Computing and Informatics, Vol. 45, 2026, 1–21, doi: 10.31577/cai_2026_1_1

HYBRID INSURANCE RECOMMENDATION ALGORITHM INTEGRATING DEEP NEURAL NETWORKS AND KNOWLEDGE GRAPHS BASED ON MATRIX FACTORIZATION

Mingshi Liu

College of Mathematics and System Science
Shandong University of Science and Technology
Qingdao 266590, China
e-mail: liumingshi@sdust.edu.cn

Xinying Liu*

College of Intelligent Equipment
Shandong University of Science and Technology
Taian 271019, China
e-mail: liuxinying@sdust.edu.cn

Man Qi

School of Engineering, Technology and Design
Canterbury Christ Church University
Canterbury, CT1 1QU, UK
e-mail: man.qi@canterbury.ac.uk

Abstract. To address issues such as data sparsity, overfitting, and the inability to fully extract latent information in traditional recommendation methods, a new insurance product recommendation algorithm is proposed that combines knowledge graphs improved by matrix factorization with deep neural networks (DNN). First, to tackle the issue of sparse data in existing insurance products, an improved knowl-

* Corresponding authors

edge graph recommendation algorithm based on matrix factorization (FunkSVD) is proposed. By training the data through matrix factorization the problem of data sparsity is mitigated. After refining the customer-product matrix, knowledge graph triples are constructed based on customer characteristics and insurance product features such as insured age and coverage. Feature extraction and customer preference prediction are carried out using an alternating learning approach within a multi-task knowledge graph framework. Then, to alleviate issues like local optima and vanishing gradients during recommendation, DNN is applied for further recommendation. A fully connected layer is constructed, and forward propagation and backpropagation algorithms are used to train customer features and product matrices, predicting customer purchasing behavior and generating recommendations. Finally, comparative experimental results show that, compared to other recommendation algorithms such as collaborative filtering and DNN, the proposed algorithm improves accuracy, recall, F1 score, and other metrics. This algorithm not only speeds up recommendations but also improves recommendation quality.

Keywords: Knowledge graph, FunkSVD matrix factorization, DNN, insurance, hybrid recommendation

Mathematics Subject Classification 2010: 62-07

1 INTRODUCTION

With the development of China's economy and the improvement of people's living standards, insurance has been playing an increasingly important role in daily life. Currently, China has become the second-largest insurance market in the world [1]. Although the insurance market is growing, many policyholders still lack sufficient knowledge of insurance products, leading to purchase unnecessary insurance. This not only results in a waste of resources but can also cause conflicts between policyholders and insurance companies during claims processing. Therefore, applying recommendation systems in the insurance field to provide customized insurance products based on individual needs is necessary. At present, most recommendation systems in China are applied in areas such as goods and news, while their application in the insurance market is still limited. Therefore, in order to solve the problem of information overload of insurance products in the era of big data, it is of practical significance to study the system recommendation of insurance products.

Pradhan and Pal studied a content and network-based academic venue recommender system [2]. Seo et al. investigated a video-on-demand recommendation system for online TV under explicit information fusion [3]. Kiran et al. studied the deep neural network optimization recommendation system by learning non-linear potential factors [4]. Da'u et al. improved weighting techniques in deep learning and applied these improvements to recommendation systems [5]. In terms of knowledge graph research, Yan et al. examined the problem of multilingual knowledge graph alignment [6]. Agibetov and Samwald focused on link prediction in knowledge graphs by implementing semantic and structural changes [7]. Xu et al. constructed a knowledge graph based on PubMed, a fundamental resource in the medical field [8].

Zhang et al. designed a model for evaluating knowledge representation learning [9]. Tao et al. proposed a news recommendation algorithm that incorporates knowledge graphs [10]. Quan et al. used knowledge graphs to assist in calculating internal similarities in movies [11]. Xu et al. introduced knowledge graph reasoning techniques into video collaborative filtering recommendations [12].

In the area of insurance products, Seifert studied the connection between statutory health insurance and pharmaceuticals [13]. Albrecher et al. explored asset-liability management for the long-term insurance business [14]. Li and Wu investigated the development of commercial insurance models on the Internet in China in the context of 5G [15]. Tang and Feng examined the relationship between social insurance contributions and the capital-labor ratio of enterprises, providing valuable insights for policy making [16]. Chun et al. improved genetic algorithms through neural network research to establish a car insurance fraud detection model [17]. Ma et al. used an improved k-means clustering algorithm to analyze automotive market intelligence [18].

In terms of DNN applications, Chen et al. proposed introducing DNN to extract word vectors in mobile video recommendations to optimize traditional recommendation algorithms [19]. Xiang studied the application of DNN in acoustics and proposed an improved speech recognition algorithm based on HMM (Hidden Markov Model) and DNN [20].

There is relatively little study on insurance recommendation algorithms in China. To better integrate product characteristics and customer characteristics into recommendations and alleviate the impact of data sparsity on recommendations, this paper proposes an improved insurance product recommendation algorithm that combines matrix factorization (FunkSVD) with knowledge graphs and deep neural networks (DNN). The algorithm first uses FunkSVD in matrix factorization to complete the customer-product matrix. It then extracts relevant features to construct triples based on the data and employs a multi-task framework of knowledge graphs to alternately learn for feature extraction and predicting customer preferences. DNN is then used for recommendations, constructing fully connected layers, and utilizing forward and backward propagation algorithms to train customer features and the product matrix, ultimately predicting customer purchasing behavior. The DNN approach partially addresses the local optimum problem in the recommendations and overcomes the gradient vanishing issue during the prediction. Finally, recommendations are generated.

The main contributions of this paper are as follows.

1. Proposing an Improved Insurance Product Recommendation Algorithm: This paper presents a recommendation algorithm that integrates an enhanced knowledge graph with FunkSVD. Using FunkSVD for matrix factorization alleviates data sparsity issues and incorporates product features and customer characteristics from the knowledge graph. This approach takes into account the specific traits of insurance products and customers during recommendations, effectively addressing the cold-start problem to some extent.
2. Combining DNN with Knowledge Graphs for Recommendations: The algorithm uses DNN in conjunction with knowledge graphs. Based on the feature training from the knowledge graph, DNN is employed to predict customer purchasing preferences. As the layers of the neural network increase, this approach partially resolves the local optimum problem during recommendations and overcomes the gradient vanishing issue.
3. Experimental Results: The experimental results demonstrate that the proposed hybrid recommendation algorithm, which combines deep neural networks with an improved knowledge graph based on FunkSVD, effectively enhances the quality of recommendations.

This paper is organized as follows. Section 2 introduces the relevant theoretical knowledge. Section 3 presents the hybrid recommendation algorithm for insurance products based on the improved knowledge graph with DNN and FunkSVD. And Section 4 presents experimental results, analysis, and verification of the effectiveness of the proposed method. Section 5 concludes the paper.

2 RELATED THEORIES

2.1 FunkSVD Theory

In recommendation systems, customer ratings for items are often sparse, and data sparsity can lead to inaccuracies in recommendation results. A popular approach to accurately predict unknown data using existing data through matrix factorization aims to achieve matrix completion. Common methods of matrix factorization include Singular Value Decomposition (SVD), BiasSVD, TimeSVD, and FunkSVD algorithms. SVD requires the matrix to be dense, whereas actual rating matrices are mostly sparse. Additionally, SVD decomposition produces three matrices, which can impact computational efficiency. BiasSVD is an improvement based on SVD, implemented under the condition that certain customers tend to give positive ratings to products or that some products are more likely to be favored, and it incorporates several constraints. TimeSVD assumes that customer interests change dynamically over time and requires a time condition. FunkSVD, on the other hand, is an improvement over SVD technology, decomposing the customer rating matrix into two low-dimensional matrices (customer and item matrices) and addressing the issues of computational efficiency and data sparsity associated with SVD [21, 22]. Therefore, this paper utilizes FunkSVD for matrix factorization training.

2.2 Collaborative Filtering Recommendation

In recommendation systems, Collaborative Filtering (CF) is a widely used recommendation algorithm [23, 24]. The core idea of the collaborative filtering recommendation algorithm is to analyze customer characteristics based on their behaviors such as purchase data and rating information, and then identify a group of neighboring customers whose characteristics are similar to those of the target customer. By integrating the behavioral characteristics of these neighboring customers, the algorithm predicts the behavior of the target customer and generates recommendations.

Collaborative filtering algorithms can be primarily divided into two categories: Item-based and User-based recommendations. The implementation process of collaborative filtering recommendations consists of two main steps:

1. Identify a neighbor set of similar customers (or items) based on customer (or item) characteristics.
2. Generate recommendations based on the behaviors of neighboring customers.

There are three main methods for calculating similarity: Cosine Similarity, Pearson Correlation Similarity, and Euclidean Distance Similarity. Among these, Pearson Correlation is the most commonly used. The "nearest neighbor" selection methods primarily include threshold-based methods and TOP-N methods, with the TOP-N method being more frequently employed.

2.3 Knowledge Graph

Google introduced the concept of the Knowledge Graph. Currently, most knowledge graphs are constructed in a bottom-up manner [25, 26]. Entities are extracted from existing data, and those with high confidence are added to the database to establish connections between entity keys. The structure of a knowledge graph includes its logical structure and architecture, as shown in Figure 1.

The essence of a knowledge graph is a semantic network that reveals the potential relationships between entities, allowing for a structured semantic knowledge base. The basic building blocks of a knowledge graph consist of triples formed by "entity-relation-entity", along with entities and their corresponding attribute-value pairs. The "relation" connects different entities, creating a networked structure of knowledge through these connections.

In knowledge representation, a knowledge graph is represented using a triple $G = (E, R, S)$. Here, $E = \{e_1, e_2, e_3, \ldots, e_{|E|}\}$ represents the set of all different entities in the knowledge base, containing $|E|$ distinct entities, and $R = \{r_1, r_2, r_3, \ldots, r_{|R|}\}$

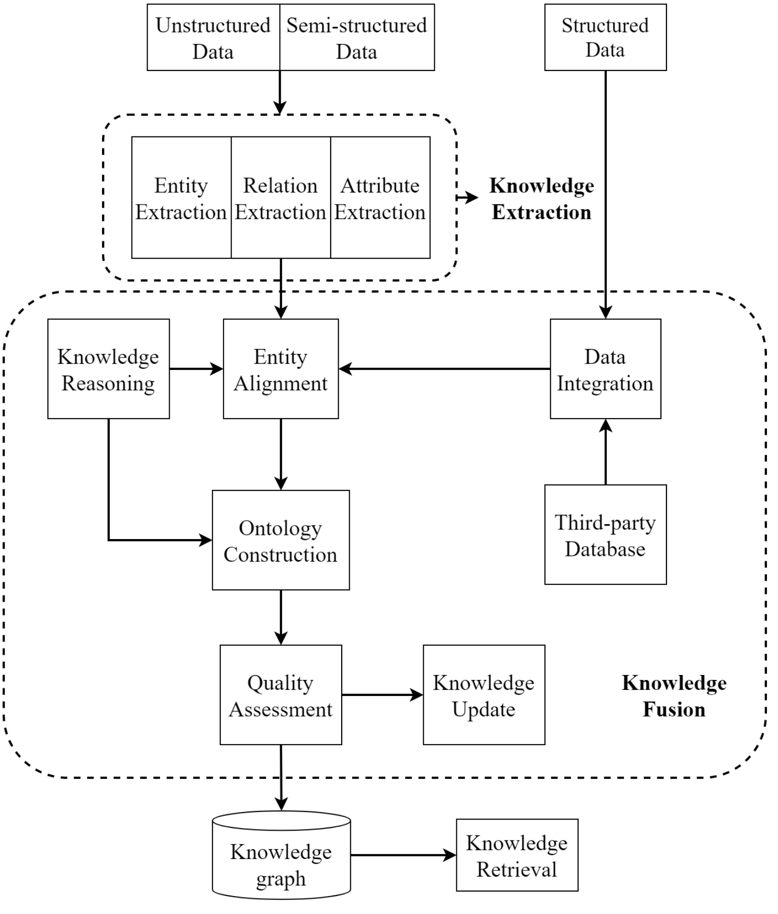

Figure 1. Knowledge graph architecture

denotes the set of all different relations, including $|R|$ different types of relations. $S \subseteq E \times R \times E$ represents the set of all triples. Different entities are identified by unique ID in the knowledge base, and attribute-value pairs depict the various inherent characteristics among entities. Thus, relations are used to link two entities to describe associations between them. Taking the insurance market as an example, as shown in Figure 2, Product 9 and Death Benefit are two distinct entities, forming an entity-relation-entity triple as "Product 9 – Coverage Scope – Death Benefit". "Product 9" is an entity, "Insured Age" is an attribute, and "0∼17 years" is the corresponding attribute value for "Insured Age". Thus, "Product 9 – Insured Age – 0∼17 years" forms an entity-attribute-attribute triple value.

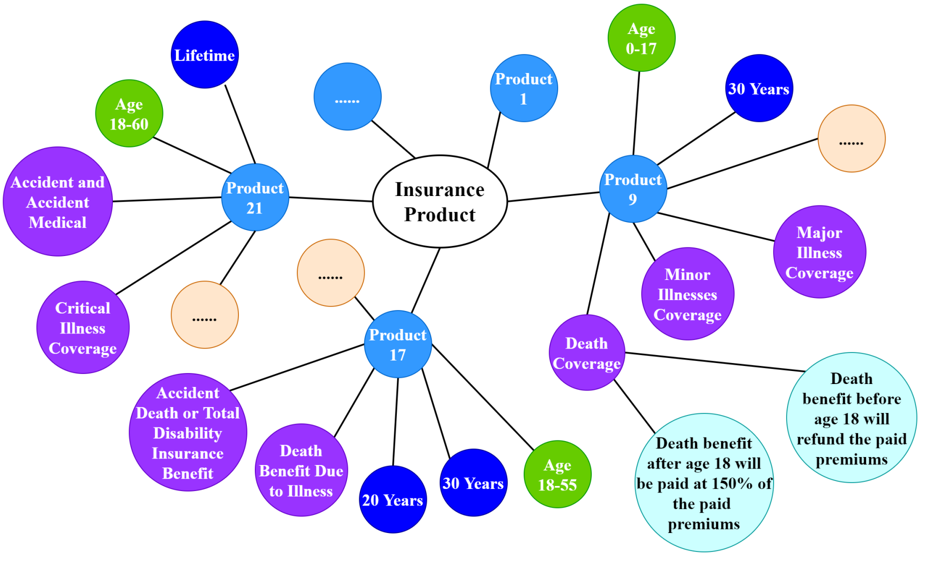

Figure 2. Example of knowledge graph

The architecture of a knowledge graph comprises two parts: its logical structure and its system architecture. The logical structure is divided into two levels: the schema layer and the data layer. The data layer stores knowledge in terms of facts, commonly using methods such as triples and graph databases. The schema layer further standardizes the expression of facts based on the data layer, making the knowledge base more structured and resulting in a knowledge base that is strong in structure and low in redundancy.

The system architecture of a knowledge graph consists of four components. Knowledge extraction involves retrieving necessary entities and other elements from big data for the knowledge base. Knowledge fusion processes the extracted elements and entities to eliminate ambiguities and improve the quality of the knowledge base. Knowledge reasoning further explores potential relationships and expands the knowledge base. Knowledge representation forms a comprehensive vector, which is significant for the construction of the knowledge base and for knowledge reasoning.

2.4 DNN

The precursor to neural networks is the perceptron, which originated in the mid-20th century. The perceptron model is relatively simple and cannot handle more complex nonlinear models, which limits its application. Neural networks represent a further development of the perceptron.

First, neural networks incorporate more hidden layers than perceptrons, enhancing the model's capability and complexity. Second, while the perceptron has multiple inputs corresponding to a single output, neural networks have multiple inputs corresponding to multiple outputs, improving the model's descriptive power. Lastly, neural networks have expanded the activation functions, including functions such as the Sigmoid function, tanh function, and Softmax function.

Deep Neural Networks (DNN) are neural networks with more hidden layers [27, 28]. Typically, the layers of a DNN are divided into three categories: the first layer is the input layer, the last layer is the output layer, and the middle layes are hidden layers. Each layer is fully connected, meaning that every neuron in one layer is connected to every neuron in the next layer. However, from a local perspective, it resembles a small perceptron, consisting of a linear relationship and an activation function.

In neural networks, regularization techniques such as Dropout are commonly used to improve generalization capabilities and prevent overfitting.

3 HYBRID RECOMMENDATION ALGORITHM FOR INSURANCE PRODUCTS BASED ON IMPROVED KNOWLEDGE GRAPH WITH DNN AND FUNKSVD

Most policyholders do not have an in-depth understanding of all insurance products when purchasing insurance, which leads to a cold start problem for the recommended insurance products. To address this, we utilize knowledge graph-based recommendations by constructing customer features and product features to create a knowledge graph. To alleviate the impact of data sparsity on recommendations, this paper proposes using FunkSVD matrix factorization to enhance the knowledge graph. Finally, DNN is integrated to predict customer purchasing preferences based on the training results from the knowledge graph, helping to mitigate issues related to local optima and gradient vanishing.

3.1 Matrix Factorization

In recommendation scenarios, the data is often sparse, meaning that for some products, we are unsure whether the customer has made a purchase. This missing data is represented by a "?", and during the recommendation process, matrix factorization is used to complete the data, further mitigating the sparsity issue when recommending through a knowledge graph. This paper employs FunkSVD for matrix factorization. The core idea of FunkSVD is that customer preferences are influenced by only a few factors. Therefore, the high-dimensional customer rating matrix R (the User-Item rating matrix) is decomposed into two low-dimensional matrices. By learning from the User and Item rating information, the customer feature matrix P and product feature matrix Q are obtained. These two low-dimensional matrices are then used to predict customer ratings, as shown in Equation (1).

$$R_{m\times n} = P_{m\times k} \times Q_{k\times n}. \tag{1}$$

Assuming that the rating of customer u for a certain item i is r_{ui}, through matrix factorization, the implicit feature vector of customer u projected into a k-dimensional space is p_u, and the implicit feature vector of item i is q_i. The dot product of p_u and q_i is used to approximate the original value before matrix factorization, representing the customer's interest in the item, as shown in Equation (2).

$$\overset{\wedge}{r_{ui}} = q_i^T p_u. \tag{2}$$

Let r_{ui} represent the actual rating and $\hat{r_{ui}}$ represent the predicted rating. The objective function can be expressed as shown in Equation (3).

$$\min_{q^*,p^*} \sum_{(u,i)ek} \left(r_{ui} - q_i^T p_u\right)^2 + \lambda\left(\|p_u\|^2 + \|q_i\|^2\right), \tag{3}$$

where λ is the regularization coefficient, and k is the set of customer-item pairs with existing ratings. In the equation, the term in the first parentheses represents the loss function, which is used to control the model's bias. The term in the second parentheses is the regularization term, which controls the model's variance to prevent overfitting.

By taking the derivative of Equation (3) with respect to p_u and q_i, we obtain Equations (4) and (5).

$$\frac{\partial J}{\partial p_u} = -2\left(r_{ui} - q_i^T p_u\right) q_i + 2\lambda p_u, \tag{4}$$

$$\frac{\partial J}{\partial q_i} = -2\left(r_{ui} - q_i^T p_u\right) p_u + 2\lambda q_i. \tag{5}$$

By iteratively updating p_u and q_i, we obtain Equations (6) and (7).

$$p_u = p_u + \alpha\left[\left(r_{ui} - q_i^T p_u\right) q_i - \lambda p_u\right], \tag{6}$$

$$q_i = q_i + \alpha\left[\left(r_{ui} - q_i^T p_u\right) p_u - \lambda q_i\right]. \tag{7}$$

The final result yields P and Q for prediction.

3.2 Knowledge Graph Learning

Regarding the knowledge graph learning section, feature extraction and prediction are carried out using an alternating learning method within a multi-task framework. The model mainly consists of a recommendation module, a Knowledge Graph Embedding (KGE) module, and a cross-compression unit. Regarding the knowledge graph, this paper uses an alternating training method, with the specific model shown in Figure 3.

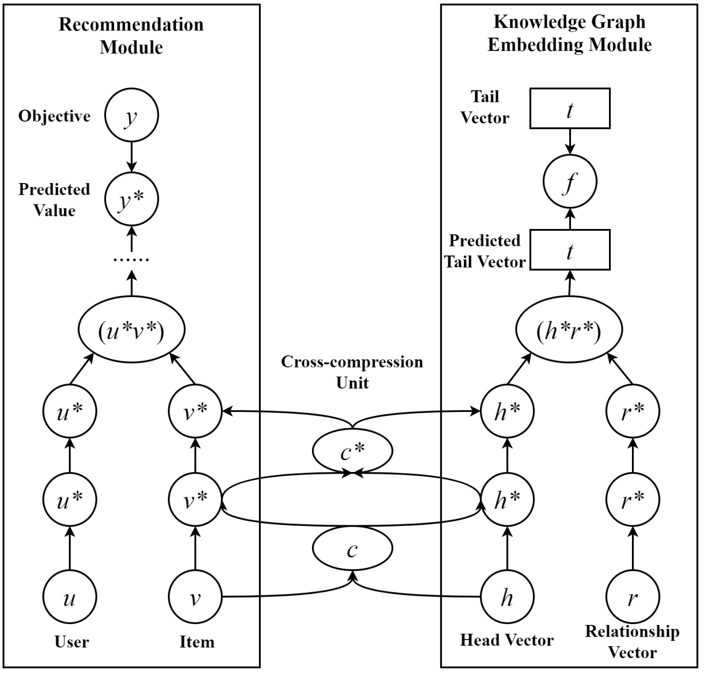

Figure 3. MKR framework

1. Recommendation Module: The customer and product are input into the system, and their respective features are extracted through a Multi-Layer Perceptron (MLP) and the cross-compression unit. The extracted features are then passed to another perceptron to obtain the predicted probability.

   In the recommendation module of MKR, vectors u and v are used to represent customer U and item V. Depending on the application, u and v can be characterized using one-hot ID, attributes, bag-of-words, or a combination of these. The latent compressed features of the customer are extracted from vector u through an L-layer MLP, as shown in Equations (8) and (9).

$$u_l = M(M(\ldots M(u))) = M^L(u), \tag{8}$$

$$M(x) = \sigma(\omega x + b), \tag{9}$$

   where $M(x)$ is a fully connected neural network and ω and b represent weights and biases. $\sigma(x)$ is a non-linear activation function.

   The features of item V are extracted using L cross-compression units, where $S(v)$ denotes the set of entities related to V.

$$\nu_L = E_{e\sim S(v)}\left[C^L(v,e)[v]\right]. \tag{10}$$

After obtaining the latent features of the customer and the item u_L and v_L, the final probability of customer U engaging with item V is calculated using a prediction function f_{RS} that combines internal machine or MLP, as shown in Equation (11).

$$\hat{y}_{uv} = \sigma(f_{RS}(u_L, \nu_L)). \tag{11}$$

2. KGE module: The KGE module extracts features of h and r from the triple G by MLP. It is supervised by the scoring function f and the real tail, outputting a representation of the predicted tail.

   Knowledge Graph Embedding (KGE) embeds entities and relationships in continuous vector space while preserving their original structure, utilizing a deep semantic matching framework in MKR. For a given triple G, the original feature vectors of h and r (such as ID, type, textual description, etc.) are processed through multiple cross-compression units and non-linear layers, after which their latent features are concatenated. Finally, a k-layer MLP is used to predict the tail t, as shown in Equations (12), (13) and (14).

$$h_L = E_{\nu \sim S(h)} \left[C^L(\nu, h)[e] \right], \tag{12}$$

$$r_L = M^L(r), \tag{13}$$

$$\hat{t} = M^K \left(\begin{bmatrix} h_L \\ r_L \end{bmatrix} \right), \tag{14}$$

   where $S(h)$ is a set associated with the head entity h and $\hat{t}$ is the prediction vector of the tail vector t. Finally, the score of the triple G is calculated by the similarity scoring function in Equation (15).

$$score(h, r, t) = f_{KG}(t, \hat{t}). \tag{15}$$

3. Cross-Compression Unit: This unit is key to linking (1) and (2), enabling the automatic learning of higher-order interaction features between items in the recommendation system and entities in the knowledge graph (KG).

   In this unit, more information is obtained by crossing item vectors and entity vector, addressing the issue of data sparsity. For each item V and its associated entities, we first construct $d \times d$ pairwise interaction latent vectors $v_1 \in R^d$ and $e_1 \in R^d$ at layer 1, forming a cross matrix, as shown in Equation (16).

$$C_l = \nu_l e_l^T = \begin{bmatrix} \nu_l^{(1)} e_l^{(1)} \dots \nu_l^{(1)} e_l^{(d)} \\ \dots \\ \nu_l^{(d)} e_l^{(1)} \dots \nu_l^{(d)} e_l^{(d)} \end{bmatrix}, \tag{16}$$

   where C_1 is the cross feature matrix at layer 1, d is the dimension of hidden layer, e is the entities associated with item v. Then, compression is performed and the next layer's v and e based on words, as shown in Equations (17) and (18).

$$\nu_{l+1} = C_l \omega_l^{VV} + C_L^T \omega_l^{EV} + b_l^V = \nu_l e_l^T \omega_l^{VV} + e_l \nu_l^T \omega_l^{EV} + b_l^V, \tag{17}$$

$$e_{l+1} = C_l \omega_l^{VE} + C_L^T \omega_l^{EE} + b_l^E = \nu_l e_l^T \omega_l^{VE} + e_l \nu_l^T \omega_l^{EE} + b_l^E, \tag{18}$$

The cross-compression unit is shown in Equation (19).

$$[v_{l+1}, e_{l+1}] = C(v_l, e_l). \tag{19}$$

3.3 DNN Algorithm

In the construction of the DNN model, the forward propagation algorithm of the DNN is first used to sequentially obtain relevant parameters and output values. This involves performing linear operations and activation function operations using the weight matrix ω, bias vector b, and input value x, starting from the input layer and continuing until the output value is obtained as the result.

To mitigate the risk of overfitting, we use Dropout and the ReLU activation function. In each training session, a random selection of neurons from the input and hidden layers is dropped with a probability p, remaining inactive until the end of that training session. However, in the next training session, a new set of neurons is selected to drop, meaning neurons dropped in the previous session might be re-included or dropped again. Here, p is the dropout rate, typically set to 0.5. To address the vanishing gradient problem, the ReLU function is commonly used for activation, as shown in Equation (20): if the value is less than 0, it outputs 0; otherwise, it remains unchanged.

$$Relu\,(z) = \max(0, z). \tag{20}$$

The gap between the predicted output and the actual data is calculated using a loss function. In logistic regression problems, Mean Squared Error (MSE) is commonly used as the loss function, as shown in Equation (21).

$$loss = \frac{1}{2m} \sum_{i=1}^{m} (y_i - y_{i\_})^2, \tag{21}$$

where y_i represents the predicted output, $y_{i\_}$ represents the actual data, and m denotes the total number of samples. The loss function for the samples is denoted as $loss$. However, experiments have shown that this is a non-convex function, and in some intervals, small gradients can make training difficult. The cross-entropy formula is shown in Equation (22).

$$loss = \sum_{i=1}^{m} y_i \log(y_{i\_}). \tag{22}$$

Experiments have shown that the curve of this function is generally monotonic: the larger the loss, the larger the gradient, which facilitates backpropagation with gradient descent. Therefore, we choose cross-entropy as the loss function. Before calculating the loss function, the predicted output obtained through forward propagation is transformed into a probability distribution using Softmax regression, and then the distance between the probability distributions of the predicted output and the actual data is calculated. Assuming the predicted output is $y_1, y_2, \ldots, y_n$, it becomes a probability distribution after Softmax regression, as shown in Equation (23).

$$softmax = \frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_i}}. \tag{23}$$

Then, the gradients are updated through backpropagation to adjust the weight parameters and minimize the value of the loss function. After obtaining the predicted output, the accuracy of that output is calculated, and the above process is repeated for iteration. This continues until the accuracy no longer shows significant improvement, at which point the optimal recommendation results are output.

3.4 Model Training

The hybrid recommendation algorithm model based on matrix factorization optimization, knowledge graph, and DNN (F-KG-DNN) consists of three main processes.

First is the matrix factorization part. Based on the existing sparse data of customer insurance product purchases, the data is divided into a customer feature matrix, a product feature matrix, and a customer rating matrix (where a purchase of a product is marked as “1” and non-purchase as “0”). FunkSVD is used to predict the sparse data, resulting in a complete customer feature matrix, product feature matrix, and customer rating matrix.

Using the optimized matrices from matrix factorization, features are extracted and predictions are made through the knowledge graph using an alternating learning method within a multi-task framework. The loss function is shown in Equation (24).

$$\begin{aligned} L &= L_{RS} + L_{KG} + L_{REG} \\ &= \sum_{u\in U, v\in V} J(y_{u\nu}, y_{u\nu}) - \lambda_1 \left(\sum_{(h,r,t)\in G} score(h,r,t) - \sum_{(h',r',t')\notin G} score(h',r,t') \right) \\ &\quad + \lambda_2 \|\omega\|_2^2. \end{aligned} \tag{24}$$

In this equation, the first term is the cross-entropy loss of the recommendation module, where U and V represent the sets of customers and items, respectively.

The second term is the loss of the KGE module, which is used to increase the scores of correct triples while decreasing the scores of incorrect triples. The third term is the regularization term, which prevents overfitting. λ_1 and λ_2 are the balancing constant.

Finally, the data obtained from the knowledge graph is input into the DNN for training. The data is fed into the model for forward propagation, and then backpropagation is used to update the gradients and adjust the parameters, resulting in a prediction function. Ultimately, a recommendation list is output.

4 EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Data Sets

This study conducts experimental analysis of the proposed algorithm using information from 133 customers of an insurance company. The customer information includes eight customer features such as gender, age, marital status, and certain purchase histories of products. Due to the inherent characteristics of insurance products, it is challenging to obtain ratings from insurance customers for these products. Therefore, we use 0 and 1 to indicate the purchase status of a product, where 0 represents "not purchased" and 1 represents "purchased", while "?" indicates unknown data, as shown in Table 1. Here, C1 denotes Customer 1, P1 denotes Product 1, and so on.

	C1	C2	C3	C4	C5	C6
P1	1	1	1	1	0	1
P2	1	0	0	0	0	0
P3	1	?	?	?	?	0
P4	0	1	0	?	0	1
P5	0	0	0	0	0	0
P6	1	0	?	?	0	1
P7	0	0	0	0	0	0
P8	0	1	1	0	0	0
P9	0	0	0	1	0	0
P10	0	0	0	0	1	0
P11	0	0	0	0	0	1
P12	0	0	0	0	0	1

Table 1. Partial customer product table

Regarding the customer feature information for DNN recommendations, the personal features are defined as follows: for gender, male is represented as 1 and female as 2; for age, 0–9 years is 0, 10–11 years is 1, and so forth; for marital status, unmarried is 1, married is 2, widowed is 3, divorced is 4, and unspecified is 5; for education level, graduate is 0, bachelor's degree is 1, associate degree is 2, secondary vocational is 3, technical school is 4, high school is 5, junior high school is 6, and elementary school is 7. Other information is represented as either 1 or 0 to indicate whether a customer has purchased a certain product or possesses a certain feature, as shown in Table 2, where T1 represents Feature 1, and so on.

	C11	C12	C13	C14	C15	C16	C17
T1	2	2	2	2	1	2	2
T2	3	3	2	3	2	2	3
T3	5	6	3	3	5	2	3
T4	2	2	2	2	2	2	2
T5	5	4	5	4	3	3	5
T6	3	2	3	4	4	4	5
T7	1	3	5	4	3	4	5
T8	4	3	4	4	3	4	5

Table 2. Partial customer features

4.2 Calculation of Recommended Accuracy

To verify the effectiveness of a recommendation algorithm, the recommendation accuracy needs to be examined, which is achieved by comparing the predicted results with the actual outcomes. Recommendation accuracy is generally evaluated using three metrics: Precision, Recall, and the F-measure, which considers both Precision and Recall. The expressions for these three metrics are shown in Equations (25), (26) and (27).

$$precision = \frac{\sum_{\mu \in u} |R(u) \cap T(u)|}{\sum_{\mu \in u} |R(u)|}, \tag{25}$$

$$recall = \frac{\sum_{\mu \in u} |R(u) \cap T(u)|}{\sum_{\mu \in u} |T(u)|}, \tag{26}$$

$$F = \frac{2 \times precision \times recall}{precision + recall}, \tag{27}$$

where $R(u)$ represents the products recommended to the customer in the recommendation list, and $T(u)$ represents the products actually purchased by the customer in the test set. In real recommendation scenarios, the higher the Precision and Recall, the better. However, this is often difficult to accomplish, so we typically use the F-measure as an indicator to evaluate the quality of a recommendation algorithm.

4.3 Recommendation Accuracy Analysis

To validate the effectiveness of the improved algorithm (F-KG-DNN) proposed in this section, we compare it with several traditional algorithms: collaborative filtering (CF), hybrid recommendation (a combination of collaborative filtering and content-based recommendation, CFCB), and a collaborative filtering algorithm that integrates DNN and time-weighting (CF-T-DNN). By comparing the Precision, Recall, and F-measure at different values of N, we assess the performance of the improved algorithm.

Precision	CF	CFCB	CF-T-DNN	F-KG-DNN
$N = 3$	0.345	0.395	0.442	0.57
$N = 5$	0.291	0.334	0.391	0.3
$N = 10$	0.23	0.285	0.322	0.3

Table 3. Comparison of Precision

The comparison of Precision at different N values under various recommendation methods is shown in Table 3. Table 3 indicates that, compared to other individual collaborative filtering or hybrid recommendation methods, the algorithm proposed in this paper achieves higher Precision.

Recall	CF	CFCB	CF-T-DNN	F-KG-DNN
$N = 3$	0.717	0.666	0.598	0.66
$N = 5$	0.806	0.76	0.629	0.75
$N = 10$	0.801	0.734	0.647	0.84

Table 4. Comparison of Recall

The comparison of Recall at different N values under various recommendation methods is shown in Table 4. Through this comparison, we find that, unlike Precision, the algorithm improvement did not yield significant gains in Recall; in fact, Recall decreases as the method improves. Thus, considering only Precision and Recall does not effectively evaluate the advantages of the algorithm improvement. Therefore, we use the F-measure as a standard to assess the quality of the improved algorithm, taking both Recall and Precision into account to evaluate the algorithm's feasibility, as shown in Table 5.

F-Meausre	CF	CFCB	CF-T-DNN	F-KG-DNN
$N = 3$	0.465	0.495	0.508	0.611
$N = 5$	0.427	0.464	0.482	0.43
$N = 10$	0.357	0.41	0.429	0.44

Table 5. Comparison of F-Measure

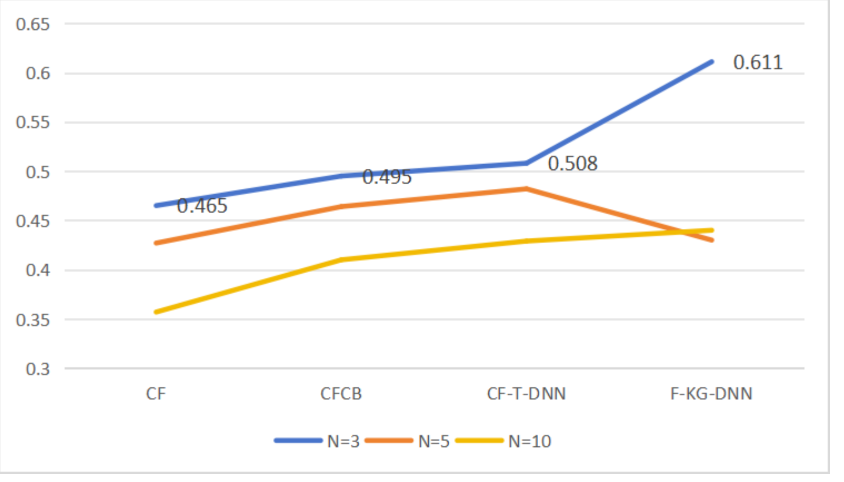

Figure 4. Comparison of F-Measure

From Table 5 and Figure 4, we can see that the F-measure of the proposed algorithm in this paper shows an improvement over other algorithms, which demonstrates that the improved hybrid recommendation algorithm enhances recommendation quality to a certain extent. This indicates that the algorithm proposed, using FunkSVD to address data sparsity issues, leverages knowledge graph recommendations to strengthen the relationship between customer features and insurance products during recommendation. The proposed algorithm addresses overfitting and local optima issues, improves recommendation quality to a certain extent and has practical application value.

5 CONCLUSION

In this paper, the recommendation algorithm (F-KG-DNN) fully considers the relationship between features of insurance products, such as coverage and eligible age range, and customer characteristics when recommending insurance products. It leverages knowledge graphs to uncover potential relationships, uses FunkSVD matrix factorization to alleviate data sparsity issues, and optimizes the recommendation results. Additionally, by integrating DNN, the algorithm effectively addresses overfitting and local optima issues in the recommendation process, generating the final recommendation based on DNN prediction results and improving the recommendation effectiveness.

Although the recommendation results show a certain level of improvement, there is still room for enhancement. Accurate insurance product recommendations require precise customer profiling. Moreover, in real life, different insurance products may have underlying connections. Even if two products appear unrelated, customers who purchase one may be inclined to purchase the other – like the well-known example of beer and diapers. However, this study has not fully explored such connections. Therefore, thoroughly uncovering the relationships between products and developing product recommendation models based on precise customer profiling will be directions for future research.

Acknowledgement

This research is supported by National College Students innovation training program of China under Grant No. 202510424061.

REFERENCES

[1] Liu, F.—Zhang, Y.—Li, Y.: 70 Years of Legal Construction of Insurance Supervision in China: Review and Prospect. Insurance Studies, Vol. 26, 2019, No. 9, pp. 3–10, doi: 10.13497/j.cnki.is.2019.09.001 (in Chinese).

[2] Pradhan, T.—Pal, S.: CNAVER: A Content and Network-Based Academic VEnue Recommender System. Knowledge-Based Systems, Vol. 189, 2020, Art. No. 105092, doi: 10.1016/j.knosys.2019.105092.

[3] Seo, Y. D.—Lee, E.—Kim, Y. G.: Video on Demand Recommender System for Internet Protocol Television Service Based on Explicit Information Fusion. Expert Systems with Applications, Vol. 143, 2020, Art. No. 113045, doi: 10.1016/j.eswa.2019.113045.

[4] Kiran, R.—Kumar, P.—Bhasker, B.: DNNRec: A Novel Deep Learning Based Hybrid Recommender System. Expert Systems with Applications, Vol. 144, 2020, Art. No. 113054, doi: 10.1016/j.eswa.2019.113054.

[5] Da'u, A.—Salim, N.—Rabiu, I.—Osman, A.: Weighted Aspect-Based Opinion Mining Using Deep Learning for Recommender System. Expert Systems with Applications, Vol. 140, 2020, Art. No. 112871, doi: 10.1016/j.eswa.2019.112871.

[6] Yan, Z.—Peng, R.—Wang, Y.—Li, W.: CTEA: Context and Topic Enhanced Entity Alignment for Knowledge Graphs. Neurocomputing, Vol. 410, 2020, pp. 419–431, doi: 10.1016/j.neucom.2020.06.054.

[7] Agibetov, A.—Samwald, M.: Benchmarking Neural Embeddings for Link Prediction in Knowledge Graphs Under Semantic and Structural Changes. Journal of Web Semantics, Vol. 64, 2020, Art. No. 100590, doi: 10.1016/j.websem.2020.100590.

[8] Xu, J.—Kim, S.—Song, M.—Jeong, M.—Kim, D. et al.: Building a PubMed Knowledge Graph. Scientific Data, Vol. 7, 2020, Art. No. 205, doi: 10.1038/s41597-020-0543-2.

[9] Zhang, X.—Sun, W.—Wang, H.: Evaluation of Knowledge Credibility Based on Knowledge Representation Learning. Computer Engineering, Vol. 47, 2021, No. 7, pp. 44–54, doi: 10.19678/j.issn.1000-3428.0057891 (in Chinese).

[10] Tao, T.—Wang, Q.—Yuwei, F.—Jian, X.—Feng, Y.—Bo, Y.: Personalized Recommendation Algorithm for Financial News Based on Knowledge Graph.

Computer Engineering, Vol. 47, 2021, No. 6, pp. 98–103, doi: 10.19678/j.issn.1000-3428.0057446 (in Chinese).

[11] Yuan, Q.—Cheng, Z.—Jiang, Y.: A Study on Film Recommendation Algorithm Based on Knowledge Graph and Collaborative Filtering. Computer Engineering and Science, Vol. 42, 2020, No. 4, pp. 714–721, doi: 10.3969/j.issn.1007-130X.2020.04.019 (in Chinese).

[12] Xu, Z.—Zhao, X.—Dong, Y.—Yan, W.: Video Recommendation Algorithm Based on Knowledge Graph Knowledge Reasoning. Computer Engineering and Design, Vol. 41, 2020, No. 3, pp. 710–715, doi: 10.16208/j.issn1000-7024.2020.03.018 (in Chinese).

[13] Seifert, W.: "Be Right"' Vs. "Get One's Rights"': Practice-Oriented Recommendations from a Legal Viewpoint. Zeitschrift für Rheumatologie, Vol. 77, 2018, No. 7, pp. 576–586, doi: 10.1007/s00393-018-0483-9 (in German).

[14] Albrecher, H.—Bauer, D.—Embrechts, P.—Filipović, D.—Koch-Medina, P.—Korn, R.—Loisel, S.—Pelsser, A.—Schiller, F.—Schmeiser, H.—Wagner, J.: Asset-Liability Management for Long-Term Insurance Business. European Actuarial Journal, Vol. 8, 2018, No. 1, pp. 9–25, doi: 10.1007/s13385-018-0167-5.

[15] Li, Y.—Wu, N.: Facing the Background of the 5G Era, the Business Model of Internet Insurance in China Is Studied. Chinese Business Theory, Vol. 21, 2019, pp. 31–32, doi: 10.19699/j.cnki.issn2096-0298.2019.21.031 (in Chinese).

[16] Tang, J.—Feng, J.: The Impact of Social Insurance Contributions on the Capital-Labor Ratio of Enterprises: A Case Study of the Change of Provincial Pension Insurance Collection Institutions in the Early 21st Century. Economic Research, Vol. 54, 2019, No. 11, pp. 87–101 (in Chinese).

[17] Yan, C.—Li, M.—Zhou, X.: Optimized BP Neural Network Auto Insurance Fraud Detection Model Based on Improved Genetic Algorithm. Journal of Shandong University of Science and Technology (Natural Science Edition), Vol. 38, 2019, No. 5, pp. 72–80, doi: 10.16452/j.cnki.sdkjzk.2019.05.009 (in Chinese).

[18] Ma, T.—Liu, T.—Xu, J.—Liu, X.: Competitive Intelligence Analysis of Automobile Market Based on Improved K-Means Clustering Algorithm. Journal of Shandong University of Science and Technology (Natural Science Edition), Vol. 38, 2019, No. 1, pp. 74–84, doi: 10.16452/j.cnki.sdkjzk.2019.01.009 (in Chinese).

[19] Chen, L.—Wang, J.—Wang, N.—Li, X.: DNN Based Mobile Online Video Recommendation. Journal of Computer Science, Vol. 39, 2016, No. 8, pp. 1626–1638, doi: 10.11897/SP.J.1016.2016.01626 (in Chinese).

[20] Yuan, X.: Research and Implementation of Speech Recognition Algorithm Based on HMM and DNN. Master Thesis. Jiangxi University of Technology, 2017.

[21] Klymash, M.—Hordiichuk-Bublivska, O.—Pyrih, Y.—Urikova, O.: A Hybrid Collaborative Filtering Based Recommender Model Using Modified Funk SVD Algorithm. In: Luntovskyy, A., Klymash, M., Melnyk, I., Beshley, M., Schill, A. (Eds.): Digital Ecosystems: Interconnecting Advanced Networks with AI Applications (TCSET 2024). Springer, Cham, Lecture Notes in Electrical Engineering, Vol. 1198, 2024, pp. 255–273, doi: 10.1007/978-3-031-61221-3_12.

[22] Yue, X.—Liu, Q.: Parallel Algorithm of Improved FunkSVD Based on Spark. KSII Transactions on Internet and Information Systems (TIIS), Vol. 15, 2021, No. 5, pp. 1649–1665, doi: 10.3837/tiis.2021.05.004.

[23] Zhang, A.—Ma, W.—Zheng, J.—Wang, X.—Chua, T. S.: Robust Collaborative Filtering to Popularity Distribution Shift. ACM Transactions on Information Systems, Vol. 42, 2024, No. 3, Art. No. 80, doi: 10.1145/3627159.

[24] Shao, P.—Wu, L.—Zhang, K.—Lian, D.—Hong, R.—Li, Y.—Wang, M.: Average User-Side Counterfactual Fairness for Collaborative Filtering. ACM Transactions on Information Systems, Vol. 42, 2024, No. 5, Art. No. 140, doi: 10.1145/3656639.

[25] Liang, K.—Meng, L.—Liu, M.—Liu, Y.—Tu, W.—Wang, S.—Zhou, S.—Liu, X.—Sun, F.—He, K.: A Survey of Knowledge Graph Reasoning on Graph Types: Static, Dynamic, and Multi-Modal. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 46, 2024, No. 12, pp. 9456–9478, doi: 10.1109/TPAMI.2024.3417451.

[26] Liang, W.—De Meo, P.—Tang, Y.—Zhu, J.: A Survey of Multi-Modal Knowledge Graphs: Technologies and Trends. ACM Computing Surveys, Vol. 56, 2024, No. 11, Art. No. 273, doi: 10.1145/3656579.

[27] Xu, R.—Ma, S.—Guo, Y.—Li, D.: A Survey of Design and Optimization for Systolic Array-Based DNN Accelerators. ACM Computing Surveys, Vol. 56, 2024, No. 1, Art. No. 20, doi: 10.1145/3604802.

[28] Hussain, H.—Tamizharasan, P. S.—Rahul, C. S.: Design Possibilities and Challenges of DNN Models: A Review on the Perspective of End Devices. Artificial Intelligence Review, Vol. 55, 2022, No. 7, pp. 5109–5167, doi: 10.1007/s10462-022-10138-z.

Mingshi Liu is now pursuing her Bachelor's degree in the Shandong University of Science and Technology, Qingdao, China. Her research interests include data analysis and statistics.

Xinying Liu is engaged in research work in big data intelligent analysis and processing, workflow, service computing, software formal analysis and verification.

Man Qi is Senior Lecturer in computing at the Canterbury Christ Church University, UK. Her main research interests are in the areas of intelligent systems and applications, data analytics, cyber security and HCI. Her research has been founded by EPSRC, EU and QR etc. She is a Fellow of the British Computer Society (FBCS) and a Fellow of the Higher Education Academy (FHEA). She has published over 70 research papers and is on the editorial board of five international journals. She has been an external Ph.D. examiner for a number of universities in the UK and Australia. She has served as a Chair and Program Committee member for over 50 international conferences and has been a long-term reviewer for many international journals.