# Neural factorization for Offer Recommendation using Knowledge Graph Embeddings

Gourab Chowdhury
Indian Institute of Technology Kharagpur
West Bengal, India
gourab@iitkgp.ac.in

Madiraju Srilakshmi
Indian Institute of Technology Kharagpur
West Bengal, India
sreelakshmi@iitkgp.ac.in

Mainak Chain
Indian Institute of Technology Kharagpur
West Bengal, India
mainakchain@iitkgp.ac.in

Sudeshna Sarkar
Indian Institute of Technology Kharagpur
West Bengal, India
sudeshna@cse.iitkgp.ernet.in

## ABSTRACT

Companies send a lot of promotional offers and coupons to customers to attract them to buy more. Offer recommendation systems can help to identify relevant offers to users. In this paper, we present a Neural Factorization (NF) model for the task of Offer recommendation. We represent users and offers with Knowledge Graph Embeddings (KGE). Specifically, we model the available data in the form of a Knowledge Graph (KG) and learn embeddings for entities and relations using a standard KGE technique called TransE. We also incorporate the user temporal features in the NF model using Long Short Term Memory (LSTM) with attention framework. We experiment with Kaggle Acquire Valued Shoppers Challenge dataset and show that the performance of our model is significantly better than tree-based methods.

## CCS CONCEPTS

• **Information systems → Recommender systems**; • **Information Systems → Information retrieval**; *Recommender Systems*; • **Computing methodologies** → Neural networks.

## KEYWORDS

Neural Factorization, Recommender Systems, Knowledge Graph Embedding, E-commerce offers

**ACM Reference Format:**
Gourab Chowdhury, Madiraju Srilakshmi, Mainak Chain, and Sudeshna Sarkar. 2019. Neural factorization for Offer Recommendation using Knowledge Graph Embeddings. In *Proceedings of the SIGIR 2019 Workshop on eCommerce (SIGIR 2019 eCom)*, 6 pages.

## 1 INTRODUCTION

Marketing and promotions are used to attract customers in the retail domain. Companies spend a lot of money to send promotional offers or discounts to customers. It is therefore important to identify relevant offers that the users are likely to accept.


*In: J. Degenhardt, S. Kallumadi, U. Porwal, A. Trotman (eds.):*
*Proceedings of the SIGIR 2019 eCom workshop, July 2019, Paris, France, published at http://ceur-ws.org*

We consider the case in which an offer is a flat discount for a given brand and category combination. An example of an offer is "Rs 200 off on Polo T-shirts". There can be other types of offers such as coupons/promo codes (Use Coupon GO15O to get Rs 150 cash-back), combo offers (Buy 3 get 1 free, Buy 3 get Rs 300 off) and loyalty points.

Xia et al. [13] proposed an approach for the task of offer recommendation based on the features of users and offers. [13] used tree-based methods namely Random forest [4] and Gradient boosted trees [5] to handle this task. The features involve attributes of the users such as their email domain and OS systems and attributes of offers such as text description, discount amount as well as shopping history features such as recent discount information and number of times the user visited the coupon before.

We propose to use a Neural Factorization (NF) model to learn user-offer interactions. The user and offer representations are given as input to the NF model whose output is the probability of the offer being accepted by the user. We predict the probabilities of all offers available at the given time and recommend the top $k$ probable offers to the customer. In this work, we explore different ways of representing users and offers.

In our first model, we represent users and offers with features extracted from the dataset. The user features contain the normalized count of items purchased in a month, the normalized count of items purchased in each category, days since the last visit, the average amount paid per visit etc. The offer features include category, brand on which offer is given, amount of discount, minimum quantity of purchase etc.

In our second model, we explore representing user and offers as embedding. For this, we construct a Knowledge Graph (KG) involving users, categories, brands, price values as nodes and belongs-to, purchase and price as edges between them. We adopt a knowledge graph embedding technique called TransE [3] to generate embeddings of users and offers.

In our third model, we capture the user sequential behaviour using Long Short Term Memory (LSTM) with attention framework. The input to the model is the sequence of baskets purchased by the user and the output is the probabilities on all categories available. We incorporate this information as an additional input to the NF model.

We experiment with *Kaggle Acquire Valued Shoppers Challenge* dataset which contains user offer interactions, user purchase history

**Table 1: Overall data statistics**

| Statistics | Acquire Valued Shoppers data |
|---|---|
| # users | 315411 |
| # brands | 35689 |
| # offers | 39 |
| # categories | 836 |
| # total transactions | 349,655,789 |

and offer content information. We apply our models on this data and show that the NF based models achieves better performance than tree-based methods.

## 2 PRELIMINARIES

In this section, we formally define the task of offer recommendation and present the details of the data available to handle this task.

### 2.1 Problem Definition

Offer Recommendation is the task of predicting the best offers for a given user. Let $U = \{u_1, u_2, ..., u_m\}$ be the set of users and $O = \{o_1, o_2, o_3 ... o_n\}$ be the set of offers. The task is to recommend top $k$ offers to the users so that they are likely to include the next offer converted by the user.

### 2.2 Dataset

In our work, we experiment with *Kaggle Acquire Valued Shoppers Challenge* dataset [1].

The dataset contains the transaction history of users from March 2012 to July 2013. A transaction consists of user_id, item_id and date. The set of items purchased by the same user in the same date are termed as a basket.

The user-offer interactions are recorded from March 2013 to July 2013. A user-offer interaction consists of user_id, offer_id and date. Each user has availed exactly one offer in this period. Each offer is specified by its category, brand, discount amount and minimum quantity. The overall data statistics are listed in Table 1.

## 3 NEURAL FACTORIZATION FOR OFFER RECOMMENDATION

We use Neural Factorization (NF) model for the task of offer recommendation. We have experimented with different methods of representing users and offers. We first explain the basic neural factorization framework and then introduce our methods.

### 3.1 Architecture of Neural Factorization

Our framework is based on Neural Collaborative Filtering proposed by He et al. [6]. The system [6] used Multi Layer Perceptron (MLP) for modelling user-item interactions which is able to capture the non-linear relations between users and items.

In our model, the user vector ($v_u$) and offer vector ($v_o$) are given as input to two input layers. Each input layer is followed by a dense layer. The output of these dense layers are concatenated and are given as input to a Neural network. We use past user-offer interactions as positive samples while the negative samples are generated by random sampling. The final output layer predicts $y_{uo}$, the probability of user $u$ accepting an offer $o$. The architecture is illustrated in Figure 2.

The model is trained by minimizing the loss between predicted value ($\hat{y_{uo}}$) and the target value $y_{uo}$. The target value represents the user action towards an offer. It is 1 when the user availed the offer and 0 otherwise. The loss function is defined as follows:

$$L = \sum_{(u,o) \in \mathcal{Y}^- \cup \mathcal{Y}^+} -(y_{uo} \log(\hat{y_{uo}}) + (1 - y_{uo}) \log(1 - \hat{y_{uo}})) \quad (1)$$

where $\mathcal{Y}^+$ denotes the set of positive interaction of users, offers and $\mathcal{Y}^-$ denotes the negative instances (sampled from unobserved data).

The user_ids and offer_ids are different in the train and test datasets. Therefore, we can't use the learned factors for the prediction. We represent the user and offer by their content information and input them to the trained neural model and predict the probability of the user accepting the offer. Similarly, we find the probabilities of all offers available at the given time. The offers are ranked based on the probability values and top $k$ offers are recommended to the user.

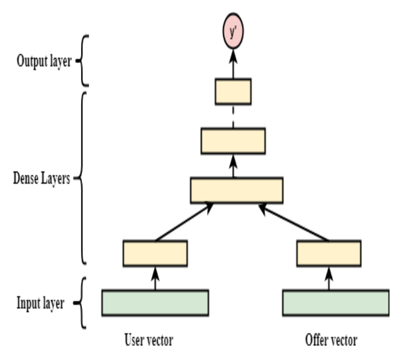


**Figure 1: Neural Factorization Model**

### 3.2 Neural Factorization with features (NF+features)

In this variation of our method, user and offer features obtained from the dataset are given as input to the neural factorization model. We extract user and offer features from user purchase history, user-offer interactions and offer content information. The following features are used to represent users and offers.

- User features:
  - Normalized count of purchase of each category
  - Number of visits in 30 days before the day of offer
  - Days since the last visit (from the day of offer).
  - Average items purchased per visit
  - Average amount paid per visit
- Offer features:
  - Category on which offer is given
  - Brand on which offer is given
  - Average price of the each brand-category combination
  - Discount amount

[1] https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data

– Quantity to be purchased to avail discount
– How cheap the product is compared to other products in the same category (Amount).

The numeric value is used as input for the numeric features while the non-numeric features such as brand and category are one-hot encoded. Since these features have multiple possible values, the feature dimensions become large. These features are also unable to capture indirect relationships between users and offers.

## 3.3 Neural Factorization with Knowledge Graph Embeddings (NF+KGE)

The representation of users and offers plays a significant role in the effectiveness of a recommendation model. We wish to use a representation that is able to capture relevant knowledge of users, items, offers, the attributes of the above entities and the interactions between them.

We propose to use Knowledge Graph Embedding (KGE) techniques to learn embeddings for users and offers. These techniques have been found to be effective in capturing complex and indirect relationships among entities in the Knowledge Graph (KG) and are proven to be successful in many applications such as link prediction and recommendation systems etc [12].

We construct a Knowledge Graph (KG) based on user purchase history and offer content information. The nodes of the graph are *users*, *categories*, *brands* and *price range*. We find the average price of items in a brand and categorize them into *high* and *low*. We have 3 types of relations in our graph. The user nodes are connected to category nodes by relation *purchased*, category nodes are connected to brand nodes by relation belongs_to and brand nodes are connected to price range nodes by relation price. The graph is formed as a set of triplets $(h, r, t)$ i.e., head node $(h)$ is connected to tail node $(t)$ by relationship $(r)$. An example graph representation is shown in Figure 3.

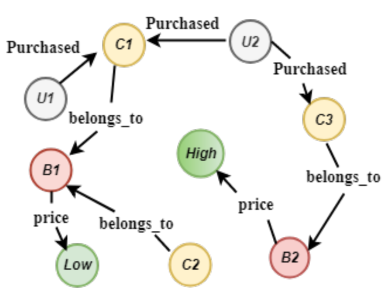


**Figure 2: Knowledge graph representation of available data**

The triplets generated for the example graph are as follows: $T = \{(U_1, purchased, C_1), (U_2, purchased, C_1), (C_1, belongs\_to, B_1), (C_2, belongs\_to, B_1), (C_3, belongs\_to, B_2), (B_1, price, low), (B_2, price, high)\}$.

We use a standard knowledge graph embedding method called TransE [3] to learn the embeddings for all entities and relationships in the graph. We have chosen TransE because this method is simple and has been found to be efficient in modelling multi-relational data [12].

Given a triplet of the form *<h,r,t>* which indicates that the head entity (h) is connected to the tail entity (t) by the relationship (r), TransE [3] learns the embedding such that $h + r \approx t$ (Figure 3). TransE uses the following scoring function:

$$f_r(h, t) = -||h + r - t||_{1/2} \tag{2}$$

TransE obtains positive triplets from the graph and negative triplets

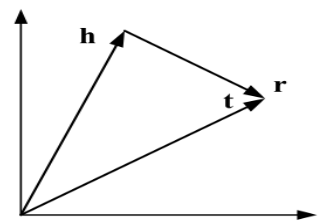


**Figure 3: Illustration of TransE (Figure is adapted from [12]**

by randomly corrupting the head or tail node of a positive triplet. The TransE training method minimizes the pairwise loss between the positive facts $\tau^+ = (h, r, t)$ and the negative facts $\tau^- = (h', r', t')$. The objective function is given as follows:

$$min \sum_{\tau^+ \in D^+} \sum_{\tau^- \in D^-} max(0, \gamma - f_r(h, t) + f_{r'}(h', t')) \tag{3}$$

$\gamma$ is the margin of separation, $D^+$ is the set of positive facts and $D^+$ is the set of negative facts. The steps to learn the embeddings are given in Algorithm 1.

**Algorithm 1:** TransE method to learn embeddings for entities and relations

**Input:** Training set $T = (h, r, t)$ , Entity set $E$, Relation set $R$
Randomly initialize the embeddings for $e \in E, r \in R$
**for** $(h, r, t)$ *in* $T$ **do**
  // Corrupt the triplet by changing $h$ or $t$ and add it to $T$
  $T \leftarrow T \cup (h', r, t')$
  Update embeddings w.r.t the loss $L$ i.e.
  $\sum_{(h',r,t'),(h,r,t) \in T} \Delta[\gamma + d(h + r, t) - d(h' + r, t')]$
  // Dissimilarity measure $d$ can be either the $L1$ or the $L2$ norm
**end**

We input the learned graph embeddings to the NF model. The users are represented with the user embedding and offers are represented with the combination of category embedding, brand embedding, discount amount and minimum quantity to be purchased to avail the offer.

## 3.4 Neural Factorization with temporal features (NF+KGE+TF)

The above Knowledge Graph Embeddings (KGE) is unable to capture the sequential behaviour of users since the knowledge graph does not represent the time stamp of the interactions or the sequentiality of the transactions. Therefore, we try to enhance our model by incorporating a temporal component by considering the user sequential purchase behaviour using a Long Short Term Memory (LSTM) with attention model.

Since the offers are given on specific categories, we formulate the task of predicting the next category to be purchased by the user. We hypothesize that this information may help to identify suitable offers for a user.

Each user has purchased a set of items per visit. The set of items can be termed as a basket. We consider the category of an item (category_id) instead of the exact item (item_id) in each basket. We give the sequence of baskets purchased by the user as input to the LSTM and predict the probabilities of he categories purchased in the next basket. The predicted category probabilities are incorporated as an additional input to the NF model.

Let $u = \{b_1, b_2, \ldots, b_t\}$ be the basket sequence of the target user. Each basket is the group of categories: $b_k = \{c_1, c_2, \ldots, c_n\}$, where $n$ is the size of the basket. We represent each category with the embedding learned from the Knowledge Graph discussed earlier. We use average pooling to represent the basket. This approach is similar to the basket prediction method proposed by Yu et al.[14].

The user sequence is now denoted as $u = \{v_1, v_2, \ldots, v_t\}$, where $v_k$ is the average of graph-based category embeddings in the basket $b_k$. We input the user sequence till the current time $t$ into the LSTM model. Let $h_t$ be the LSTM hidden unit and $y_t$ be the output at $t$-th time step. The hidden state $h_t$ of each interaction is updated by the previous hidden state $h_{t-1}$ and the current basket embedding $v_t$.

$$h_t = LSTM(h_{t-1}, v_t) \tag{4}$$

We apply attention on top of the LSTM layer to give weights to the baskets at different time-steps. Let $H = (h_1, h_2, \ldots, h_t)$ be the output vectors that are produced by LSTM layer. They are inputs to the attention layer and the weights for each time-step are learned i.e. $A = (a_1, a_2, \ldots, a_t)$. The weighted sum of the hidden states $(M)$ is input into a dense layer $(D)$ to find the scores of all categories. We find their probabilities using the sigmoid activation function. This architecture is illustrated in Figure 4.

$$A = softmax(w^T * \tanh(H)) \tag{5}$$

$$M = A^T * H \tag{6}$$

$$P\,(y_{t+1} \mid v^i_{1 \le i \le t}) = sigmoid(D[M]) \tag{7}$$

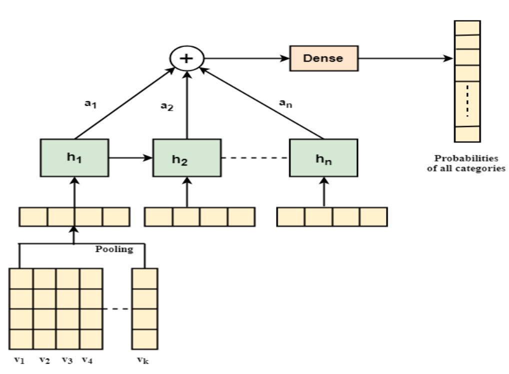


**Figure 4: LSTM model for next category prediction**

**Table 2: Train and Test data statistics**

| | # Users | # Distinct Offers | Start Date | End Date |
|---|---|---|---|---|
| Train | 160057 | 24 | 01-March-2013 | 30-April-2013 |
| Test | 151484 | 29 | 01-May-2013 | 31-July-2013 |

The learned category probabilities are given as an auxiliary input to the NF model. The rest of the architecture is similar and is shown in Figure 5.

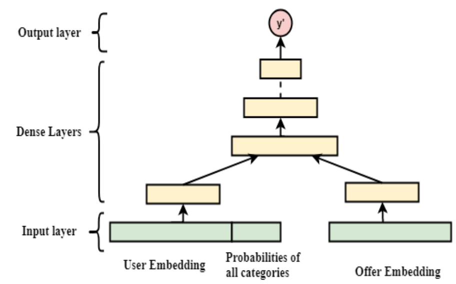


**Figure 5: Incorporating category preferences in NF**

# 4 EXPERIMENT

## 4.1 Train and testset details

The train and test split are considered as given in the dataset i.e., the first 2 months of user-offer interaction data are used as train and the rest of the 3 months as the test set. The train and test statistics are listed in Table 2. To train the Knowledge Graph Embeddings, we have used recent 5 months of user purchase history before offers are given to the users (01-October-2012 to 28-February-2013). This has been done on the assumption that the recent purchase history reflects the current preferences of the user. The same data is used for predicting user category preferences as well as feature creation.

## 4.2 Evaluation Metrics and Experimental Setup

*4.2.1 Evaluation Metrics.* We use the following metrics to evaluate our model.

- Recall@1, Recall@3 and Recall@5, where Recall@k is the proportion of cases when the offer accepted by the user is among the top-$k$ recommended offers among all test cases.
- Mean Reciprocal Rank@5 (MRR@5): It is the mean of reciprocal ranks of correctly recommended offers. If the actual offer is not in the top-5 recommended offers, then the rank is set to zero.

*4.2.2 Experimental Setup.* We use the following parameters to learn knowledge graph embedding. The size of the embedding for all entities (user, brand, category, price range) is 100. We use Adam optimizer, and the learning rate is set to 0.001.

The parameters for predicting user next preferred categories using LSTM are as follows. We use one LSTM layer with 100 hidden

**Table 3: Performance comparison of our approach with baseline methods(in %)**

| Method | Recall@1 | Recall@3 | Recall@5 | MRR@5 |
|---|---|---|---|---|
| RFC | 4.93 | 11.43 | 22.49 | 9.23 |
| XGBoost | 6.73 | 12.56 | 24.05 | 14.69 |
| NF+features | 15.41 | 22.05 | 33.19 | 23.48 |
| NF+KGE | **21.77** | 26.55 | 34.90 | 25.57 |
| NF+KGE+TF | 15.96 | **34.00** | **46.68** | **27.28** |

units. There is a dropout layer in between the LSTM layer and the attention layer with 25% dropout. The learning rate is set to 0.001.

The parameters for neural factorization models are as follows. The activation function for dense layers is Leaky Relu and the batch size is set at 512. The learning rate is set to 0.0001. We applied batch normalization at every layer.

### 4.3 Comparison of Models

We compare the NF based models namely NF+features, NF+KGE and NF+KGE+TF against XGBoost and Random Forest Classifier (RFC) methods with features.

The standard item KNN method can't be applied to this dataset because each user in the dataset has availed exactly one offer. There is no possibility of finding similar offers to the offers that are previously availed by the user and recommend them.

### 4.4 Result and Analysis

The results for the two baseline methods and the three NF based models discussed in this paper are presented in Table 1. It is evident from the results that the NF based models outperform the XGboost and RFC methods with features.

Neural factorization model with graph-based embedding (NF+KGE) performs better than Neural factorization with features (NF+features).

The final variation of our model with temporal features (NF+KGE+TF) gives the significant improvement over all other models considered in terms of Recall@3 and Recall@5 and MRR@5. Neural factorization model with graph-based embedding (NF+KGE) performs best in terms of Recall@1.

We hypothesize that the knowledge graph based embedding is effective as it is able to use the connections between different entities and can therefore effectively capture indirect as well as latent relationships.

However the limitation of the above model is that it fails to capture the temporal interactions of the user. Our third model (NF+KGE+TF) addresses this by using an additional input from the LSTM based on the user's sequence of baskets.

## 5 RELATED WORK

Related work to our model is categorized into three parts. Subsection 5.1 reviews the work reported on offer recommendation systems. Subsection 5.2 reviews the methods used in a related task of repeat purchase prediction methods. In subsection 5.2, we discuss about Knowledge Graph Embedding (KGE) based recommendation systems.

### 5.1 Offer Recommender Systems

There are very few works on offers and coupons recommendation systems.

Xia et al. [13] approached the problem of offer recommendation in e-commerce domain. They used a private dataset consists of customer's shopping trips, shopping trip counts, clicked coupons, and retailers that issued the coupon. The coupons are characterized based on their textual descriptions and validity period. The authors curated a number of features from the data to represent users and coupons and ranked the coupons based on scores generated by XGBoost [5] and Random Forest [4] algorithms.

Similar work is proposed by Hu et al. [7] in the telecom domain. Hu et al. [7] used Random Forest method [4] to provide telecom offers to mobile users. The authors extracted user features such as age, gender, voice call duration, SMS count etc. from the customer profile repository, and historical usage repository. These features are given as input to the Random Forest algorithm [4].

### 5.2 Repeat Purchase Prediction Systems

Repeat purchase prediction is the task given for *Kaggle Acquire Valued Shoppers Challenge*. Since we have used the same dataset for our task of offer recommendation systems, we present a review of the work done on repeat purchase prediction.

Anand et al. [2] proposed a prediction model based on a combination of temporal and aggregate level models. They extracted three types of features capturing different aspects of user behaviour namely *customer-based, product-based, customer-product interactions based. customer-based* features include total visits made, total spend, the loyalty of the customer etc. *product-based* features include the fraction of repeat customers for the offer-product etc. *customer-product interactions based* features include the number of visits, quantity bought, the amount spent etc.

To capture the aggregate level behaviour of the user, the above features are computed over the entire transaction history. To capture the temporal behaviour, the features are split and computed over non-overlapping time windows. The authors used Long Short Term Memory (LSTM) model as the classifier for temporal features and quantile regression (QR) model as the classifier for aggregate level features. The two models are combined using a mixture of experts (ME).

Nikulin et al. [10] used Random forest [4] and Gradient boosted trees [5] to predict the repeat purchase behaviour of the users. The authors curated a number of statistical features from the data and applied the above methods.

### 5.3 Knowledge Graph Embedding based Recommender Systems

In recent times, knowledge graph embeddings have been shown to be effective for recommendation systems. The basic idea is to represent the available data in the form of a graph, learn embeddings for entities using Knowledge graph embedding methods [12] and incorporate them into recommendation.

[15] presents Collaborative Knowledge base Embedding (CKE) which uses TransR [9] to learn the structural representations of items which are combined with visual and textual embeddings.

Deep Knowledge-aware Network (DKN) [11] learns entity embeddings using TransD [8] and and designs a CNN framework by combining them with word embeddings for news recommendation. Ai et al. [1] learn embedding of users and items by the method of TransE [3] and the recommendation is based on user-item similarity score in the projected space.

## 6 CONCLUSION

In this paper, we have presented a neural factorization method for the task of offer recommendation with different representations of users and offers. We have shown that our models perform better than the tree-based methods. We have also shown that the learned graph-based user and offer embeddings capture deeper and indirect connections between users and offers, which helps to improve the quality of recommendation. The incorporation of temporal features involving transaction sequences improves the performance further in some cases.

## ACKNOWLEDGMENTS

This research was supported by Capillary Technologies International Pte Limited. We thank Dr. Subrat Panda and Mr. Jyotiska Bhattacharjee from Capillary Technologies International Pte Limited who provided insight and expertise that greatly assisted the research.

## REFERENCES

[1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 11(9):137, 2018.
[2] Gaurangi Anand, Auon H Kazmi, Pankaj Malhotra, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Deep temporal features to predict repeat buyers. In *NIPS 2015 Workshop: Machine Learning for eCommerce*, 2015.
[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.
[4] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
[5] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.
[7] Wan-Hsun Hu, Shih-Hsien Tang, Yin-Che Chen, Chia-Hsuan Yu, and Wen-Cheng Hsu. Promotion recommendation method and system based on random forest. In *Proceedings of the 5th Multidisciplinary International Social Networks Conference*, page 11. ACM, 2018.
[8] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, 2015.
[9] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.
[10] Vladimir Nikulin. Prediction of the shoppers loyalty with aggregated data streams. *Journal of Artificial Intelligence and Soft Computing Research*, 6(2):69–79, 2016.
[11] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Dkn: Deep knowledge-aware network for news recommendation. *arXiv preprint arXiv:1801.08284*, 2018.
[12] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.
[13] Yandi Xia, Giuseppe Di Fabbrizio, Shikhar Vaibhav, and Ankur Datta. A content-based recommender system for e-commerce offers and coupons. In *eCOM@SIGIR*, 2017.
[14] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 729–732. ACM, 2016.
[15] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 353–362, 2016.