultiRec trained on all auxiliary relational data achieves up to 10.9% improvements in HR and NDCG over the next best baseline on VWFS, and it achieves up to 6.5% improvements over the basic model that was not trained on the relational data.

Table 1: Datasets Statistics

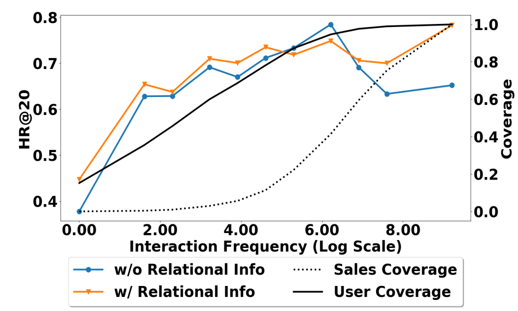
Dataset	Users	Items	Purchases	Biddings	Purchases Density (%)	Biddings Density (%)	User Features	Item Features	Unique Items (%)
VWFS	3220	269,104	269,104	375,349	0.031	0.043	7750	572	99.999%
eBay	3388	626	626	10,055	0.029	0.474	2	4	35.943%

On the eBay dataset, MultiRec with auxiliary relational data achieves significantly higher improvements over the next best baseline, with percentages ranging between 31% and 135.7%. It also achieves up to 44% improvements over the basic model that was not trained on the relational data. Such high improvement percentages on the eBay dataset are mainly due to the fact that eBay bidding relation had 16 times more density than the main purchase relation, which means we have 16 times more information to learn from bidding than from purchase interactions alone. On the other hand, the difference between the two relations densities was much smaller in the VWFS dataset, which explains the lower improvement percentages.

Results also show that most of the lift in performance is achieved by incorporating the bidding relation, and a slight lift can still be achieved by training on the sale price. This is due to the fact that bidding relation have much more implicit interactions between users and items compared to the purchase and sale price relations.



(a) NDCG



(b) HitRatio

Figure 2: Impact of the relational information on MultiRec performance across different users' interaction frequencies on the VWFS dataset.

5.4 Impact of using auxiliary relational information on MultiRec performance (RQ2)

To further study the impact of using auxiliary relational information during the training process, we compared the impact of using this information on the HR@20 and NDCG@20 of different users who had a different number of purchase interactions in the training phase.

Figure 2a shows that training on the auxiliary relational data provides a lift for all users in terms of NDCG with a slightly higher impact on users with the fewest number of interactions and on the noisy users with an extremely high number of interactions. On the other hand, Figure 2b shows that the positive impact on the HitRatio affects 80% of the users whom all cover 60% of the sales records. It also shows a similar trend to the NDCG, where there is a higher impact on noisy users and users with the least number of interactions.

5.5 Impact of different relations' weights (RQ3)

In this section, we study the sensitivity of MultiRec's performance with respect to the relation weights by analyzing the MultiRec performance using a fixed α_P of 1.0 and different values of α_B and α_S. Figure 3 shows that the relation weights are crucial and have a more significant effect on the MultiRec's performance on the eBay dataset than on VWFS. On the VWFS dataset, the model performance is relatively stable with respect to the bidding relation weight α_B while lower weight values for the sale relation provide better overall performance than higher values.

5.6 Reproducibility of the Experiments

The source code of MultiRec is available at our GitHub repository[2]. The best found hyper-parameters are indicated in Tables 4 and 5. We used ADAM optimizer with weight decay $\beta = 0.00001$ [15] and batch sizes of 1000 and 100 for the VWFS and eBay datasets respectively. We tuned the hyper-parameters using a grid-search as follows:

- MultiRec: We tested different number of embedding and scoring layers of [1, 2, 3, 4] with sizes that range from 5 to 150, learning rates of [0.0005, 0.0001, 0.00001, 0.00002, 0.00003, 0.00006, 0.00008, 0.000001], α_S of [0.01, 0.001, 0.0001, 0.00008, 0.00006, 0.00004, 0.00002, 0.00001, 0.000001], α_B of [0.0001, 0.001, 0.005, 0.01, 0.05, 0.1 ,0.5, 1.0, 10.0], and activation functions [Linear, Leaky_ReLU, Relu and CReLU].
- AANeuMF: We tested layers numbers of [1, 2, 3, 4], prediction factors of [8, 16, 32, 64], and learning rates of [0.0005, 0.0001, 0.00001, 0.00002, 0.00003, 0.00006, 0.00008, 0.000001]

[2] https://github.com/ahmedrashed-ml/MultiRec

Table 2: Performance comparison on the VWFS dataset

Model	HR@10	HR@20	HR@50	NDCG@10	NDCG@20	NDCG@50
Random	0.095	0.193	0.520	0.041	0.066	0.129
TopPopular	0.302	0.388	0.628	0.238	0.260	0.307
DSSM [8]	0.445	0.577	0.792	0.272	0.305	0.348
DeepFM [5]	0.461	0.606	0.836	0.290	0.326	0.372
AANeuMF	<u>0.480</u>	<u>0.627</u>	<u>0.836</u>	<u>0.299</u>	<u>0.336</u>	<u>0.377</u>
Graph2R	0.470	0.596	0.825	0.285	0.317	0.363
MultiRec	0.492	0.635	0.843	0.311	0.347	0.389
MultiRec + Sale Price	0.497	0.636	0.838	0.316	0.351	0.391
MultiRec + Bidding	0.510	0.650	0.859	0.324	0.359	0.401
MultiRec + Bidding + Sale Price	**0.518***	**0.657***	**0.862***	**0.331***	**0.366***	**0.407***
Improvement over best baseline (%)	7.84	4.69	3.05	10.91	9.12	7.87

(*) Significantly outperforms the best baseline at the 0.01 levels.

Table 3: Performance comparison on the eBay dataset

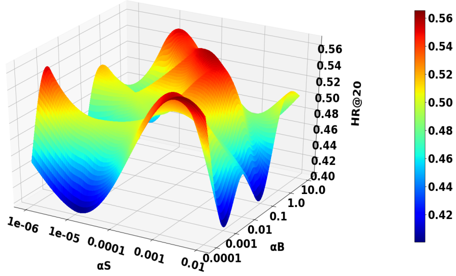
Model	HR@10	HR@20	HR@50	NDCG@10	NDCG@20	NDCG@50
Random	0.120	0.270	0.540	0.057	0.094	0.148
DSSM [8]	0.150	0.300	0.580	0.069	0.107	0.162
DeepFM [5]	<u>0.190</u>	0.300	0.550	<u>0.077</u>	0.103	0.152
AANeuMF	0.160	0.340	0.600	0.068	0.112	0.156
Graph2R	0.170	<u>0.380</u>	<u>0.700</u>	0.067	<u>0.119</u>	<u>0.183</u>
MultiRec	0.290	0.570	0.840	0.109	0.177	0.227
MultiRec + Sale Price	0.270	0.560	0.830	0.103	0.173	0.225
MultiRec + Bidding	**0.420***	0.600	0.910	**0.160***	**0.204***	**0.257***
MultiRec + Bidding + Sale Price	0.400	**0.610***	**0.920***	0.158	0.202	0.256
Improvement over best baseline (%)	135.29	60.53	31.43	135.75	70.24	39.99

(*) Significantly outperforms the best baseline at the 0.01 levels.

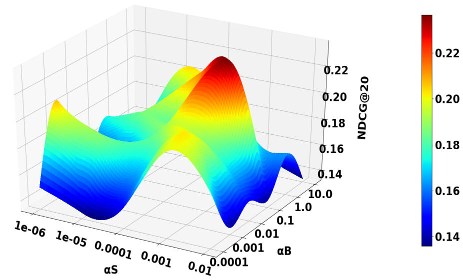
Table 4: Best found hyper-parameters on VWFS dataset

Model	Embedding layers	g_P	g_B	g_S	Activation Function	α_P	α_B	α_S	lr	Epochs
MultiRec	[60]	Dot product	[5]	[5]	Leaky_ReLU for Embedding and Scoring Functions	1.0	0.5	8×10^{-5}	1×10^{-5}	495
AANeuMF	[128,64,32,16]	-	-	-	ReLU	-	-	-	2×10^{-5}	620
DSSM	[60,30]	-	-	-	Tanh	-	-	-	2×10^{-5}	20
Graph2R	[60]	-	-	-	CReLU	-	-	-	1×10^{-5}	620
DeepFM	FM : [32] FFN :[32,32]	-	-	-	ReLU	-	-	-	1×10^{-5}	110 +

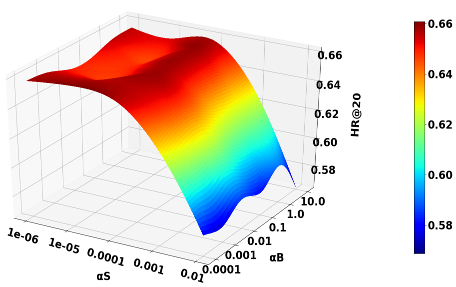
- DSSM: We tested layers numbers of [1, 2, 3, 4] with sizes that range from 5 to 150, and learning rates of [0.0005, 0.0001, 0.00001, 0.00002, 0.00003, 0.00006, 0.00008, 0.000001].
- DeepFM: We tested different embedding sizes of [8, 16 , 32, 40], layers numbers of [1, 2, 3, 4] with sizes that range from 5 to 50, and learning rates of [0.0005, 0.0001, 0.00001, 0.00002, 0.00003, 0.00006, 0.00008, 0.000001]
- Graph2R: We tested different embedding sizes that range from 5 to 150, and learning rates of [0.0005, 0.0001, 0.00001, 0.00002, 0.00003, 0.00006, 0.00008, 0.000001]



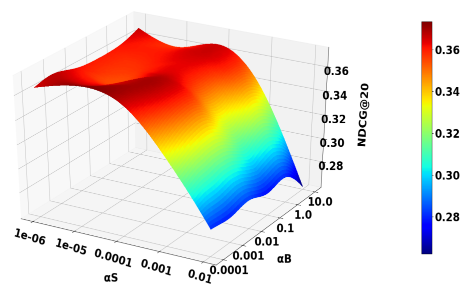
(a) eBay HR@20



(b) eBay NDCG@20



(c) VWFS HR@20



(d) VWFS NDCG@20

Figure 3: Sensitivity analysis on the different relations weights values α_R

Table 5: Best found hyper-parameters on eBay dataset

Model	Embedding layers	g_P	g_B	g_S	Activation Function	α_P	α_B	α_S	lr	Epochs
MultiRec	[85,60]	Dot product		[5]	ReLU for Embedding and Linear for Scoring Functions	1.0	0.5	8×10^{-5}	3×10^{-5}	2000
AANeuMF	[128,64,32,16]	-	-	-	ReLU	-	-	-	3×10^{-5}	2000
DSSM	[30]	-	-	-	Tanh	-	-	-	3×10^{-5}	385
Graph2R	[60]	-	-	-	CReLU	-	-	-	3×10^{-5}	2000
DeepFM	FM : [8] FFN :[32,32]	-	-	-	ReLU	-	-	-	3×10^{-5}	110

6 CONCLUSION AND FUTURE WORK

In this paper, we propose MultiRec, a simple non-linear multi-relational recommender system model that leverages user and item attributes along with the auxiliary relational information that generally exists in most auction settings such as biddings and sale prices. Experimental results on two real-world auction datasets show that MultiRec significantly outperforms state-of-the-art attribute aware models on the task of item recommendation in auction-based settings. Results also showed that auxiliary relational information could significantly improve the model performance across all users regardless of having different interaction frequencies. In future works, we plan to extend this model for context and session-based prediction by incorporating the time factor and external domain knowledge about users. We also plan to extend the model capacity by further exploring new possible relations between user and items as we noted before.

ACKNOWLEDGMENTS

This work is co-funded by the industry project "Data-driven Mobility Services" of ISMLL and Volkswagen Financial Services.(https://www.ismll.uni-hildesheim.de/projekte/dna_en.html)

REFERENCES

[1] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation

with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 335–344.
[2] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*. 278–288.
[3] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *2010 IEEE International Conference on Data Mining*. IEEE, 176–185.
[4] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2015. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.
[5] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).
[6] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 355–364.
[7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.
[8] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2333–2338.
[9] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 233–240.
[10] 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*. 426–434.
[11] Sheng Li, Jaya Kawale, and Yun Fu. 2015. Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 811–820.
[12] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. CCCFNet: A content-boosted collaborative filtering neural network for cross domain recommender systems. In *Proceedings of the 26th international conference on World Wide Web companion*. 817–818.
[13] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1754–1763.
[14] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*. 689–698.
[15] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).
[16] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 43–52.
[17] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.
[18] Ahmed Rashed, Josif Grabocka, and Lars Schmidt-Thieme. 2019. Attribute-aware non-linear co-embeddings of graph features. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 314–321.
[19] Ahmed Rashed, Josif Grabocka, and Lars Schmidt-Thieme. 2019. Multi-Relational Classification via Bayesian Ranked Non-Linear Embeddings. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1132–1140.
[20] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.
[21] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
[22] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. 253–260.
[23] Alessandro Suglia, Claudio Greco, Cataldo Musto, Marco De Gemmis, Pasquale Lops, and Giovanni Semeraro. 2017. A deep architecture for content-based recommendations exploiting recurrent neural networks. In *Proceedings of the 25th conference on user modeling, adaptation and personalization*. 202–211.
[24] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-pointer co-attention networks for recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2309–2318.
[25] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1235–1244.
[26] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.
[27] Xinxi Wang and Ye Wang. 2014. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*. 627–636.
[28] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. 153–162.
[29] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.
[30] Shuai Zhang, Lina Yao, and Xiwei Xu. 2017. AutoSVD++ An Efficient Hybrid Collaborative Filtering Model via Contractive Auto-encoders. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 957–960.