Hotel Recommendation System

Nguyễn Ngô Thành Đạt[1], Nguyễn Phước Thắng[2], Huỳnh Văn Tín[3]
[1][2][3]Faculty of Information Science and Engineering, University of Information Technology, Ho Chi Minh City, Vietnam
[4]Vietnam National University, Ho Chi Minh City, Vietnam
[1]21522923@gm.uit.edu.vn
[2]21522590@gm.uit.edu.vn
[3]tinhv@uit.edu.vn

Abstract

In the context of modern research on recommendation systems, leveraging user information becomes a major challenge when facing the problem of limited information. This often leads to results that are not accurate enough and are not effective. In an attempt to solve this problem, many research methods have emerged, focusing on the creative use of user information. However, most of them face difficulties in collecting detailed and accurate information about users. We introduce the **Graph-based Hybrid User Context - GHUC model** to make the most of available information and at the same time solve the problem of cold start and sparse rating matrix. By combining the power of similarity graphs and user intelligence,*GHUC* not only delivers more accurate results but also creates a seamlessly personalized experience for users. We apply this model to a hotel recommendation system and experimental results demonstrate the effectiveness of *GHUC* compared to current state-of-the-art methods, marking a step forward in this field. To facilitate development and transparency in research, we have made the project's source code public on GitHub. You can access and contribute at https://github.com/Sonny-Inkai/Hotel-Recommendation-System.

1 Introduction

Recommendation systems play an important role in recommending products and services by being able to analyze and understand user data and predict their preferences based on past activities. This system is becoming increasingly popular, especially in the tourism sector, when many companies choose it as a tool to optimize customer experience and satisfaction. However, in the fast growing of technology, traditional recommendation systems cannot clearly optimize the results for businesses and companies. While most recommendation systems rely on either *Content-based filtering* or *Collaborative filtering*, combining two of them can greatly improve recommendation accuracy. Although there has been much research on this issue, with the emergence of new algorithms, it is necessary to continue to dig deeper to optimize the results of old research projects.

At present, with the development of the tourism industry as well as the increasingly strong need for accommodation services, online hotel booking is more and more popular. To help users find suitable hotels according to their preferences, we have built a hotel recommendation system using different recommendation methods. In this research paper, our team will focus on introducing the *GHRS (Graph-based hybrid recommendation system)* method to solve cold start problem and sparse matrices, and use *GLocal-K (Global and Local Kernels for Recommender Systems)* to compare results and performance.

Recent research such as *Attribute-aware non-linear co-embeddings of graph features*, *Scalable probabilistic matrix factorization with graph-based priors*, *A Geometric Deep Learning Perspective*, or the *GHRS* method, focuses on using trivial information, such as user opinions or attributes. However, in most real-life contexts (e.g., platforms and websites), there is no, or not enough additional information available about users. To handle this problem, we also further develop the *GHRS* method by applying *User-Context*, which takes advantage of existing data to put user into the corresponding vector space. This improves model performance because user are classified into clusters.

The main goals of our team are:

- Introducing collaborative filtering methods to help build hotel recommendation systems.
- Apply the introduced methods, and develop and propose other methods to handle the problem of sparse data and cold start.

The remainder of the research paper is organized as follows. In part 2, we look back at research projects related to our topic. Section 3 will mention details of the dataset used in the research process. In part 4, we will learn more about the *Graph-based User Context* model used in this study. Finally, conduct experiments and evaluate the model according to the *MSE*, *RMSE*, *MAE*, *NMAE* measures, thereby drawing conclusions in part 5 and 6.

2 Related works

This section will discuss about previous researchs and models applied in hotel reccomendation system. The main purpose of recommendation systems is to predict user interests in a certain product. Recommendation systems are traditionally divided into two methods: *Collaborative Filtering* and *Content-based Filtering*.

2.1 Recommendation system based on Graph-based Hybrid Recommendation System

In the past decade, recommendation systems have evolved from traditional digital approaches to advanced neural network-based models. *GBRS (Graph-based Recommendation System)* - a graph-based recommendation system model has also been recognized in many aspects when put into practice. By using *GHRS*, it is much easier dealing with cold-start problems and sparse rating matrices which collaborative filtering methods often encounte. This method also includes many special points that we have applied to develop the new method mentioned in section 4.

2.2 Recommendation system based on the Autoencoder method

Autoencoder is an unsupervised machine learning model used to refactor the input data in the output layer. Fig. 2 shows how *Autoencoder* works (Zhang et al., 2019). In this picture, X is depicted as the input data. After being processed through layer z, the output X' will be reconstructed data and is similar to the input data X. Almost all variants of *Autoencoder* such as *Noise Reduction Autoencoder*, *Variational Autoencoder* and *Compact Autoencoder* can be used for recommendation systems (Goodfellow et al., 2016). *Autoencoder* method is often applied in the following ways (Zhang et al., 2019):

- Use *Autoencoder* to learn features with smaller dimensions at the innermost layer.
- Use the output data to fill in the interaction matrix at the refactoring layer.

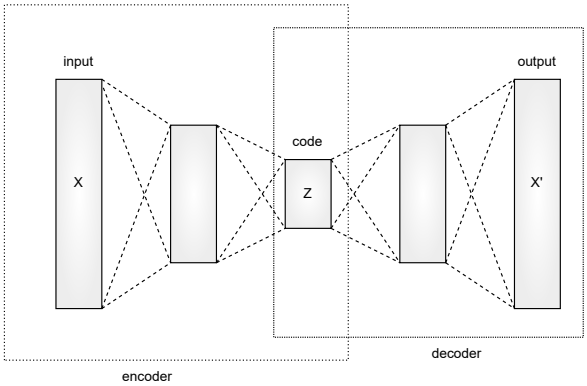

Figure 1: The model shows the structure of the Autoencoder

Using Autoencoder in collaborative filtering Using *Autoencoder* to serve in collaborative filtering is one of the applications considered to be the most effective. The *AutoRec* model (Sedhain et al., 2015) is a good example. It is divided into two categories: *item-based AutoRec (I-AutoRec)* and *user-based AutoRec (U-AutoRec)*. Some things to note about this method include:

- *I-AutoRec* has better performance than *U-AutoRec*.
- Changing the activation functions can affect performance significantly.
- By increasing the dimensions of the hidden layer, *AutoRec* is increased in its ability to model the characteristics of the input data.

3 Data

3.1 Data collection

The dataset used in this study is named '**Hotel Booking Rating Dataset**' and has been published on the Kaggle website (https://www.kaggle.com). This data set was collected by the author from the online booking website *Booking.com*, to serve the research purposes of the topic. Details of the data set of 38,801 data rows, 9 attributes are seen in Figure 3, with 4,506 hotels in 10 different provinces/cities, and 6,471 users.

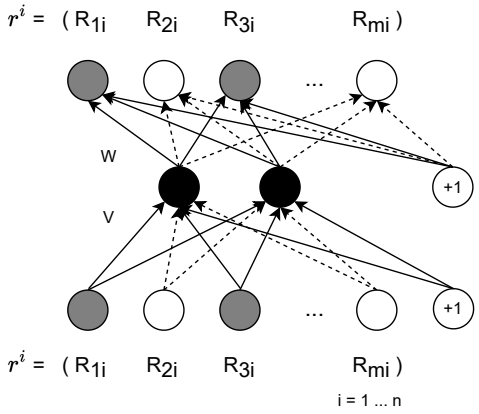

Figure 2: Sơ đồ thể hiện mô hình I-AutoRec

Index	Thuộc tính	Ý nghĩa
0	URL Hotel	URL của khách sạn
1	Location	Địa điểm của khách sạn (Tên tỉnh/TP)
2	HotelID	ID của khách sạn
3	Name Hotel	Mô tả, giới thiệu của khách sạn
4	Descriptions	Mô tả, giới thiệu của khách sạn
5	Address	Địa chỉ của khách sạn
6	UserID	ID của user
7	User	Tên của user (người đánh giá)
8	Ratings	Rating của user

Figure 3: Detailed attributes

3.2 Data analysis

Ratings are evaluated on a scale of 0-10. The ratings in the data set are distributed from 4-10 and are uneven, with the lowest rating being 4 and also the rating with the smallest number (nearly 1,000 ratings), and the highest rating being 10 with nearly 3,000 ratings. The rating range from 6-9 is quite similar (7,000-8,000 ratings) as shown in Figure 4.

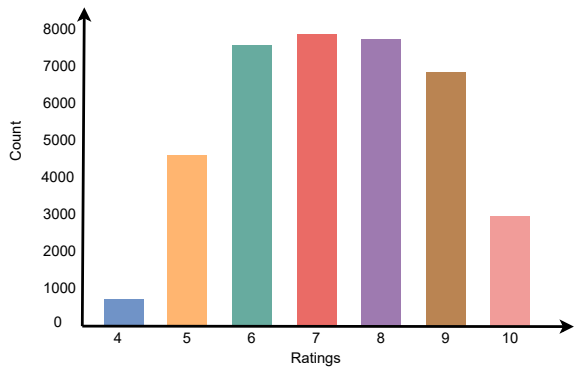

Figure 4: User ratings distribution

3.3 Data processing

Clean data plays an important role in creating highly accurate and generalizable forecasting models. The data set we used encountered the problem of sparse data, where the number of users leaving reviews for the hotel was not much and there were many users who left reviews for just one hotel, which also happened, leading to difficulty in achieving high results when evaluating recommendation systems. In this report, we build a recommendation system focusing on the *Collaborative filtering* method. Therefore, to optimize the suitability of the data, we process the data to create appropriate training and testing sets.

To create a data set for user-based collaborative filtering, we first counted the number of ratings for each user to select users with 30 ratings or more, resulting in 139 satisfied users. Next, we filter out those users from the *Booking Hotels Dataset* to create a new dataset. From this data set, for each different user, we select rows. This means that for each user there will be 5 hotels rated use for testing. The training set is the remaining data in the *Booking Hotels Dataset*. The results obtained are two sets of testing and training with the amount of data being 695 data rows and 38,066 data rows respectively.

4 Graph-based Hybrid User Context (GHUC)

In Figure 5 we can see that the number of users who only travel at one hotel accounts for the majority, the number of users traveling at 2 hotels or more accounts for a smaller part. It can be seen in Table 1 that the level of data density compared to the *MovieLens 100K* is very low. With this very sparse data set, if we apply Link Prediction methods on the graph, we will not get good results due to the lack of relationships between users.

But the *GHRS* method has solved the above problem by clustering users based on the combination of *Similarity Graph* and *User Demographic*. However, collecting additional user information such as age, address, etc. is not easy. Therefore, we have developed a new method to take advantage of information about Hotels that Users have rated to build a Vertor Embedding for Users. This method is called **Graph-based Hybrid User Context (GHUC)**. Based on *GHRS*, we can inherit the methods used by that model to further improve the GHUC model.

Table 1: Comparison data metrics of HotelData dataset and MovieLens 100K

Dataset	Users	Items	Ratings	Density
HotelData	**6471**	**4506**	**38,801**	**0,133%**
MovieLens 100K	943	1,682	100,000	6,304%

In this part, we will also introduce how to cluster users and how to create User Context, and the **GHUC** architecture.

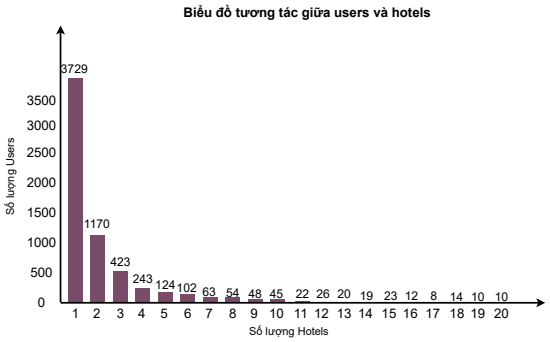

Figure 5: The diagram shows the interaction between User and Hotel

4.1 User Clustering

In *GHRS* each user will be classified into a specific cluster using the *K-means* algorithm based on data characteristics obtained with *Autoencoder*. An important issue when using this algorithm is finding the right number of groups to customize performance. With regular *GHRS*, to choose the number of clusters, we can use the *Elbow* method and *Average Silhouette* algorithm, but our team only uses *Average Silhouette* for this method.

K-means algorithm The K-*K-means* algorithm is a simple clustering algorithm, using measure d and classes K in the data set to calculate the centroids, thereby sorting the classes according to the appropriate centroids. where k are the focal points and K are the data layers

$$d = \sum_{k=1}^{k} \sum_{i=1}^{n} \parallel (x_i - u_k) \parallel^2$$

Average Silhouette Usually, *Average Silhouette* is used to measure the performance of classified groups. The higher the *Average Silhouette* score, the better the grouping.

4.2 User Context

In Figure 7, it can be seen that we use *Word Embedding* tool to convert the Hotel's description to vector space and concurrently multiply the weight which is the Rating that the User has rated for that Hotel. Finally, we divide by the total weight of the Ratings. In this project, we use *PhoBert_base (dim=768)* as the main *Word Embedding* tool.

In this way, we have personalized Users with the purpose of combining with *Similarity Graph Feature* to support more accurate clustering of Users. Thus, improve the performance of the model.

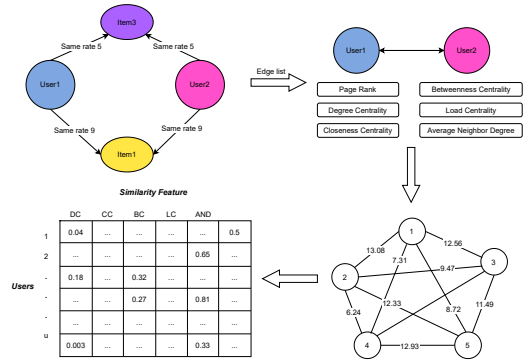

Figure 6: Diagram depicting similarity graph feature

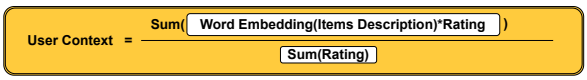

Figure 7: Formula represent User Context

4.3 Architecture

The overall structure of our proposed system (*GHUC - Graph-based Hybrid User Context*) is presented in Figure 8. The system architecture includes six main steps:

1. **First step**, we build *Similarity Graph* with number of users as nodes. Two users will be connected based on their similarities. The edge connects a pair of users that have more than α percent of products with similar ratings.

   A set of information will be extracted from the similarity graph for each user. For example,

we calculate the *PageRank* of the nodes, degree centrality, proximity centrality, betweenness centrality of the shortest path, load centrality, and the average degree of the neighbors of each node in the graph. As a result, this matrix depends on different data processing magnitudes using an interest-based co-operative approach.

2. **Second step**, we combine *User Context* with *Similarity Graph*. Therefore, we have a combination matrix from different feature types, which is then used as input to the *Autoencoder* stage.

3. **Third step**, we apply *Autoencoder* to extract new features and reduce dimensionality. This includes choosing an appropriate optimizer, using a correct loss function and neural network architecture, and preventing overfitting.

4. **In the fourth step**, we use the new features encoded by *Autoencoder* to cluster users, using the K-*K-means* algorithm to create several groups.

5. **The fifth step**, after dividing users into appropriate groups, calculates the rating for Hotels by averaging the Rating among users in a cluster, then updating that rating for people in the same cluster but not yet rating for that Hotel.

6. **Sixth step**, After updating the average rating based on users in the same cluster, there may still exist clusters without users rating all the hotels. To complete the missing ratings, we use the *Content-based filtering* method. This method will find k hotels with the highest similarity to the hotel being processed. Then, take the average rating of these k hotels to update the hotel being processed

5 Experiment and performance evaluation

5.1 Evaluation measures

To be able to evaluate the recommendation system, we can use many methods such as *offline evaluation without caring about the sorting order*, *offline evaluation considering the order arrangement*, *online reviews*, *A/B testing*, etc. Here we only use the measures *Mean Squared Error (MSE)*, *Root Mean Squared Error (RMSE)*, *Mean Absolute Error (MAE)* and *Normalized Mean Absolute Error (NMAE)* to evaluate and compare.

$$MSE = \frac{\sum_{(u,i)}(\hat{r}_{u,i} - r_{u,i})^2}{n}$$

$$RMSE = \sqrt{\frac{\sum_{(u,i)}(\hat{r}_{u,i} - r_{u,i})^2}{n}}$$

$$MAE = \frac{\sum_{(u,i)} |\hat{r}_{u,i} - r_{u,i}|}{n}$$

$$NMAE = \frac{MAE}{r_{max} - r_{min}}$$

5.2 Methods

During the experiment, we use *GLocal-K* and *GHRS* because these two models are leading in solving problems related to data sparsity as well as the cold start problem in recommendation systems. One technique we use for *GHRS* to improve the recommendation system is to combine *Similarity Graph* with *User Demographic*. This will help solve the problem of cold start and sparse rating matrix more effectively.
At the same time, we also introduce another model, the *GHUC (Graph Hybrid User Context)* model to handle the problem of missing User information. This is also a model that we use to develop new methods that will be mentioned in this section. Finally, we combine the above traditional methods together based on *GLocal-K*. Specifically, *Ensemble GlocalK+GHUC* and *Ensemble GlocalK+GHRS*.

5.3 Experimental results

After training and running proposed methods, including *Graph-based Hybrid User Context*,*Ensemble GlocalK+GHUC* and *Ensemble GlocalK+GHRS*, and running traditional methods such as *GlocalK*, *Graph-based Hybrid Recommendation System* and *user-user cosine* collaborative filtering for comparison, our team obtained results according to *MSE*, *RMSE*, *MAE*, *NMAE* measures.

From Table 2, we can see that the *Graph-based User Context* method brings the highest efficiency with results of 2,390 MSE, 1,546 RMSE, 1,322 MAE and 0.220 NMAE; shows that using *User Context* can increase the performance of the GHRS method. At the same time, combining *GLocal-K* with *GHRS* also brings good results with 2,286 MSE, 1,512 RMSE, 1,306 MAE, 0.217 NMAE.

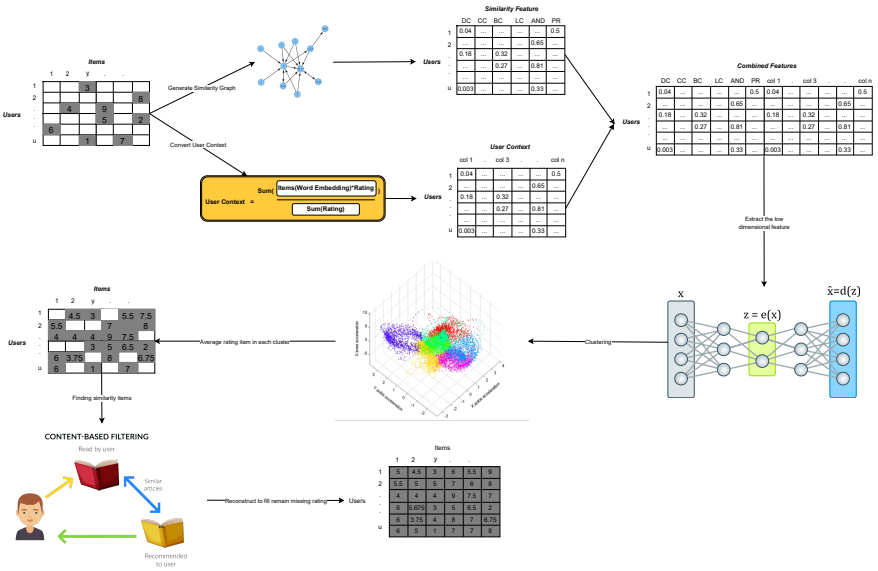

Figure 8: Graph-based Hybrid User Context Architecture

Table 2: The table results are obtained from the experimental process

	MSE	RMSE	MAE	NMAE
User-user cosine	12.66	3.56	2.60	0.43
GlocalK	2.492	1.578	1.357	0.226
GHRS	2.457	1.567	1.328	0.221
GHUC	**2.390**	**1.546**	**1.322**	**0.220**
Ensemble GlocalK(0.4) + GHUC(0.6)	2.287	1.512	1.307	0.217
Ensemble GlocalK(0.4) + GHRS(0.6)	**2.286**	**1.512**	**1.306**	**0.217**

6 Conclusion

In this project, we have developed a new model that can be used in a hotel recommendation system using the *Graph-based Recommendation System* method as the foundation combined with *User Context*, at the same time, we offer a number of other combined methods to improve performance. With the results obtained, we believe that the *Graph-based User Context* method we developed can be used for hotel recommendation systems, and can then be further developed and applied to other recommendation systems. other suggestions in the future. Some of the difficulties we encountered during our research included the fact that the data set was quite sparse, but we were able to come up with the best available solutions to address this obstacle.

In the future, we will focus and further develop *Matrix Completion* and *Context-Aware* to improve the accuracy and performance of the recommendation system.

References

[1] Florian Strub, Romaric Gaudel, and Jeremie Mary. *Hybrid Recommender System based on Autoencoders.* arXiv, 2016

[2] Yang Li, Kangbo Liu, Ranjan Satapathy, Suhang Wang, and Erik Cambria. *Recent Developments in Recommender Systems: A Survey.* arXiv, 2023

[3] Soyeon Caren Han, Taejun Lim, Siqu Long, Bernd Burgstaller, and Josiah Poon. *GLocal-K: Global and Local Kernels for Recommender Systems.* arXiv, 2021

[4] Muhan Zhang and Yixin Chen. *Inductive Matrix Completion Based on Graph Neural Networks.* arXiv, 2019

[5] Zahra Zamanzadeh Darban and Mohammad Hadi Valipour. *GHRS: Graph-based Hybrid Recommendation System with Application to Movie Recommendation.* Elsevier journal, 2021

[6] Shoujin Wang, Liang Hu, Yan Wang, Xiangnan He, Quan Z. Sheng, Mehmet A. Orgun, Longbing Cao, Francesco Ricci, and Philip S. Yu. *Graph Learning based Recommender Systems: A Review.* IJCAI, 2021

[7] M. Nanthini, and K. Pradeep Mohan Kumar. *Resolving cold start and sparse data challenge in recommender systems using multi-level singular value decomposition.* Springer, 2022

[8] Duc The Pham, Thanh Luan Tran, Kiet Nguyen Van, and Tin Huynh Van. *Hotel recommendation system.* UIT, 2022

[9] Xin Dong, Lei Yu, Zhonghuo Wu, Yuxia Sun, Lingfeng Yuan, and Fangxi Zhang. *A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems.* Proceedings of the AAAI Conference on Artificial Intelligence, 2017