



RecNet: A Resource-Constraint Aware Neural Network for Used Car Recommendation

Haihua Shi^{1,2} · Jianjun Qian² · Nengjun Zhu³ · Tong Zhang² · Zhen Cui² · Qianliang Wu² · Shanshan Feng⁴

Received: 3 June 2022 / Accepted: 17 October 2022
© The Author(s) 2022

Abstract

Resource constraints, e.g., limited product inventory or financial strength, may affect consumers' choices or preferences in some recommendation tasks but are usually ignored in previous recommendation methods. In this paper, we aim to mine the cue of user preferences or intention in the case of resource-constraint recommendation tasks. For this purpose, we specifically build a largely used car transaction dataset possessing resource-constraint characteristics. Accordingly, we propose a resource-constraint-aware network to predict the user's future interaction based on dynamic connections between users and items. To describe the user-item connection dynamically, mutually recursive recurrent neural networks (MRRNNs) are introduced to capture long-term interactive dependencies, and effective representations of users and items are obtained. To further consider the resource constraint, a resource-constraint branch is built to explore resource variation's influence on user preferences. Finally, mutual information is introduced to measure the similarity between the future user action and fused historical behavior features to predict future interaction. The fused features come from both MRRNNs and resource-constraint branches. We test the performance on the built used car transaction dataset and the Tmall dataset, and the experimental results verify the effectiveness of our framework.

Keywords Recommender system · Resource constraint · Recurrent neural networks(RNNs) · Mutual information

Abbreviations

RNN Recurrent neural networks (RNNs) are a class of neural networks that allow previous outputs to be used as inputs while having hidden states

MRRNN Mutually-recursive recurrent neural networks (MRRNNs) means that two RNN networks take each other's output as input and update the network alternately

CNN A convolutional neural networks (CNNs) is a Deep Learning algorithm which can take in an input image, assign parameters (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other

✉ Nengjun Zhu
zhu_nj@shu.edu.cn

✉ Tong Zhang
tong.zhang@njust.edu.cn

Haihua Shi
shihaihua@jit.edu.cn

Jianjun Qian
csjqian@njust.edu.cn

Zhen Cui
zhen.cui@njust.edu.cn

Qianliang Wu
wuqianliang@njust.edu.cn

Shanshan Feng
fswl6869@foxmail.com

¹ Jinling Institute of Technology, Nanjing 210094, People's Republic of China

² Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, Nanjing 210094, People's Republic of China

³ School of Computer Engineering and Science, Shanghai University, Shanghai 200444, People's Republic of China

⁴ School of Information Science and Engineering, Shandong Normal University, Jinan 610101, People's Republic of China

NDCG	Normalized discounted cumulative gain (NDCG) is a ranking metric. In the context of sequential recommendation, it is formulated as $NDCG = \frac{1}{\log_2(1+rank_{pos})}$, where $rank_{pos}$ is the rank of correctly predicted items
GRU	Introduced by [44], Gated Recurrent Unit (GRU) aims to solve the vanishing gradient problem which comes with a standard recurrent neural network (RNN)
NCE	Noise contrastive estimation (NCE) [39] is a powerful parameter estimation method for log-linear models, which avoids calculation of the partition function or its derivatives at each training step, a computationally demanding step in many cases. It is closely related to negative sampling methods
MSE	In statistics, the mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value

1 Introduction

Nowadays, recommender systems have become key tools for users to filter information and locate their preferences on e-commercial websites. Based on browsing and transaction records of users during online shopping, recommender systems aim to capture users' interests accurately and further recommend multiple products that users may be interested in. In this way, recommender systems effectively promote the online shopping experience by facilitating users to find interesting products and creating opportunities for the e-commercial website to increase revenue.

Numerous algorithms have been proposed for recommender systems in previous literature [1–15] and achieved great success. However, there are still various challenges existing in real-world applications. One crucial problem is the significant impact of resource constraint on user interests, which is always ignored in previous works. In general, resource constraint exists in products and users, e.g., Inventory constraint and Limited financial strength. For instance, after purchasing valuables, in one period, users' desire to buy other high-end products may decrease due to financial limitations. Also, consumers may turn to other sub-optimal products when the most interesting ones are out of inventory. Therefore, it is non-trivial to consider the resource constraint in designing recommendation models, especially to appropriately pick out resource

constraint-related information and further model them with effective algorithms. However, there are multiple issues to be tackled in this process, including two crucial ones:

- (i) Suitable datasets. Existing recommendation datasets usually focus on recording user action and product attributes but often ignore the collection of resource constraint-related information. This makes them hardly be employed to investigate resource-constraint aware recommendation tasks.
- (ii) Effective inference of the intricate system. The complicated information of resource constraints, the dynamics of user-item interaction, and the similarity between each pair of user and product indicating the user preference should be well represented and further modeled in the designed framework.

Based on the analysis above, in this paper, we investigate the resource-limited recommendation as a sequential recommendation task and propose a Resource-constraint Aware Neural Network (denoted as RecNet for short) to mine the cue of the user's preferences for products. First, we construct a large recommendation dataset named 'UsedCars' about used car transactions with resource-limitation characteristics. As a particular recommendation task, used car recommendation based on the 'UsedCars' dataset well meets the resource-limited situation from two main aspects: (1) limited supply of products: obviously, the inventory of each product is not constant and restricted by those used car owners; (2) deep pocket requirement: the high price of used cars makes it unrealistic for users, even for some companies, to frequently make orders just like buying daily products. Then, an interest-behavior multiplicative network is constructed to model the system. Considering dynamic user-item connections, we introduce the mutually-recursive recurrent neural networks (MRRNNs) to capture the long-term interactive dependencies of users and items and extract their high-level features. To model the resource constraint, we build another resource-limited branch to explore the influence of resource variation caused by user behavior. Specifically, an RNN branch is employed to learn the representation from the user and product status variation once a transaction happens. Finally, both MRRNNs and resource-limited branch features are fused and further used to measure the similarity between user interest and the product. Specifically, mutual information is introduced for user-product similarity measurement in an unsupervised manner. To optimize the entire framework, supervised and unsupervised losses are calculated based on which the parameters are tuned through back-propagation. In our experiments, the performance is tested on the built 'UsedCars' dataset

and the Tmall dataset, and the results verify the effectiveness of our proposed framework.

To summarize, in this paper, our contributions are four-fold:

- (i) **A new used car dataset:** We construct a new large-scale used car dataset based on the records of an online shopping platform. This is the first dataset for the used car recommendation task with resource-limitation characteristics.
- (ii) **Resource-constraint modeling:** We specifically investigate the resource-limitation problem, which is always ignored in previous works. Accordingly, we propose the novel RecNet to model the intricate recommender system by using a special resource-limited branch to capture the influence of resource variation on user interests.
- (iii) **Similarity measurement:** We introduce mutual information to measure the similarities between users' history and future targets. Considering the informative representation of items and users, mutual information may be more suitable in the similarity measurement as it considers the associated distribution of the representation.
- (iv) **Effectiveness:** We test the proposed framework on our built 'UsedCars' and Tmall datasets. The experimental results verify the effectiveness of our proposed RecNet.

This paper is structured as follows: Sect. 2 survey the related works and Sect. 3 briefly describes the problem definition. Section 4 describes the proposed method in detail. Section 5 describes and analyzes the results of the experiment. The conclusion is drawn in Sect. 6.

2 Related Work

In the early stage of the recommender system development, recommendation algorithms mainly include the Markov chain [16, 17], and matrix factorization [3, 4, 7, 18]. Factorization-based approaches decompose user-item interaction matrix built from users' feedback to obtain low-rank embedding of users/items, then make matrix completion or subsequent predictions via the inner product of user and item embedding vectors. However, these methods do not consider the time order of interactions. Therefore, they are not suitable for sequential recommendation scenarios.

Recently, deep neural network models, e.g., Recurrent Neural Networks(RNNs) and Convolutional Neural Networks(CNNs), have achieved significant success

compared to conventional methods. RNNs methods, based on the original intention of their design, have become the most successful ones in sequence recommendation [19]. Several RNN variants [19–23] have been proposed to adapt to different application scenarios. On the other hand, considering the tremendous success of the CNN model in computer vision, there have been various CNN-based sequential recommendation models [14, 24, 25] proposed. As the attention mechanism achieved promising performance in NLP [26–28], many attention-based sequential models [29–31] have also been developed. In particular, the self-attention model [2] showed powerful ability in the sequential recommender system of daily consuming e-commerce. Most of them, seek to learn effective representation based on a suboptimal prior or capture similarities between interacted items in one period. However, few attempts have been made to model the resource constraints.

In information theory, mutual information is a kind of measurement for statistic dependence between two random variables X, Y with $p(x), p(y)$ representing their associated probability distributions, respectively [32]. By leveraging domain knowledge, mutual information estimation has been successfully applied to a variety of problem areas, including image processing [33–35], video classification [36], and natural language understanding [37]. The first application of mutual information in the recommendation task is [38], which deploys mutual information to measure similarities between items' ratings of users in collaborative filtering.

3 Problem Definition

The symbols in this paper are defined as follows. The sets of users and items are denoted by \mathcal{U} and \mathcal{I} , respectively. $X = [x_1, x_2, \dots, x_T]$ is a time-ordered sequence of temporal user-item interactions and $x_t = (u_t, i_t, t), t \in [1, T]$. Here, x_t denotes the user-item interaction between the user u_t and item i_t at timestamp t . We define the sequential recommendation problem as: given the interaction history of the user u , we aim to predict the probability of the user-item interaction denoted as $P(x_{T+1} \| x_1, x_2, \dots, x_T)$ at time $T + 1$.

4 The Proposed Model

In this section, we first overview the entire architecture of our proposed model and then introduce the involved key modules in detail, including MRRNNs, the resource constraint branch, the mutual information estimator, and the loss function. Table 1 lists symbols used in the following sections.

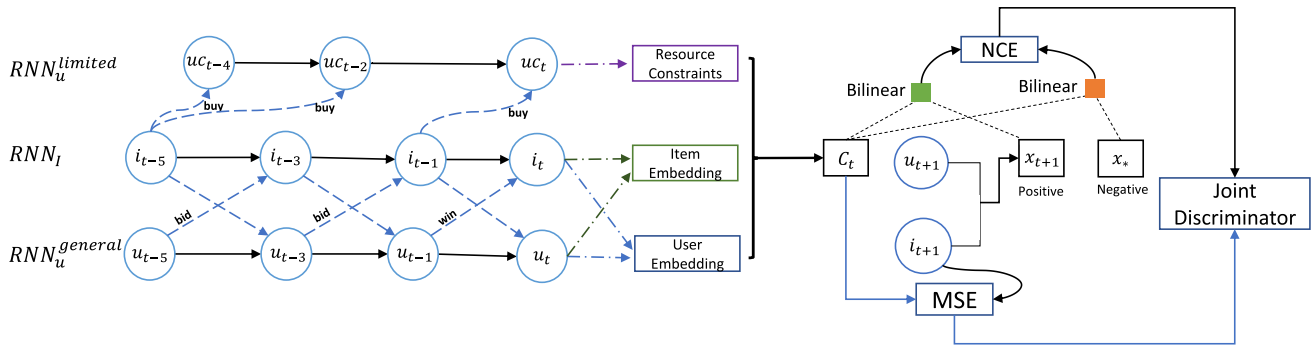


Fig. 1 Illustration of our model architecture. ‘NCE’ are the contrastive loss function built by the mutual information between history context and future item targets. ‘MSE’ are Mean Squared Error between prediction and ground truth. $RNN_U^{limited}$, $RNN_U^{general}$ and RNN_I are

resource constraint embedding, user and item embedding, respectively. The joint discriminator consists of two metric distances (MSE and mutual information) and is used in the validation and testing stage to make a composite ranking in all candidate items

Table 1 Symbols used in this paper

Symbol	Meaning
$\mathbf{h}_u^g(t)$ and $\mathbf{h}_i(t)$	General dynamic embedding of user u and item i at time t
\bar{u} and \bar{i}	Static embedding of user u and item i
$\tilde{\mathcal{J}}(t)$	Predicted item embedding
$\mathbf{h}_u^l(t)$	Resource constraint user dynamic embedding
c_t	History context including user and item embedding
f_i	Density ratio function
$\bar{\mathcal{J}}$	A one-hot item embedding of target item to be clicked or purchased
v_t/v_p	Feature vector of a car

4.1 Method Overview

The whole structure of our proposed model is illustrated in Fig. 1. $RNN_U^{limited}$ denotes the resource constraint embedding branch while $RNN_U^{general}$ and RNN_I construct the mutually recursive recurrent neural networks. Taking user interactions as inputs, we exploit $RNN_U^{general}$ and RNN_I to learn the user and item dynamic embeddings (i.e., $\mathbf{h}_u^g(t)$ and $\mathbf{h}_i(t)$ in Sect. 4.2). We leverage $RNN_U^{limited}$ to capture the resource constraints information (i.e., $\mathbf{h}_u^l(t)$ in Sect. 4.3) once a transaction happens. Then, the updated features of both user and item and the resource constraints information are fused and used to measure the mutual information between user’s historical behavior context and future candidate items (see Personal Mutual Information in Sect. 4.4). Finally, several loss functions (see Sect. 4.5), including one supervised MSE loss, one unsupervised NCE loss, and two regularization losses are used to optimize the whole network. In the inference stage (i.e., validation and test), we make item predictions using a Joint Discriminator consisting of two metrics (see Sect. 5.2.2).

4.2 Mutually-Recursive Recurrent Neural Networks

To learn the dynamic connection between users and items, our model leverages the time-ordered interaction $S_t = (u_t, i_t, \mathbf{v}_t)$ between user u and item i at time t to generate two mutually recursive RNN embedding $\mathbf{h}_u(t)$ and $\mathbf{h}_i(t)$:

$$\mathbf{h}_u^g(t) = \sigma(\mathbf{W}_1^u \mathbf{h}_u^g(t-1) + \mathbf{W}_2^u \mathbf{h}_i(t-1) + \mathbf{W}_3^u \Delta_u + \mathbf{W}_4^u \mathbf{v}_{t-1}), \tag{1}$$

$$\mathbf{h}_i(t) = \sigma(\mathbf{W}_1^i \mathbf{h}_i(t-1) + \mathbf{W}_2^i \mathbf{h}_u^g(t-1) + \mathbf{W}_3^i \Delta_i + \mathbf{W}_4^i \mathbf{v}_{t-1}), \tag{2}$$

where Δ_u denotes the time elapsed since user u ’s last interaction (with any item) and Δ_i is the time elapsed since the last interaction (with any user) of item i . The matrices $\mathbf{W}_1^u, \mathbf{W}_2^u, \mathbf{W}_3^u, \mathbf{W}_4^u$ are the parameters of $RNN_U^{general}$ and $\mathbf{W}_1^i, \mathbf{W}_2^i, \mathbf{W}_3^i, \mathbf{W}_4^i$ are the parameters of RNN_I . \mathbf{v}_{t-1} is the interaction feature vector, which involves the information of both image features and car attributes. $\sigma(\cdot)$ is the sigmoid function to introduce non-linearity. These two embeddings learn the long-life user preference from the different kinds of interactions. The mutually recursive update of two embeddings is performed along the timeline.

4.3 The Resource Constraint Branch

This section introduces a sub-module that explores how to model resource constraint information that affects users' future interactions. There are two resource constraint factors, i.e., product inventory and user financial strength. The user's financial strength would fluctuate with the user status variation, e.g., a user's financial strength would degrade after a recent purchase of an expensive car. Due to sellers' dynamic selling intention, the product inventory may be updated daily. For instance, if the seller withdraws the on-sale product, one specific item category would become limited or out of inventory (e.g., a seller may turn to other online selling platforms for a better price).

Accordingly, we introduce a resource constraint branch to obtain user's resource constraint embedding, denoted as $\mathbf{h}_u^l(t)$, once a transaction happens. As the financial strength of users may not be directly accessible in the real application, to calculate $\mathbf{h}_u^l(t)$, we represent the financial strength by multiple related factors which implicitly or indirectly reflect the purchasing power of users. Concretely, financial strength is potentially revealed by the user purchase history (i.e., $\mathbf{h}_u^g(t-1)$), the time interval since the last purchase (i.e., Δ_p), current purchasing product (i.e., \bar{i}_p), and user id (i.e., \bar{u}_p). Similarly, we advocate that the product inventory is implicitly contained in the dynamic embedding (i.e. $\mathbf{h}_u^g(t)$ and $\mathbf{h}_i(t)$) and the decisive factors that how many cars are on sales (i.e., initial inventory I^l) one day.

The user's resource constraint embedding $\mathbf{h}_u^l(t)$ is formulated as a function of these related factors:

$$\begin{aligned} \mathbf{h}_u^l(t) = & \sigma(\mathbf{W}_1^l \mathbf{h}_u^l(t-1) + \mathbf{W}_2^l \mathbf{h}_u^g(t) + \mathbf{W}_3^l \mathbf{h}_i(t) \\ & + \mathbf{W}_4^l \bar{u}_p + \mathbf{W}_5^l \bar{i}_p + \mathbf{W}_6^l \Delta_p \\ & + \mathbf{W}_7^l \mathbf{v}_p + \mathbf{W}_8^l I^l), \end{aligned} \tag{3}$$

where $\mathbf{W}_1^l, \dots, \mathbf{W}_8^l$ are parameters of \mathbf{h}_u^l . \mathbf{v}_p is the feature vector of the purchased car. Item id \bar{i}_p contains product price information.

Since some time has elapsed from the user's last interaction at time t while making predicting future time $t+1$, we employ a projection operation to moderately avoid user embedding's information expiration (e.g., predict one user's preference at current with his latest interaction that happened yesterday). Two inputs are required for the projection operation: the user embedding at time t and the elapsed time Δ .

Specifically, in Eqs. (4,5), We first convert Δ^g (i.e., the time elapsed since last clicked) to a time-context vector w^g using a linear layer (represented by vector \mathbf{W}_t). The projected embedding $\mathbf{h}_u^g(t+\Delta)$ is then obtained by leveraging a temporal attention vector $1+w^g$ scaling the past user embedding $\mathbf{h}_u^g(t)$:

$$w^g = \mathbf{W}_t \Delta^g, \tag{4}$$

$$\mathbf{h}_u^g(t+\Delta) = (1+w^g) * \mathbf{h}_u^g(t), \tag{5}$$

where ' $t+\Delta$ ' means the very current timestamp (i.e. now).

Similarly, we also make the same projection to resource constraint user embedding $\mathbf{h}_u^l(t)$:

$$w^l = \mathbf{W}_t \Delta^l, \tag{6}$$

$$\mathbf{h}_u^l(t+\Delta) = (1+w^l) * \mathbf{h}_u^l(t). \tag{7}$$

To model the historical actions' influence on future targets, we introduce a new combined user embedding $f(\mathbf{h}_u^g(t+\Delta), \mathbf{h}_u^l(t+\Delta))$ which combine (i.e., adding, concatenation, product, etc) the general user embedding and resource constraint user embedding. For simplicity but without loss of generality, we let $f(\mathbf{h}_u^g(t+\Delta), \mathbf{h}_u^l(t+\Delta)) = \mathbf{h}_u^g(t+\Delta) + \mathbf{h}_u^l(t+\Delta)$. The predicting target is a one-hot item embedding \bar{j} to be clicked or purchased. The prediction is made by a fully connected linear layer as following:

$$\begin{aligned} \tilde{j}(t+\Delta) = & \mathbf{W}_1^j * f(\mathbf{h}_u^g(t+\Delta), \mathbf{h}_u^l(t+\Delta)) \\ & + \mathbf{W}_2^j \bar{u} + \mathbf{W}_3^j \mathbf{h}_i(t-1) + \mathbf{W}_4^j \bar{i}(t-1), \end{aligned} \tag{8}$$

where $\mathbf{W}_1^j, \dots, \mathbf{W}_4^j$ are the parameters.

4.4 Personal Mutual Information

To improve representation learning, we propose to maximize the mutual information between the users' historical behavior context c_t and the next-timestamp action x_{t+1} to learn the shared latent space representations between them:

$$I(x_{t+1}; c_t) = \sum_{x_{t+1}, c_t} p(x_{t+1}, c_t) \log \frac{p(x_{t+1}, c_t)}{p(x_{t+1}) p(c_t)}. \tag{9}$$

Inspired by [37], we can model a density ratio $f_1(x_{t+1}, c_t)$ which is proportional to the mutual information between x_{t+1} and c_t :

$$f_1(x_{t+1}, c_t) \propto \frac{p(x_{t+1} \| c_t)}{p(x_{t+1})}, \tag{10}$$

where we define density ratio f_1 as a simple log-bilinear model:

$$f_1(x_{t+1}, c_t) = \exp(x_{t+1}^T \mathbf{W}_1^f c_t). \tag{11}$$

The history context c_t are composed of resource constraint user embedding at the last purchase time (mapping to current

time t), last updated general user embedding (also mapping to current time t), and last interacted item embedding:

$$c_t = \text{cat}[f(\mathbf{h}_u^l(t + \Delta), \mathbf{h}_u^g(t + \Delta), \bar{u}, \mathbf{h}_i(t - 1), \bar{i}(t - 1))]. \quad (12)$$

As we cannot evaluate two distributions $p(x_{t+1})$ and $p(x_{t+1}|c_t)$ directly or computationally intense, when given one positive sample $pos_{x_{t+1}}$ [1] from ‘conditional interaction distribution’ $p(x_{t+1}|c_t)$ and other $N-1$ random negative ones $neg_{x_{t+1}}$ [2 : N] from ‘interaction distribution’ $p(x_{t+1})$, the Noise-Contrastive-Estimation [39] of density ratio f_1 can be computed as follows:

$$pos_{x_{t+1}} [1] = [\mathbf{h}_u^g(t + 1), \mathbf{h}_i(t + 1), \bar{i}(t + 1)], \quad (13)$$

$$neg_{x_{t+1}} [2 : N] = [\mathbf{h}_u^g(t + 1), \mathbf{h}_i(t), \bar{i}(t)], \quad (14)$$

$$pn_{x_{t+1}} = \begin{bmatrix} pos_{x_{t+1}} \\ neg_{x_{t+1}} \end{bmatrix}, \quad (15)$$

$$f_1(pn_{x_{t+1}}, c_t) = \exp(pn_{x_{t+1}} * \mathbf{W}_1^f * c_t). \quad (16)$$

4.5 Loss Functions

We employ a joint loss consisting of an MSE(mean-squared error) Loss and an NCE(Noise-Contrastive-Estimation) Loss to train the network. The MSE loss is used to minimize the sum of L_2 distance between the predicted item and the ground truth item:

$$L_{MSE} = \sum_{(u,j,t) \in S} \|\tilde{J}(t + \Delta) - \bar{J}\|_2, \quad (17)$$

where \tilde{J} and \bar{J} are predicted and ground truth items, respectively.

Given one N samples set $X = x_1, \dots, x_n$ which contains one positive sample from distribution $p(x_{t+1}|c_t)$ and $N-1$ negative samples from distribution $p(x_{t+1})$, we define the NCE loss as:

$$L_{NCE} = -E_X \left[\log \frac{f_1(pos_{x_{t+1}}, c_t)}{\sum_{pn_{x_j} \in X} f_1(pn_{x_j}, c_t)} \right]. \quad (18)$$

Optimizing L_{NCE} is equivalent to estimating the density ratio in Eq (10). By combining the above two losses, we get final loss functions:

$$Loss = \lambda_m L_{MSE} + \lambda_n L_{NCE} + \lambda_U \sum_u \|\mathbf{h}_u^g(t) - \mathbf{h}_u^g(t - 1)\|_2 + \lambda_I \sum_i \|\mathbf{h}_i(t) - \mathbf{h}_i(t - 1)\|_2, \quad (19)$$

where $\lambda_n, \lambda_m, \lambda_U, \lambda_I$ are hyper parameters. The last two regularization terms we add ensure the general interesting ‘slow

drift” phenomenon [40] that the embedding of instance (user and item) drifts at a relatively slow rate when the model tends to converge. To verify the ‘slow drift’ phenomena, we remove two regular terms and find that the performance of our fusion model descends a lot.

The fusion strategy needs to be carefully designed during training for the different datasets since each dataset has its specific business characteristics. We treat the fusion loss as a joint training multi-task object, and the RNN embedding can be substituted by Gated Recurrent Unit(GRU) or enhanced by self-attention. We leave these improvements in future work.

5 Experiments

This section conducts experiments on two commercial datasets respectively and compares our model RecNet with five well-known baselines of sequential recommendation.

5.1 Dataset Description

The first dataset, ‘UsedCars’, is built from China’s biggest online used car auction website. This website is a used car auction website where each car will be sold to the buyer within one week. The quantity and categories of commodities on the goods shelves change daily. The average price of commodities is more than 10,000 dollars. Most users in the ‘UsedCars’ dataset are used car sales enterprises who come to the platform every day and click product pages to obtain market information and search for new products. This yields many click/transaction sequences. For each user, we adopt all of their interaction data without filtering. This dataset has three user actions: bid, win, and trade. We treat trade actions as ‘purchase’ interactions while the other two kinds of actions as ‘click’. We obtain the image feature by vgg16 [41] pre-trained model, while other features use id embedding.

The second data set is extracted from Tmall, the largest B2C(business-to-consumer) electronic commerce platform in China. It is a dataset obtained from Tmall/Koubei IJCAI16 Contest¹(Tmall for short). This data set comes from a discount coupon APP, from which users mainly order consumption discounts. Therefore, there are two behaviors in the dataset: click and purchase.

Furthermore, to make our model a general framework for the e-commercial purchase-aware sequential recommendation, we only adopt two of the dataset’s dimensions, ‘user’ and ‘item’. The statistics of these two datasets are summarized in Table 2.

¹ <https://tianchi.aliyun.com/dataset/dataDetail?dataId=53>

Table 2 Dataset statistics

Dataset	#Users	#Items	#Actions	Time span	Purchase ratio	Price (CNY)
UsedCars	57,000	40,820	18,700,000	2017.1–2020.10	2.42%	150k
Tmall/Koubei 2016	1,160,135	2,353,208	44,528,128	2015.7–2015.11	24.5%	1k

5.2 Experimental Settings

5.2.1 Training Settings

The datasets are split by time order for all compared methods. The two datasets are split as 80%/10%/10% for train/validation/test. Dimensions of the dynamic embedding of all models are set to 128. We use Adam optimizer for training, with a learning rate of 10^{-3} and batch size of 256. All models run 50 epochs and report test results corresponding to the best-performing validation set. Following [42], our models use t-batch [42] for training data mini-batch.

5.2.2 Implementations of Our Method

The hyper parameter N in Eq. (14) is searched in $\{32, 64, 128, 256, 320\}$ and finally set N to 128, resulting in the best performance. As can be concluded from section 2.3 in CPC [37], the mutual information between the context c_t and target x_{t+k} becomes tighter as N becomes larger. During tuning, we found that the best N is proportional to the size of the datasets. The other two hyper parameters λ_m and λ_n can be tuned according to the characteristics of the datasets.

Since the huge number of users and items will lead to a large number of parameters of bilinear parameter W_1 in Eq. (11), two id compression layers for user and item static one-hot encodings are added. These two compression embeddings' input and output dimensions are $[N_{user}, 128]$ and $[N_{item}, 128]$, respectively, where N_{user} and N_{item} are the total numbers of users and items.

Joint discriminator: During the validation and testing stage, we exploit the fused distance of two metrics (MSE and mutual information) as the joint discriminator to make predictions from all candidates of items to choose the highest ranking item:

$$\begin{aligned} fusion_distance(i) = & \alpha_m \|\tilde{J}(t + \Delta) - \bar{J}_i\|_{i \in [1, \dots, N_{item}]} \\ & + \alpha_n f_1(x_i, c_t)_{i \in [1, \dots, N_{item}]}, \end{aligned} \quad (20)$$

$$i_{t+1} = \arg \max_{i \in [1, \dots, N_{item}]} fusion_distance(i), \quad (21)$$

where i_{t+1} is the predicted item at timestamp $t + 1$.

The hyper parameters α_m and α_n are obtained by greedy searches on both datasets, respectively. The best values of these two hyper parameters are $\{0.75, 0.25\}$ and $\{0.9, 0.1\}$ for Tmall and 'UsedCars' datasets, respectively.

5.2.3 Evaluation Protocols

We evaluate all models with two popular ranking-based metrics: Recall@K and Normalized Discounted Cumulative Gain (NDCG). Recall@K measures the proportion of the top-K recommended items that are in the ground truth candidate ones where $K = \{10, 20\}$. In the context of sequential recommendation, $NDCG = \frac{1}{\log_2(1 + rank_{pos})}$, where $rank_{pos}$ is the rank of correctly predicted items.

5.2.4 Baselines

To evaluate the performance of our proposed method, we compare it with five state-of-the-art methods:

- SASRec [2] is a self-attention based sequential model, and it can consider consumed items for next item recommendation.
- NextItNet [14] applies 1D CNNs with dilated convolution filters and residual blocks to model sequential recommendation.
- GRU4Rec [19] applies GRU to model user click sequences for the session-based recommendation.
- FPMC [16] fuses matrix factorization and first-order Markov Chains to capture long-term preferences and short-term item-item transitions, respectively, for next item recommendation.
- JODIE [42] introduce mutually recursive RNN to model general user and item embedding for sequential recommendation.

5.3 Experiment Results

5.3.1 Results

In Table 3, the results on the Tmall dataset illustrate that our proposed RecNet outperforms all baselines except $\{\text{Recall@10}, \text{NDCG@10}\}$ of SASRec [2]. We analyze that the advantage of Recall metrics of SASRec over RecNet* is mainly due to the dataset's characteristics. The Tmall/Koubei dataset mainly contains users' daily repeated cheap consumption of small amounts, such as catering and clothing. The small purchase amount can not impact the users'

Table 3 Performance comparison on two datasets

Datasets	Metrics	FPMC	SASRec	GRU4Rec	NextItNet	JODIE	RecNet_MSE	RecNet_NCE	RecNet_Cos	RecNet
UsedCars	Recall@10	0.0554	0.0414	0.0320	0.0228	0.0659	0.0642	0.0480	0.0527	0.0856
	NDCG@10	0.0274	0.0264	0.0159	0.0112	0.0527	0.0518	0.0355	0.0410	0.0660
	Recall@20	0.0933	0.0640	0.0589	0.0413	0.0933	0.0950	0.0783	0.0854	0.1281
	NDCG@20	0.0432	0.0292	0.0226	0.0158	0.0627	0.0630	0.0463	0.0529	0.0815
Tmall	Recall@10	0.4012	0.5108	0.3211	0.3917	0.4014	0.4107	0.3429	0.3602	0.4296
	NDCG@10	0.3089	0.4862	0.2484	0.3305	0.5089	0.5169	0.3939	0.4388	0.5393
	Recall@20	0.4503	0.5256	0.3675	0.4256	0.4148	0.4247	0.3822	0.4077	0.4562
	NDCG@20	0.3213	0.4900	0.2600	0.3391	0.5138	0.5221	0.4092	0.4214	0.5490

Best results are shown in bold

resource constraint factors. When in bulk commodities trading (i.e., user car transactions), our method is better.

Comparing the results of RecNet_NCE/RecNet_MSE (see Sect. 5.3.2) and RecNet in Table 3, we find that the mutual information and resource-limited branch significantly improve NDCG@k metrics. On the ‘UsedCars’ dataset, RecNet is ahead of all baselines.

5.3.2 Ablation Study

To further prove the roles of our proposed resource constraint branch and mutual information constraint for representation learning, we conduct the following three experiments:

Efficacy of resource constraint branch: In the variant model RecNet*, the resource constraint branch $\mathbf{h}_u^l(t + \Delta)$ is removed from the combined user embedding [used in Eq. (12)] while other parts of RecNet* are consistent with RecNet. The other experiment’s settings for this variant are the same as RecNet. The results in Table 4 show that RecNet*’s performance degraded compared with RecNet, on both datasets, which implies that the resource constraint branch plays an important role in RecNet.

Efficacy of NCE Loss: To prove the effectiveness of NCE loss, in the variant RecNet_MSE, the NCE loss is abandoned from RecNet while both the MSE loss and two regularization terms are preserved:

$$\begin{aligned}
 Loss = \tilde{L}_{MSE} = & \sum \|\tilde{J}(t + \Delta) - \tilde{J}\|_2 \\
 & + \lambda_U \sum_u \|\mathbf{h}_u^g(t) - \mathbf{h}_u^g(t - 1)\|_2 \\
 & + \lambda_I \sum_i \|\mathbf{h}_i(t) - \mathbf{h}_i(t - 1)\|_2.
 \end{aligned} \tag{22}$$

We employ $f(\mathbf{h}_u^g(t + \Delta), \mathbf{h}_u^l(t + \Delta))$ in Eq. (11) to model the resource constraint, while the other parts of the model are the same as RecNet. As shown in Table 3, RecNet_MSE’s performance drops heavily compared with RecNet, which shows NCE loss’s effectiveness in performance improvement.

On the other hand, we can prove the effectiveness of mutual information. In the variant RecNet_NCE, the whole MSE loss is discarded. The model was trained following an unsupervised style by using NCE loss alone :

$$\begin{aligned}
 Loss = L_{NCE} + & \lambda_U \sum_u \|\mathbf{h}_u(t) - \mathbf{h}_u(t - 1)\|_2 \\
 & + \lambda_I \sum_i \|\mathbf{h}_i(t) - \mathbf{h}_i(t - 1)\|_2
 \end{aligned} \tag{23}$$

Results in Table 3 reveals that only unsupervised training can achieve competitive performance on the ‘UsedCars’ dataset, compared with supervised baselines: FPMC, SASRec, GRU4Rec, NextItNet. During this experiment, we use the following distance metric for evaluation and testing:

$$mi_distance(i) = f_1(x_i, c_{t_{i \in \{1, \dots, N_{item}\}}}), \tag{24}$$

$$i_{t+1} = \arg \max_{i \in \{1, \dots, N_{item}\}} mi_distance(i). \tag{25}$$

Advantages of mutual information maximization: As discussed in [35, 43], most of the research about mutual information’s application focuses on the areas of image, audio, etc. However, despite some works [34, 33, 37, 35] successfully obtaining promising results via mutual information maximization, other research [43] shows that maximizing

Table 4 The ablation study results of the resource constraint branch

Datasets	Metrics	JODIE	RecNet	RecNet*
‘UsedCars’	Recall@10	0.0659	0.0856	0.0710
	NDCG@10	0.0527	0.0660	0.0518
	Recall@20	0.0933	0.1281	0.1027
	NDCG@20	0.0627	0.0815	0.0669
Tmall	Recall@10	0.4014	0.4296	0.4154
	NDCG@10	0.5089	0.5393	0.5164
	Recall@20	0.4148	0.4562	0.4330
	NDCG@20	0.5138	0.5490	0.5264

*denotes that the resource constraint branch is removed from RecNet

tighter bounds on mutual information may lead to worse representation learning. This contrast shows that we need to add other auxiliary sub-modules to help to learn. Due to these reasons, we designed the joint discriminative framework and conducted the above three experiments to prove the effectiveness of our design.

To explain the advantages that we take mutual information maximization to learn the connections between the user's historical actions and future ones, we choose another cosine distance loss, commonly used in metric learning, to construct a variant RecNet_Cos of RecNet. We compared the performance between RecNet_Cos and RecNet in Table 3, showing that the cosine distance version's performance is even worse than that of RecNet_MSE, let alone RecNet.

6 Conclusion

In this paper, we argue that the limited product inventory and the decline of users' financial strength will affect consumers' choices or preferences, which has been ignored in the previous literature. Our proposed model uses two user state embeddings to capture the user's historical preference and resource constraint factors. Also, we use a mutual information estimator to help improve representation learning. Experiments show that our model achieves a competitive result in the sequential recommendation of bulk commodities compared with the state-of-the-art. In future work, we will study how to learn representations that separate the latent explanatory factors behind the data (e.g., merchant inventory, user inventory, capital strength, display strategy) rather than only two rough branches in RecNet.

Acknowledgements The authors would like to thank the anonymous reviewers for their constructive comments and the School of Computer Science and Engineering for financial support. This work was partially supported by the Key Project of the Chinese Ministry of Education (Grant No. JYB202103), the PhD research startup foundation of Jinling Institute Technology (Grant No. jit-b-202021), and the Shanghai Youth Science and Technology Talents Sailing Program (Grant No. 22YF1413700).

Author Contributions HS: Methodology, Resource and writing, Including methodology development, Data collection, Experiment design, Paper writing. JQ: Mainly for model construction and data analysis. NZ: Mainly responsible for writing, including review and editing. TH: Mainly responsible for writing, including review and editing. ZC: Makes an important contribution on Algorithm design. QW: Conceives the study, participates in its design and coordination and helps to draft the manuscript. SF: Mainly for software and coding.

Funding This work was partially supported by the Key Project of Chinese Ministry of Education (Grant No. JYB202103), the PhD research startup foundation of Jinling Institute Technology (Grant No. jit-b-202021), and the Shanghai Youth Science and Technology Talents Sailing Program (Grant No. 22YF1413700).

Data availability The 'UsedCars' dataset can not be shared, since it belongs to one enterprise.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval and consent to participate Not applicable.

Consent for publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Yera, R., Martinez, L.: Fuzzy tools in recommender systems: a survey. *Int. J. Comput. Intell.* **10**(1), 776 (2017). <https://doi.org/10.2991/ijcis.2017.10.1.52>
2. Kang, W.-C., McAuley, J.J.: Self-attentive sequential recommendation. 2018 IEEE International Conference on Data Mining (ICDM), 197–206 (2018). <https://doi.org/10.1109/icdm.2018.00035>
3. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017). <https://doi.org/10.1145/3038912.3052569>
4. Rao, N., Yu, H.-F., Ravikumar, P.K., Dhillon, I.S.: Collaborative filtering with graph information: Consistency and scalable methods. In: Advances in Neural Information Processing Systems, pp. 2107–2115 (2015)
5. Yang, Y., Zhou, D., Zhan, D., Xiong, H., Jiang, Y.: Adaptive deep models for incremental learning: Considering capacity scalability and sustainability. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, pp. 74–82 (2019). <https://doi.org/10.1145/3292500.3330865>
6. Yang, Y., Wu, Y., Zhan, D., Liu, Z., Jiang, Y.: Complex object classification: A multi-modal multi-instance multi-label deep network with optimal transport. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, pp. 2594–2603 (2018). <https://doi.org/10.1145/3219819.3220012>
7. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **8**, 30–37 (2009). <https://doi.org/10.1109/MC.2009.263>
8. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: *The Adaptive Web*, pp. 325–341. Springer (2007). https://doi.org/10.1007/978-3-540-72079-9_10
9. Bai, B., Fan, Y., Tan, W., Zhang, J.: Dltsr: a deep learning framework for recommendation of long-tail web services. *IEEE Trans.*

- Serv. Comput. (2017). <https://doi.org/10.1109/TSC.2017.2681666>
10. Van den Oord, A., Dieleman, S., Schrauwen, B.: Deep content-based music recommendation. In: *Advances in Neural Information Processing Systems*, pp. 2643–2651 (2013)
 11. Oh, K.J., Lee, W.J., Lim, C.G., Choi, H.J.: Personalized news recommendation using classified keywords to capture user preference. In: *16th International Conference on Advanced Communication Technology*, pp. 1283–1287. IEEE (2014). <https://doi.org/10.1109/ICACT.2014.6779166>
 12. Wang, D., Liang, Y., Xu, D., Feng, X., Guan, R.: A content-based recommender system for computer science publications. *Knowl. Based Syst.* **157**, 1–9 (2018). <https://doi.org/10.1016/j.knosys.2018.05.001>
 13. Bai, T., Du, P., Zhao, W.X., Wen, J.-R., Nie, J.-Y.: A long-short demands-aware model for next-item recommendation. *ArXiv abs/1903.00066* (2019). <https://doi.org/10.48550/arXiv.1903.00066>
 14. Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J.M., He, X.: A simple convolutional generative network for next item recommendation. In: *WSDM '19* (2019). <https://doi.org/10.1145/3289600.3290975>
 15. Wang, C., Zhang, M., Ma, W., Liu, Y., Ma, S.: Modeling item-specific temporal dynamics of repeat consumption for recommender systems. In: *WWW '19* (2019). <https://doi.org/10.1145/3308558.3313594>
 16. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: *WWW '10* (2010). <https://doi.org/10.1145/1772690.1772773>
 17. Ruining, H., Julian, M.: Fusing similarity models with markov chains for sparse sequential recommendation. (2016). <https://doi.org/10.48550/arXiv.1609.09152>
 18. Tomassini, M., Luthi, L.: Empirical analysis of the evolution of a scientific collaboration network. *Phys A* **385**(2), 750–764 (2007). <https://doi.org/10.1016/j.physa.2007.07.028>
 19. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. *CoRR abs/1511.06939* (2015). <https://doi.org/10.48550/arXiv.1511.06939>
 20. Zhu, Y., Li, H., Liao, Y., Wang, B., Guan, Z., Liu, H., Cai, D.: What to do next: Modeling user behaviors by time-lstm. In: *IJCAI* (2017)
 21. Quadrona, M., Karatzoglou, A., Hidasi, B., Cremonesi, P.: Personalizing session-based recommendations with hierarchical recurrent neural networks. In: *RecSys '17* (2017). <https://doi.org/10.48550/arXiv.1706.04148>
 22. Smirnova, E., Vasile, F.: Contextual sequence modeling for recommendation with recurrent neural networks. *ArXiv abs/1706.07684* (2017). <https://doi.org/10.1145/3125486.3125488>
 23. Yu, Z., Lian, J., Mahmood, A., Liu, G., Xie, X.: Adaptive user modeling with long and short-term preferences for personalized recommendation. In: *IJCAI* (2019)
 24. Yan, A., Cheng, S., Kang, W.-C., Wan, M., McAuley, J.J.: Cosrec: 2d convolutional neural networks for sequential recommendation. In: *CIKM '19* (2019)
 25. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. *ArXiv abs/1809.07426* (2018). <https://doi.org/10.1145/3159652.3159656>
 26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: *NIPS* (2017)
 27. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. *CoRR abs/1409.0473* (2014). <https://doi.org/10.48550/arXiv.1409.0473>
 28. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.-Y.: Mass: Masked sequence to sequence pre-training for language generation. In: *ICML* (2019)
 29. Ying, H., Zhuang, F., Zhang, F., Liu, Y., Xu, G., Xie, X., Xiong, H., Wu, J.: Sequential recommender system based on hierarchical attention networks. In: *IJCAI* (2018)
 30. Yuan, W., Wang, H., Yu, X., Liu, N., Li, Z.-H.: Attention-based context-aware sequential recommendation model. *Inf. Sci.* **510**, 122–134 (2020). <https://doi.org/10.1016/j.ins.2019.09.007>
 31. Xu, C., Zhao, P., Liu, Y., Sheng, V.S., Xu, J., Zhuang, F., Fang, J., Zhou, X.: Graph contextualized self-attention network for session-based recommendation. In: *IJCAI* (2019)
 32. Yu, K., Xu, X., Ester, M., Kriegl, H.-P.: Feature weighting and instance selection for collaborative filtering: an information-theoretic approach*. *Knowl. Inform. Syst.* **5**, 201–224 (2003). <https://doi.org/10.1007/s10115-003-0089-6>
 33. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. *ArXiv abs/1906.05849* (2019). https://doi.org/10.1007/978-3-030-58621-8_45
 34. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. *ArXiv abs/1808.06670* (2019). <https://doi.org/10.48550/arXiv.1808.06670>
 35. Hénaff, O.J., Srinivas, A., Fauw, J.D., Razavi, A., Doersch, C., Eslami, S.M.A., van den Oord, A.: Data-efficient image recognition with contrastive predictive coding. *ArXiv abs/1905.09272* (2019). <https://doi.org/10.48550/arXiv.1905.09272>
 36. Sun, C., Baradel, F., Murphy, K., Schmid, C.: Contrastive bidirectional transformer for temporal representation learning. *ArXiv abs/1906.05743* (2019). <https://doi.org/10.48550/arXiv.1906.05743>
 37. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. *ArXiv abs/1807.03748* (2018). <https://doi.org/10.48550/arXiv.1807.03748>
 38. Yu, L., Liu, L.: Collaborative filtering algorithm based on mutual information. In: *PACIS* (2004)
 39. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: *AISTATS* (2010)
 40. Wang, X., Zhang, H., Huang, W., Scott, M.R.: Cross-batch memory for embedding learning. *arXiv:1912.06798* (2019). <https://doi.org/10.1109/CVPR42600.2020.00642>
 41. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
 42. Kumar, S., Zhang, X., Leskovec, J.: Predicting dynamic embedding trajectory in temporal interaction networks. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2019). <https://doi.org/10.1145/3292500.3330895>
 43. Tschannen, M., Djolonga, J., Rubenstein, P.K., Gelly, S., Lucic, M.: On mutual information maximization for representation learning. *ArXiv abs/1907.13625* (2019). <https://doi.org/10.48550/arXiv.1907.13625>
 44. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches (2014). *arXiv preprint arXiv:1409.1259*

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.