RECOMMENDER SYSTEM USING MACHINE LEARNING AND DEEP LEARNING TECHNIQUES

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By:
ANKUR SAXENA
(2K21/CSE/06)

Under the Supervision of
Dr. SANJAY KUMAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

MAY, 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Ankur Saxena, Roll No. 2K21/CSE/06 student of M. Tech (Computer Science and Engineering), hereby declare that the Project Dissertation titled **"Recommender System using Machine Learning and Deep Learning Techniques"** which is being submitted by me to the Department of Computer Science & Engineering, Delhi Technological University, Delhi, in partial fulfillment of requirements for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi
Date:

ANKUR SAXENA
(2K21/CSE/06)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I, hereby certify that the Project titled **"Recommender System using Machine Learning and Deep Learning Techniques"** which is being submitted by Ankur Saxena, Roll No. 2K21/CSE/06, Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date:

Dr. SANJAY KUMAR
SUPERVISOR

ABSTRACT

With the advancements in modern communication technology, ecommerce has gained immense popularity. Today, there are online stores like amazon, flipkart etc. where people can buy stuff from the comfort of their home, entertainment venues like youtube, spotify etc. where they can get entertained and so on. Since, online platforms can host a large volume of items from which people can select, it presents another problem of "spoiled for choices".

To address this issue, recommender systems have emerged as a very potent tool. These systems take into consideration the user's past behaviour as well as the attributes of the various items and after applying some algorithm it generates a candidate set from the complete set of items, it then ranks the items in the candidate sets and then present them to the user according to the ranking of the items.

In this dissertation, we present a comprehensive overview of a recommender system. We discuss the model behind it; the phases it has; algorithms that power these systems whether they are traditional like matrix factorization or modern techniques based on machine learning and deep learning, their benefits and challenges associated with each; and the applications of recommender systems.

We have focussed our attention on session-based recommender systems as the reach of internet is growing more-and-more people have started consuming digital content as well as pursuing ecommerce. A session-based recommender system is a win-win solution for both the consumers and the producers as the consumers get a better purchasing experience and businesses can optimize their decision-making with the analyses of the data generated by recommender system and using that same data to boost the performance of the recommender systems which in-turn enhances consumer experience. We are proposing a novel method for session-based recommender system. We are using graphical neural networks (GNN) for our model.

Our model takes the datasets of user-item interactions, preprocess them, then it learns item embeddings and positional embeddings, learn relevant neighbourhood of each item using item-KNN, creates local graph as well as global graph and gets an embedding of every item in each local as well as global context, then add the local and global embedding of the item to get final representation, then it takes a dot product of the final representation and initial item embedding to get a final score which signifies the importance of the item to the user. Finally, it recommends item based on the final score to the user.

We have used precision (P@K) and mean reciprocal rank (MRR@K) for evaluating the effectiveness of our model. We have used Diginetica, TMall, Nowplaying datasets. Our method performs much better than any non-graphical neural network based recommneder system. Among the models based on GNN, our model performs much better.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I am extremely grateful to my project guide, Dr. Sanjay Kumar, Assistant Professor, Department of Computer Science and Engineering, Delhi Technological University, Delhi for providing invaluable guidance and being a constant source of inspiration throughout my research. I will always be indebted to him for the extensive support and encouragement he provided. I am highly indebted to the panel faculties during all the progress evaluations for their guidance, constant supervision and for motivating me to complete my project work. They helped me throughout by giving new ideas, providing necessary information and pushing me forward to complete the project work.

Place: Delhi
Date:

ANKUR SAXENA
(2K21/CSE/06)

Contents

List of Figures

List of Tables

List of Abbreviations

- **GNN** – Graphical Neural Network
- **GRU** – Gated Recurrent Unit
- **ML** – Machine Learning
- **NLP** – Natural Language Processing
- **RNN** – Recurrent Neural Network
- **RS** – Recommendation System
- **SBRS** – Session-Based Recommendation System

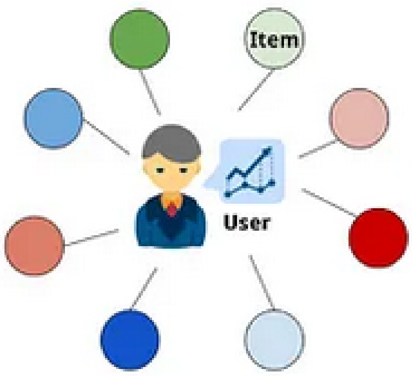
CHAPTER 1

INTRODUCTION

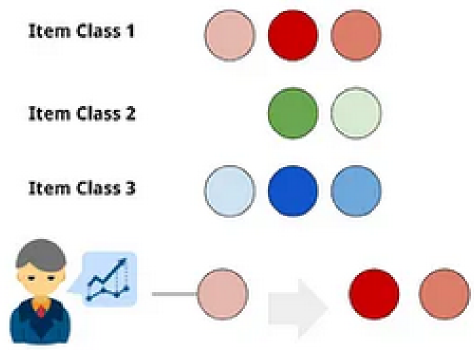
A recommender system is a computer-based algorithmic approach that analyzes user preferences and item characteristics to provide personalized recommendations. It utilizes various techniques, such as collaborative filtering, content-based filtering, and hybrid methods, to generate suggestions for users.

By leveraging user feedback, historical data, and similarity measures, recommender systems aim to predict user preferences accurately. These systems play a crucial role in improving user experience, enhancing customer engagement, and increasing revenue for businesses.

They are widely employed in e-commerce, streaming platforms, social networks, and other domains where personalized recommendations can enhance user satisfaction and drive engagement. With continuous advancements in machine learning and data mining techniques, recommender systems continue to evolve, delivering increasingly accurate and relevant recommendations to users.



(a) Recommender system targetting users as main actors



(b) Recommender system finding community of users for similar items

Figure 1.1: User-centric vs Item-centric Recommender Systems

The two main actors in a recommender system are-

- Users: user is the actor "for whom" the recommendation is made. For eg., users of social media websites like TikTok.
- Items: item is the actor "of which" the recommendation is made. For eg., on platforms like youtube, videos are recommended to users.

The cornerstone of a recommender system is the user-item matrix that captures the relationship among users and items. One such user-item matrix is shown is Table I where there are five users and five items. Items are rated by the users on a scale of 1 to 5. Higher the rating, more are the chances that the user has liked the item. As we can see, there are 25 cells in-total but only 12 cells has a rating value. This means not all the items were rated by every user, some users rated some items with which they had any kind of interaction.

Table I: User-Item Matrix

User/Item	Item 1	Item 2	Item 3	Item 4	Item 5
User 1		2		1	
User 2	5			1	4
User 3	2	4			
User 4				2	3
User 5		2		4	3

The basic task of a recommender system is to predict a value for these empty cells and then recommend those items which are rated higher to the user. For eg., User 4 has rated item 4 and item 5 only, the task of the recommender system is to predict a rating for items 1, 2 and 3 by user 4, and then, recommend user 4 that item for which the predicted rating is higher.

These recommendations are generated by leveraging user-item interactions and similarity measures, improving user satisfaction and engagement in the process.

Table II: Benefits of Recommender Systems

Benefits for Users	**Benefits for Businesses**
• Personalized Recommendations • Time-saving • Discovery of New Items • Enhanced User Experience • Personalized Notifications	• Increased Sales and Revenue • Improved Customer Satisfaction • Enhanced Customer Engagement • Competitive Advantage • Data-Driven Insights

1.1 Recommendation System Phases

A full-fledged recommendation system consists of many stages which are broadly classified into three phases as shown in Fig. 1.2. As we can see, a recommendation system acts as a control system as the output plays an important role in further recommendations through the feedback mechanism.

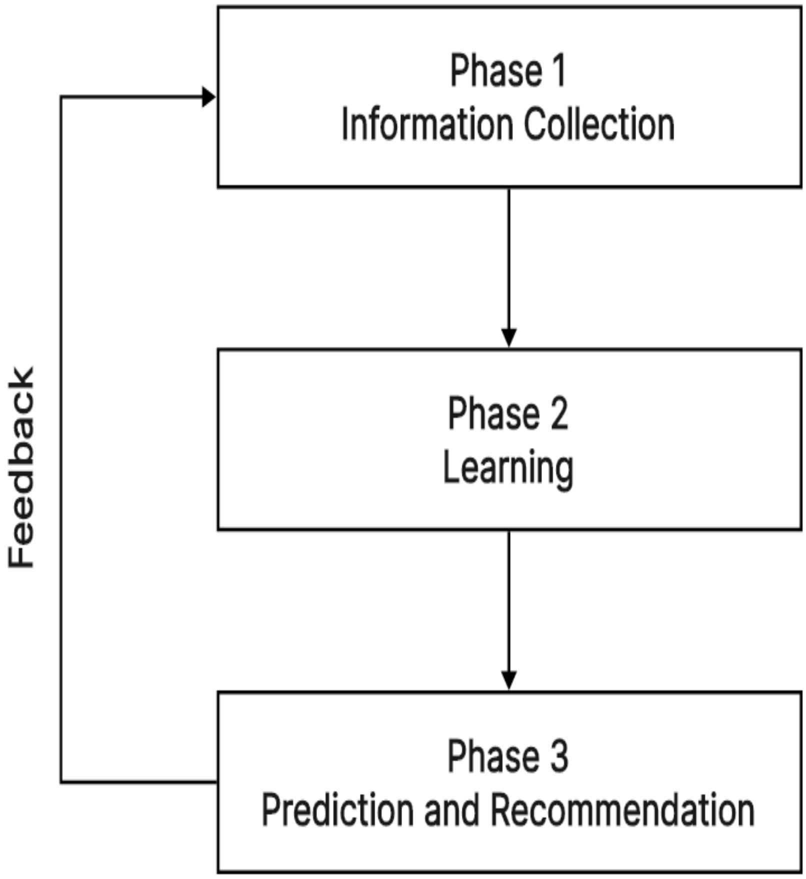


Figure 1.2: Phases of Recommender System

1.1.1 Information Collection phase

The very first phase is to collect relevant and accurate information about the user for creating a user profile including the user's attributes, behaviors, and information of the item a user access. Recommendation systems highly depend on this phase as it the dataset generated determines the accuracy of predictions.

- Explicit feedback requires that the user provides a rating for items. The information gathered is highly reliable. But, the challenge it presents is that it requires the user to take extra pain to provide ratings.
- Implicit feedback refers to finding out the preferences of a user by the way they behave. This is done by monitoring the behavior of the users such as time spent on web pages, links accessed, button clicks, and e-mail content among others. The benefit of implicit feedback is that it does not require any effort from the user but it can lag behind explicit feedback in case of accuracy as some form of assumption is there while inferring the preferences of the user from his behavior. It is argued that implicit feedback is much more objective than explicit feedback as it does not require the user to act in a socially-desirable way which helps in reducing the bias.
- Hybrid feedback takes together the strengths of explicit and implicit feedback mechanisms. Many innovative approaches are being designed by combining these two.

One such technique can be using implicit rating as a check on explicit rating, which enhances accuracy.

1.1.2 Learning phase

In this phase, the recommendation system applies algorithms to filter the dataset consisting of the profiles of the users and train the model.

The model learns the user preferences by looking at the past relationship of the user with different items and classifying users with similar users as well as classifying items that share similar characteristics. In this phase, different communities of users and items come up.

1.1.3 Prediction phase

In this phase, our recommendation system churns out recommendations using the model learned in the learning phase.

Now, we can query our recommender system to find out whether a particular item will be liked by the user or not, and then, the model gives the probability of the likelihood of an item by the user and by using some kind of threshold we can decide whether to recommend this item to the user or not.

1.2 Objective of this Dissertation

The objective of this dissertation is to study the amazing area of recommender systems. We aim to present a comprehensive overview of the recommender systems by focussing our attention on the session based recommender systems. We aim to propose a novel method for session-based recommender systems using the combination of graphical neural networks and item-based KNN.

1.3 Organization of this Dissertation

This dissertation is organized as follows-

Chapter 2 presents an overview of a recommender system. It presents a detailed analyses of algorithms used in recommender systems like content-based filtering, collaborative filtering as well as hybrid mechanisms. Their benefits, challenges, specific use-cases are discussed. Deep-learning based approaches has also been discussed extensively. It also introduces session-based recommender systems. We have provided a detalied study of what a SBRSs is, its components, comparison of sequence data and session data, properties of a session like duration, order etc. and their impact, related work done, applications and challenges faced by SBRSs.

Chapter 3 presents our proposed model for session-based recommender system. We have presented our algorithm in a step-by-step procedure. The two main components of our session graph and global graph are discussed in detail. All the equations used are also mentioned. A flowchart of our proposed model is also present.

Chapter 4 contains the specification of our experiment and its results. The analyses of the performance of our model in comparison to other popular models is presented in this chapter. We provide various ablation studies on our model as well.

CHAPTER 2

LITERATURE REVIEW

2.1 Recommendation system techniques

There are many techniques that are used to make a recommender systems. Each technique has its own benefits and challenges.

2.1.1 Content-based Recommendation Systems

Content-based recommendation systems take advantage of user profiles by extracting characteristics from the contents of things with which the user has previously interacted. Therefore, the similarity is used as the metric in recommendations.

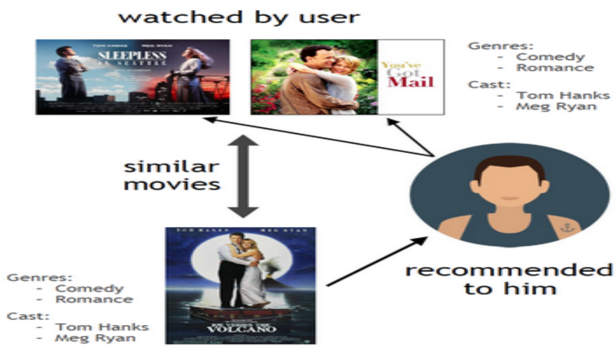


Figure 2.1: Content-based Filtering

Similarity can be calculated by Naïve Bayes classifier, decision trees, cosine-similarity, etc. These systems are individualistic in nature as only the profile of the end user is used to make recommendations. These systems are serial in nature as only the profile of the end-user is utilized for recommendation and not the profile of other similar users.

- Benefits of these systems involve that they can recommend new items even if no user rating is provided earlier. Also, it manage to adjust its recommendations with the change in user's preferences in short span of time.
- Challenges faced by these systems involve content overspecialization where users are given recommendations that are very similar to items that are already present in their profiles, limited content analysis issues as these systems are highly dependent on item's metadata which should have high-quality description, so, if item metadata is not rich enough, recommendations suffer.
- Examples of such systems include Citeseer, LIBRA.

2.1.2 Collaborative-Filtering-based Recommendation Systems

Collaborative filtering is a technique that uses a metric that calculates similarity among the users as well as the items simultaneously as its fundamental algorithm to give recommendations to the end user.

It is parallel in nature as other user choices are also used.

It is domain-independent. The basic idea behind this technique is to build a user-item matrix where each entry represents a rating that was given by the user to the item. The group of users which are similar forms a community.

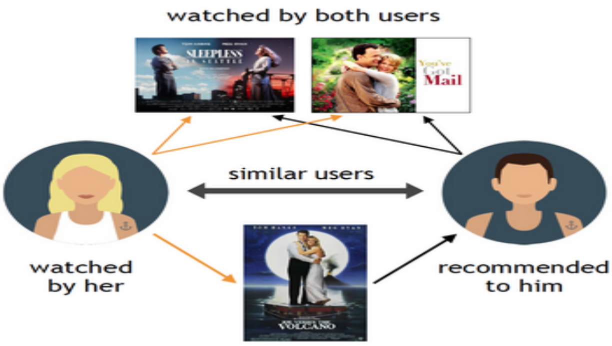


Figure 2.2: Collaborative Filtering

The user gets the recommendation of those products which were not rated by them but were rated by other users in the community. Similarly, items that share features can be classified together.

- Memory-based filtering algorithms generate recommendations using the complete dataset of user and item interactions. They use various similarity measures like cosine-similarity, Pearson correlation coefficient, etc. Memory-based systems are further divided into-
  - User-based filtering: In user-based collaborative filtering, users are grouped together and share similar nature. This is done by calculating a metric that represents a similarity score among users. For any current user, the system will first find other users who have a high similarity score with respect to the current user and then will provide recommendations based on what other similar users are preferring. This kind of recommendation is useful when there is a very large community of users like social-media platforms etc.
  - Item-based filtering In the item-based collaborative filtering technique, items that share similar features are grouped together and if the user has rated anything from the group earlier so other things from the group can also be recommended. This kind of technique is very useful in situations where there is a very large number of items present such as e-commerce stores. It groups items and then recommends groups based on the situation.
- Model-based filtering algorithms generates a novel model from the dataset and uses the model to give further recommendations. Some techniques are-
  - Clustering: Clustering algorithms partition the datasets into groups to identify meaningful communities existing within them[36]. The performance of

a clustering algorithm can be evaluated by looking at the similarities at two levels - intra-cluster similarity which should be as high as possible and inter-cluster similarity which should be as low as possible. Clustering helps by reducing the candidature set in the recommendation systems.

  - Decision trees: The foundation of decision trees are tree graphs, which are built by simulating a collection of training instances for which class labels are known, and then applying them to things that have never been seen before. Some lacking characteristics can be handled via decision trees..
  - Link analyses: A pattern is generated by exploring relations among interconnected objects. It is highly used in web searches.
  - Regression: Regression is a technique that is highly used for modeling of linear systems. It is used the cases where we want to predict the value of a dependent variable whose value is determined by multiple other independent variables. Curve fitting is the most visible face of regression.
  - Association rule: Association rule is a data mining algorithm that extracts patterns, and correlations from the dataset.
  - Bayesian classifier: These are probabilistic frameworks used for classifying purpose and uses conditional probability and Bayes theorem as their fundamental guide[22]. They have the advantage of being resilient to single isolated noise points and handling missing values.

- Benefit of recommenders based on collaborative filtering is that they can perform in areas where items are not highly feature-rich. They can provide serendipitous recommendations such as recommending an item to user M on the basis of interests of the similar user N.

- Collaborative-filtering based systems have many merits but at the same time they face many challenges-

  - Cold start: When a new person or item is added and the recommender system lacks sufficient data to make reliable predictions, it is known as the "cold-start problem" [10]. If a new user is added, the system does not know their preferences because they do not have a user profile. In a similar vein, whenever a new item is introduced, the system lacks sufficient knowledge on its characteristics, such as popularity, user interactions with the item, etc. In other words, when more people or things are added, the matrix grows larger and more sparse.
  - Scalability: As the volume of the dataset increases, the computational efficiency of the recommendation system decreases which leads to unsatisfactory results[49]. One such solution is to reduce the dimensions by using techniques like Singular Value Reduction (SVD).
  - Synonymy: Synonymy is the issue which recommender system faces when there is a large number of items present where many items are having similar features. For a recommender system to distinguish among items that have a high number of similar features is a tough task.

- Examples of such system includes Ringo, GroupLens, amazon eCommerce store, etc.

2.1.3 Hybrid-filtering-based recommendation systems

A new domain of algorithms has emerged by combining the two traditional approaches of content-based filtering and collaborative filtering. The objective is to use suppress the weakness of one model with the use of another model.

- Weighted hybridization takes a combination of filtering techniques and creates an overall score by assigning different weights to each individual technique. One example is P-tango[13].
- Switching hybridization uses different techniques according to the situation. It avoids the problem of one technique by switching to another. One such example is DailyLearner.
- Cascade Hybridization uses a step-by-step refinement procedure. Recommendations from one stage are refined in the next stage. One such example is EntreeC [9].
- Mixed hybridization combines the recommendations at the item level from different recommendation techniques. One such example is PTV [65].
- Feature combination is a technique where features generated by one technique are fed into the second recommendation system as a part of the input. For example, ratings generated by collaborative filtering can be used in content-based filtering systems. One such example is Pipper [5].

Table 2.1: Machine Learning Techniques for Recommender Systems

Technique	**Research Papers**
Collaborative Filtering	[32] [61] [18]
Content-Based Filtering	[50] [46] [21]
Matrix Factorization	[60] [31] [6]
Association Rule Mining	[3] [8] [25]
Factorization Machines	[55] [54]
Probabilistic Graphical Models	[2] [45] [42]

2.2 Deep Learning based Recommender Systems

Deep learning has significantly transformed various domains like cyber-security in the field of bot detection[63], image processing[7], healthcare[47], social-network analyses[4] etc.

Deep Learning based techniques are an effective approach in solving many real-life problems like various problems related to Natural language processing like fake news and rumor detection, multilabel text classification, bot detection, abusive content detection and many others[35, 33, 34, 64, 58, 44, 59].

With the recent advances in the neural networks, a lot of models have come up that utilizes them. Advantages of neural networks in recommender system-

- End-to-end Differentiability: One significant advantage of neural architectures is their end-to-end differentiability.
- Leveraging Intrinsic Structure: Deep neural networks excel at exploiting the intrinsic structure inherent in the input data.
- Composite and Multi-modal Representation Learning: Deep neural networks offer the capability to make a single function to handle multiple neural building blocks.

As there are many models based on deep learning, we are providing a lookup table for the various publications of the model in Table II.

Table II: A lookup table for reviewed publications.

Categories	**Publications**
Multilayer Perceptron	[75] [84] [14]
Autoencoder	[19] [38] [39]
Convolutional Neural Networks	[90] [48] [74]
Recurrent Neural Networks	[79] [69]
Neural Attention	[28] [40]
Adversary Network	[70] [72]
Hybrid Models	[12] [20]

The deep learning based models can be applied in various application domains. Table III lists various publications according to their application domains.

Table III: Deep Learning based models with applications

Data Sources/Tasks	**Publications**
Sequential Information (w/t User ID)	[11] [79] [80]
Sequential Information (Session based w/o User ID)	[66] [68] [69]
Sequential Information (Check-In, POI)	[74] [84]
Text (Hash Tags)	[23] [48]
Text (Review texts)	[90] [89]
Images (Visual features)	[89] [87]
Audio (Music)	[76] [77]
Video (Videos)	[12] [14]
Social Network	[17] [75]
Cross Domain Network	[17] [75]

2.3 Challenges in Deep Learning based Recommender Systems

- Interoperability
  - Lack of interpretability associated with deep learning models.
  - Hidden weights and activations in these networks are non-interpretable, limiting explainability.

  - Interpreting individual neurons remains challenging.

- Data Requirement

  - Deep learning models are known to be data-hungry.
  - Sufficient data is required to support the rich parameterization of deep learning models.

- Extensive Hyperparameter Tuning

  - Extensive hyperparameter tuning is often required in deep learning.
  - Deep learning introduces additional hyperparameters in comparison to machine learning models.

2.4 Session-based Recommendation System

A recommendation system that bases recommendations on a user's current session, or series of activities, on a website or app, is known as a session-based recommendation system.

Session-based recommendation systems pay close attention to the user's current context, in contrast to other recommendation systems that frequently rely on a user's whole history of actions or profile information to produce recommendations. They are thus able to make recommendations that are more timely, relevant, and match the user's current requirements and interests.

Table IV: A comparison between session data and sequence data

Data type	**Boundary**	**Order**	**Time interval**	**Main relations embedded**
Session	Unordered	Multiple	No	Dependencies based on co-occurrence
Session	Ordered	Multiple	Yes	Dependencies based on co-occurence as well as sequential dependencies also exist
Sequential	Single	Yes	Not included	Sequential dependencies

Session-based recommendation systems often examine a user's most recent session data, including their search queries, page views, clicks, and other pertinent data, to produce recommendations. The algorithm then makes recommendations for things or content based on what it predicts the user would be interested in next using this data.

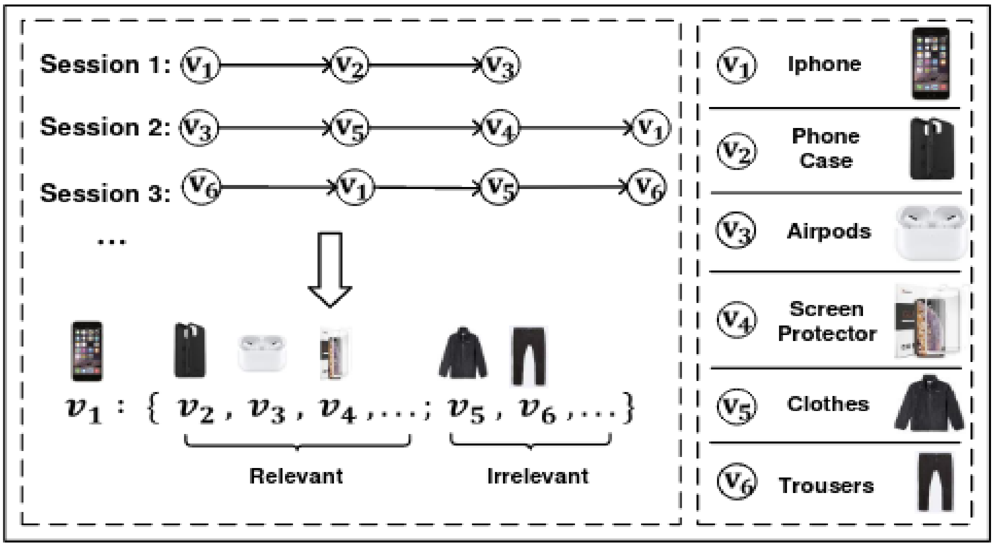


Figure 2.3: Session-Based Recommender System

2.5 Components of an SBRS

2.5.1 User

In a session-based recommendation system, a user is a person who does actions, such as clicking or buying products, and then receives recommendations as a result. Each user has a specific ID and a set of attributes that serve to identify them[73].

Along with the apparent features that are clearly visible, some implicit attributes, such as the user's intentions and moods, may also have a substantial influence on her behaviour.

Let u denote a user and all the users form a set $U = \{u_1, u_2, ...u_n\}$.

2.5.2 Items

Items are the entity that are to be recommended such as a product on a ecommerce store, song on a music app, travel destination etc[73].

Let i denote an item with all its attributes and they are grouped together to form a set of items as $I = \{i_1, i_2, ...i_n\}$.

2.5.3 Action

An action refers to click, view, purchase by the user on an item. The action leads to an interaction of the user with the item.

Let a denotes actions which can be described according to the problem statement and interaction by o which is a tuple represented as $o =< u, i, a >$.

2.5.4 Session

As session is the most important component of a session-based recommendation system, it is important to understand its characteristics to make a better recommendation system.

Table V: SBRS sub-areas

Sub-area	Input	Output	Typical research topic
Next interaction recommendation	Mainly known part of the current session	Next interaction (item)	Next item recommendation, next song/movie recommendation, next POI recommendation, next web page recommendation, next news recommendation, etc.
Next partial-session recommendation	Mainly known part of the current session	Subsequent part of the session	Next items recommendation, session/basket completion
Next session recommendation	Historical sessions	Next session	Next basket recommendation, next bundle recommendation, etc.

- Duration of a session is one of the most important attribute. These can be broadly divided into long, medium and short categories[88].
  - Long duration sessions has more interactions which provides more contextual input data. But, at the same time, not all interactions are relevant which leads to noisy information. Another problem is to embed long-term dependencies.
  - Medium duration sessions are the most commonly found sessions on ecommerce platforms. They are apt in the sense that they usually contains the necessary contextual information as well as less likely to contain too many irrelevant interactions.
  - Short duration sessions have very less contextual information. One of the extreme cases is to recommend the very first interaction of the session.
- Order of the session
  - Ordered session contains has multiple interactions which have a strong sequential dependency existing among them. For example, a user's interaction with a online course platform would be more sequential.
  - Unordered sessions consists of interactions that are more dependent on one-another's co-occurence. These dependencies based on co-occurence are very weak as they are likely ambiguous as compared to sequential dependencies. These are very difficult to learn.
- Impact of user actions
  - Single type action sessions has only one type of actions resulting in only one type of dependency like only clicking, only commenting etc. These kind of dependencies are easy to model.

- Multiple type action sessions include many kinds of interactions giving rise to a complex dependencies. For example a user clicks, comments and purchase in one session.

- Impact of user information
  - Anonymous sessions are the one where the user profile is absent. Any previous interaction of the user is unknown and only the present session interaction only can be used to get some contextual information.
  - A non-anonymous session is the one where user is interacting by disclosing their online identity. Their profiles are present. As and when they interact, their interaction is captured in their profile. It helps to learn about the user as its long-term preferences can be understood. The evolution of their preferences can also be seen. It helps in recommending them better items.

2.6 Related Work

A lot of work is going on in the field of session based recommender systems. SOme of which is listed in Table VI.

Table VI: Techniques for Session-based Recommender Systems

Technique	**Research Papers**
Sequential Models	[26] [91] [29]
Matrix Factorization	[53] [57]
Graph Neural Networks	[81] [24] [41]
Markov Chain	[57] [57]
Convolutional Models	[67] [85] [16]
Self-Supervised Learning	[1] [15] [86]

2.7 Applications

In numerous real-world circumstances and areas, SBRSs are frequently used to the advantage of both customers and enterprises.

- Product recommendation in ecommerce stores by recommending next product to purchase or recommending a basket of items.
- Content recommendation in media and entertainment by recommending next song to listen, next movie to watch, next website to visit etc.
- Service recommendation in tourism by recommending next point-of-interest, next restaurant to visit etc.

Besides these conventional applications, there are many emergent applications like recommending a next trade or investment in the area of finance, recommending next treatment in the area of healthcare etc.

2.8 Challenges faced in SBRS

SBRSs faces following challenges-

- Sparsity: Session data is often limited, making it challenging to accurately capture user preferences.
- Cold-start problem: Recommending items to new users or sessions with limited data is difficult due to the lack of historical information.
- Sequential dependencies: Recommendations need to consider the order of items within a session to capture user intent and provide relevant suggestions.
- Data noise: Session data can contain irrelevant interactions, which may affect the quality of recommendations.
- Scalability: Handling large-scale session data and generating real-time recommendations for numerous users pose scalability challenges.
- Contextual information: Incorporating factors like time, location, and device into recommendations adds complexity.
- Evaluation metrics: Selecting appropriate metrics to measure user satisfaction and account for the sequential nature of recommendations is a challenge.
- Long-tail items: Effectively recommending less popular items with limited session data is challenging.
- Diversity and serendipity: Ensuring varied and novel recommendations to avoid monotony can be challenging in session-based settings.

2.9 Graphical Neural Networks

A deep learning model with the special purpose of working with graph-structured data is referred to as a graphical neural network (GNN), sometimes known as a GNN. It makes use of the connections and interactions that exist naturally inside a graph to carry out a number of operations such node categorization, link prediction, and graph-level predictions. Due to its capacity to extract and make use of the structural information available in graph data, GNNs have attracted a lot of interest recently.

At the core of a GNN is the message passing mechanism, which allows information to be exchanged between connected nodes in a graph [78]. The GNN iteratively aggregates and updates node representations by combining information from neighboring nodes. This process enables the model to capture both local and global dependencies within the graph, enabling effective learning on complex graph structures.

One of the fundamental components of a GNN is the graph convolutional layer [30]. It applies a graph-based operation to transform the representations of nodes, taking into account both the node's own features and the features of its neighbors. This allows the GNN to capture the graph's structural patterns and learn meaningful representations for each node.

GNNs have been effectively used in a variety of fields, including computer vision, molecular chemistry, recommendation systems, and social network analysis. In tasks like node classification, where the objective is to predict the labels or qualities of certain nodes in a network based on their features and connection patterns, they have shown higher performance.

To summarize, a graph neural network (GNN) is a deep learning model tailored for graph-structured data. It utilizes message passing and graph convolutional layers to capture and leverage the structural information present in a graph. GNNs have found applications in diverse domains and have shown promising results in various tasks[82].

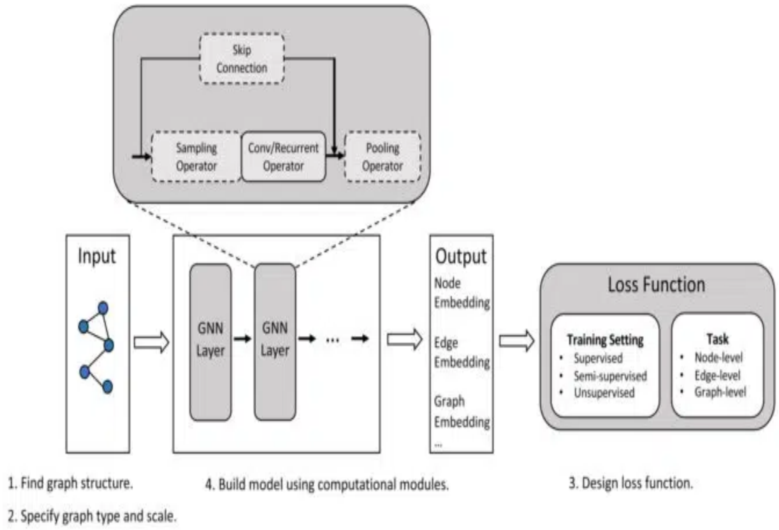


Figure 2.4: Graphical Neural Network

Benefits of graphical neural neyworks in session-based recommender system-

- Incorporating Contextual Information: GNNs can naturally incorporate contextual information present in session data. By considering the connections and interactions among items within a session graph, GNNs can leverage contextual signals such as the order of items, time intervals, and user behaviors. This allows the recommendation system to adapt to dynamic user preferences and provide personalized recommendations based on the current context.

- Handling Cold-Start and Sparsity: Cold-start circumstances, when there is little past data about a user, provide difficulties for session-based recommendation systems. By using the shared item-item connections across several users' sessions, GNNs can alleviate this problem. GNNs may transmit knowledge and provide suggestions even for users with little or no history since they propagate information along the graph.

- Enhanced Recommendation Accuracy: GNNs can capture both local and global patterns in the session graph, enabling them to leverage the collective knowledge from similar sessions and improve recommendation accuracy. By aggregating information from neighboring sessions and considering their influence on the target session, GNNs can provide more accurate and context-aware recommendations.

- Scalability and Efficiency: GNNs can scale to large-scale session graphs efficiently. They leverage efficient message passing mechanisms and parameter sharing techniques, making them suitable for handling large-scale datasets with millions of sessions and items. GNNs can process the graph data in a parallel and distributed manner, enabling faster training and inference times.

CHAPTER 3

METHODOLOGY

In this chapter, we present a detailed step-by-step procedure for our proposed model.

3.1 Algorithm for our proposed model

Let $Q = \{q_1, q_2, \ldots, q_m\}$ be all the items. Any session can be denoted by $S = \{q_{s_1}, q_{s_2}, \ldots, q_{s_l}\}$ consisting of sequential interactions (i.e., items clicked by a user) in an order, where q_{s_i} denotes item q_i clicked within session S, and the length of S is l.

The fundamental task of a session-based recommendation is to recommend the top-N items $(1 \leq N \leq |Q|)$ from Q that are most likely to be acted upon by the user during the current session S.

A snapshot of our proposed model is presented in the Fig.3.2. The model takes unprocessed datasets that contains user-item interactions. The step-by-step procedure followed by us in our model training is presented here.

- Data preprocessing: We perform some analyses on the raw dataset and make some intelligent decision-making to make the datasets relevant for the model.
  - Removing irrelevant sessions like those which have length 1.
  - Removing items which have occurences less than 5.
  - Dividing the dataset into training and testing.
- Learning stage: We find out the relationship among various items and their characteristic features.
  - The adjacency list is a list of lists representing the adjacency information, where each element in the outer list corresponds to an item in the dataset, and the inner list represents the neighboring items of that item. For example, adj[0] would give the list of neighbors for the item with ID 0.
  - The weight list represents the weights associated with the adjacency relationships. Each element in the weight list corresponds to an item in the dataset, and the inner list contains the weights of the neighboring items. The weights indicate the strength or frequency of the connections between items. For example, weight[0] would give the list of weights for the neighbors of the item with ID 0.
  - Item embeddings are dense vectors that capture the features of the item. These vectors are learned during the training process of the model, where the model

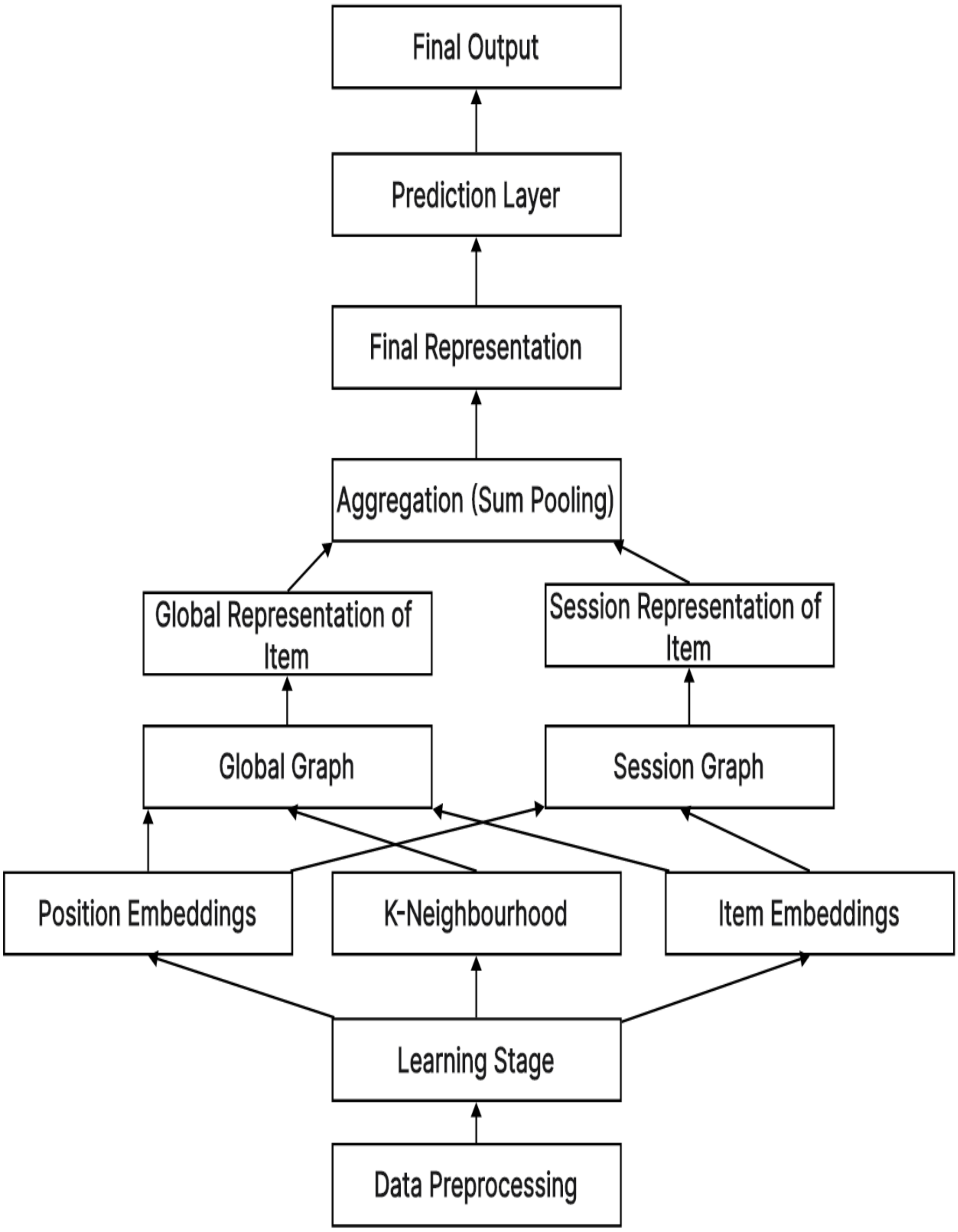


Figure 3.1: Flow chart of the proposed model

tries to find meaningful representations for the items based on the available data.

Every item that is a member of Q is represented by encoding into a unified embedding space at time-step t, i.e., h_{t_i} which is then transformed into a d-dimensional latent vector space.

– Positional embeddings are used to encode the order and position of items within a sequence (e.g., a user's browsing history, a playlist, or a session). It gives the sequential representation of items in a sequence.

- Global graph: It is the mechanism to capture the item-transitions at inter-session level.

  – The idea of the global graph is to take a top-view of all the item-transitions of the dataset. This is achieved by considering all the pairwise item transitions made over various sessions. In our model, we are trying to create a global graph by linking every item-transition that has taken place during any session. All the sessions are considered not only the current one. In this way, we introduce contextual information in our model as item-transitions from previous models are taken into account.

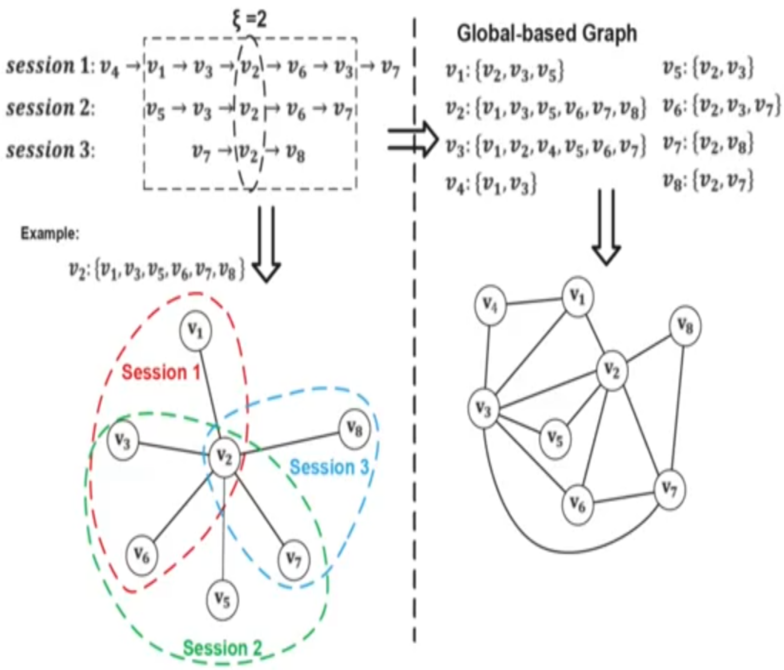


Figure 3.2: Global Graph

  – We find the neighbourhood for each item using item-KNN. It is represented by $N_k(q)$.

  – Let $G_g = (Q_g, E_g)$ be the global graph, where Q_g represents the graph node set containing all items in Q, and $E_g = \{e_{g_{ij}} \,|\, (q_i, q_j) \,|\, q_i \in Q, q_j \in N_k(q_i)\}$ denotes the set of edges. Each edge corresponds to two pairwise items selected from all the sessions.

– We obtain the representation of the item. We apply dropout to remove overfitting.

$$h_{g,(k)q} = \text{dropout}(h_{g,(k)q}) \tag{3.1}$$

- Session graph: It is the mechanism that take note of item transitions at intra-session level.

  – In order to comprehend the session-level embeddings of the items, the session graph aims to describe the specific sequential patterns of the pair-wise neighbouring items of the current session. Using graphical neural networks (GNN), each sequence of the session is converted into a session graph in order to learn the embeddings of the objects in the particular session.

  – Let there be a session $S = \{q_1^S, q_2^S, q_3^S....q_n^S\}$. So, the associated session graph will be represented as $G_S = \{Q_S, E_S\}$. Q_S is a subset of Q representing items on which any action has been taken which can be a click, comment, rating, purchase etc during the session S. E_S is a set of edges of the form e_{ij}^S where each edge represents a session-level transition pattern among two adjacent items $\{q_i^S, q_j^S\}$. An edge is represented by two end-points $e = \{q_i, q_j\}$. We have used four kind of edges by taking the idea [78, 51]

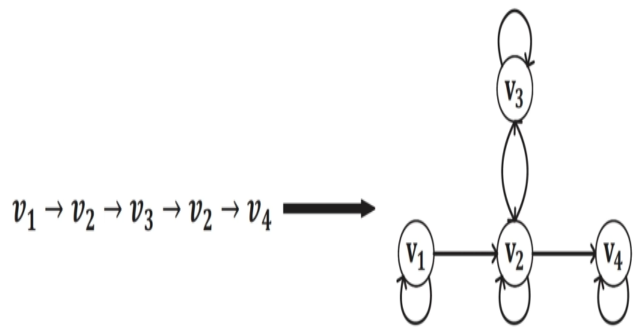


Figure 3.3: Local graph

  * e_{in} refers to the edge from q_i to q_j representing forward transition.
  * e_{out} refers to the edge from q_j to q_i representing backward transition.
  * e_{dual} refers to the edge that has both transition between the two vertices q_i and q_j.
  * q_{self} refers to the edge that both ends are on the same vertex.

  – To learn session level embedding,an attention mechanism is being employed to find the weights among various nodes. Attention score is calculated using the following equation

$$e_{ij} = \text{LeakyReLU}\left(\mathbf{a}^\top \mathbf{r}_{ij}\left(\mathbf{h}_{q_i} \odot \mathbf{h}_{q_j}\right)\right) \tag{3.2}$$

where e_{ij} denotes the significance of node q_j's features to node q_i. Non-linearity is introduced by utilizing LeakyReLU activation function. The relation between q_i and q_j is denoted as $\mathbf{r}_{ij}$, and $\mathbf{a}^* \in \mathbb{R}^d$ represents the weight vectors.

– To normalize the attention weights, we use softmax function.

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top \mathbf{r}_{ij}\left(\mathbf{h}_{q_i} \odot \mathbf{h}_{q_j}\right)\right)\right)}{\sum_{q_k \in \mathcal{N}_s} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top \mathbf{r}_{ik}\left(\mathbf{h}_{q_i} \odot \mathbf{h}_{q_k}\right)\right)\right)} \tag{3.3}$$

Here, α_{ij} represents the normalized attention weight between node q_i and q_j. The LeakyReLU function introduces non-linearity, $\mathbf{r}_{ij}$ denotes the relation between q_i and q_j, $\mathbf{a}^*$ is the weight vector, and $\mathcal{N}_s$ represents the set of neighboring nodes for q_i within the session.

– By taking a linear combination of the features corresponding to the coefficients, the output features for each node are calculated by following equation 3.4.

$$\mathbf{h}_{s,q_i} = \sum_{q_j \in \mathcal{N}_s(q_i)} \alpha_{ij} \mathbf{h}_{q_j} \tag{3.4}$$

Here, $\mathbf{h}_{s,q_i}$ represents the output features for node q_i within the session. The coefficients α_{ij} are computed based on the attention weights, and $\mathbf{h}_{q_j}$ denotes the features of node q_j. The sum is taken over the neighboring nodes q_j within the session $\mathcal{N}_s(q_i)$.

• Final item representation of item in the session

– The final session representation of items can be represented by combining the session-level and global-level information obtained from the session graph and global graph respectively. The session-level and global-level information are combined into a single representation, which reflects both the immediate and broader context of each item.The resulting combined representation incorporates both local details and global patterns, providing a more comprehensive understanding of each item.

– We have used sum-pooling for this task as represented in equation 3.5.

$$h'_q = h_{g,(k)q} + h_s \tag{3.5}$$

– By calculating the average of the session's item representations, the session's information may be derived by following equation 3.6.

$$s' = \frac{1}{l} \sum_{i=1}^{l} h'_{qs_i} \tag{3.6}$$

The corresponding weights are learned through a soft-attention mechanism:

$$\beta_i = t^\top \sigma(M_4 z_i + M_5 s' + b_4) \tag{3.7}$$

In equation 3.7, S represents the session representation, which is obtained by linearly combining the item representations $h'vs_i$ weighted by the corresponding β_i values. The parameters M_4 and M_5 are matrices of size $d \times d$, while t_2 and b_4 are vectors of size d. These parameters can be learned and will contribute to the computation of the weights β_i.

– The final outcome of the session S can be found out by taking a linear combination of the item representations $h'q_{s_i}$ for each item involved in the current session. By combining data from the session graph and the sequential order, it is possible to calculate the contribution that each item contributed by following equation 3.8.

$$S = \sum_{i=1}^{l} \beta_i h'_{q_{s_i}} \tag{3.8}$$

- Recommendation generation: In this step, we take the final representation and item-embedding of each item. We then perform a dot product and gets a final score as shown in equation 3.9. This score represents the similarity or relevance of the item to the user. A higher score indicates a stronger match between the item and the user's preferences.

$$\hat{y}_i = \text{Softmax}(S^\top h_q) \tag{3.9}$$

- Utilising the cross-entropy, the loss function for our model is constructed by following the equation 3.10.

$$L(\hat{y}) = -\sum_{i=1}^{m} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \tag{3.10}$$

CHAPTER 4

EXPERIMENTAL ANALYSES AND RESULTS

In this chapter, we are representing all the specific values for our parameters that we took to carry out our experiments.

4.1 Datasets used

We have used the following datasets to run our model.

- Diginetica dataset has been taken from the CIKM Cup 2016. It consists of typical transaction data.
- TSmall dataset has been taken from IJCAI-15 competetion. It contains session data which were carried out as anonymous users. TSmall is a shopping website.
- Nowplaying dataset has the data of music listened by users.

Table I: Dataset statistics after preprocessing

Dataset	Diginetica	TMall	Nowplaying
Number of clicks	9,82,961	8,18,479	13,67,963
Number of items	43,097	40,728	60,417
Average length	5.12	6.69	7.42

4.2 Data Preprocessing

We have followed [82, 83, 78] for doing the precprocessing work.

- We filtered off the sessions which had only 1 item length.
- We filtered off the items whose appearance count taken as aggregate among all sessions is less than 5.
- For dividing the datasets into training and test data, we have folllowed [43] and have set the sessions that took place in the last week as test data because they were of latest nature and rest of the remaining data as training data.

4.3 Evaluation metrics

For evaluating our model against various other models, we have used the following metrics by following [43, 82, 78].

- P@k:Precision@k measures the accuracy of the top-k recommendations. It evaluates how many of the recommended items at position 1 to k are actually relevant to the user. A greater precision@k value tells that a larger proportion of the recommended items are relevant.

  For instance, if we take precision@5 into account, it computes the proportion of relevant things to the top 5 suggested items. A P@5 grade of 0.8 indicates that 4 of the top 5 suggestions were pertinent to the user.

- MRR@k: Mean Reciprocal Rank at position k is abbreviated as MRR@k. It evaluates the first pertinent item's ranking quality in the suggestions. The first relevant item's location is taken into consideration by MRR@k, which rewards models that rank relevant items higher with better scores.

  Consider MRR@10, which determines the average reciprocal rank of the first pertinent item among the top 10 suggested things. The first relevant item was typically located at position 5 in the recommendations when the MRR@10 score was 0.5. .

- We took k=10, 20 for our evaluation process by following [78], [82].

4.4 Parameter setup

We have used the following parameters for our model by following [43, 82, 37, 78].

- The parameters are initialized by utilizing Gaussian distribution. The standard deviation and mean is 0.1 and 0 respectively.
- Size of latent vector is 100.
- We are using Adam optimizer for which the initial learning rateis taken to be 0.001. It will decay at every third epoch by 0.1.
- The penalty for L2 is set to 10^{-5}.
- Number of neighbors is set to 12.
- Maximum distance between adjacent items is set to 12.

4.5 Item-KNN Parameter

In Item-KNN, "K" represents the parameter that determines the number of closest neighbors to consider when recommending items based on their similarity to the current item in a session. A larger value of "K" allows for a wider exploration of item similarities, leading to improved recommendation results.

We have used K equal to 40 in our model.

4.6 Baseline Algorithms

Table II lists all the algorithms with which we compare our model with.

Table II: Baseline algorithms

Technique	**Research Papers**
Item-KNN[62]	It makes suggestions for items depending on how closely related the items in the current session are to those in previous sessions.
FPMC[56]	The first-order Markov chain is combined with matrix factorization. It takes into consideration sequential effects as well as preferences of the user.
GRU4Rec[27]	Gated Recurrent Units (GRU) are used in this RNN-based model to simulate user sequences.
NARM[37]	It is an extension of GRU4Rec[27] as it introduces attention to RNN for session-based recommender systems.
STAMP[43]	By depending on the user's self-attention on the previous item in the current session, it uses attention layers to capture the user's fleeting interest.
SR-GNN[82]	It obtains item embeddings via a gated GNN layer and computes session-level embeddings using self-attention.
CSRM[71]	It makes use of a memory network to take into consideration the structure of previous n sessions. It helps in making a better prediction of the current session.
FGNN[52]	It uses attention weights of the graphical layers for understanding item embeddings and a feature extractor at the graphical level for making the session recommendations.

4.7 Result

We have used google collaboratory for our experiments. We used 8 vCPUs, 16 GB of RAM, 16 GB of GPU RAM, 128 GB of storage. We ran our model on each dataset for 20 epochs.

- Table III represents the performance of our proposed model in comparison to all the above listed baseline models. The metrics on which this comparison was carried out are P@20 and MRR@20.

- Table IV represents the performance of our proposed model in comparison to all the above listed baseline models. The metrics on which this comparison was carried out are P@10 and MRR@10.

- Fig. 4.1 represents the effect of dropout ratio on P@20 for the TMall dataset.

- Fig. 4.2 represents the trajectory followed by the loss function with respect to the number of epochs.
- Fig. 4.3 represents the trajectory followed by the P@20 with the number of epochs for TMall dataset.
- Table V shows the result in the cases when one of the key components of our model was removed. The two key components are global graph and session graph.
- Table VI presents the results obtained when we used different mechanisms used for the very important aggregation steps. We used gate mechanism, max pooling, concatenation, sum pooling.

Table III: Results for @20

Models/Dataset	Diginetica		TMall		Nowplaying	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
Item-KNN	35.75	11.57	9.15	3.31	15.94	4.91
FPMC	22.14	6.66	16.06	7.32	7.36	2.82
GRU4Rec	30.79	8.22	10.93	5.89	7.92	4.48
NARM	48.32	16.00	23.30	10.70	18.59	6.93
STAMP	46.62	15.13	26.47	13.36	17.66	6.88
CSRM	50.55	16.38	29.46	13.96	18.14	6.42
SR-GNN	51.26	17.78	27.57	13.72	18.87	7.47
FGNN	50.58	16.84	25.24	10.39	18.78	7.15
Our model	**53.97**	**19.02**	**32.04**	**15.05**	**21.30**	**8.47**

Table IV: Results for @10

Models/Dataset	Diginetica		TMall		Nowplaying	
	P@10	MRR@10	P@10	MRR@10	P@10	MRR@10
Item-KNN	25.07	10.77	6.65	3.11	10.96	4.55
FPMC	15.43	6.20	13.10	7.12	5.28	2.68
GRU4Rec	17.93	7.73	9.47	5.78	6.74	4.40
NARM	35.44	15.13	19.17	10.42	13.6	6.62
STAMP	33.98	14.26	22.63	13.12	13.22	6.57
CSRM	36.59	15.41	24.54	13.62	13.20	6.08
SR-GNN	38.42	16.89	23.41	13.45	14.17	7.15
FGNN	37.72	15.95	20.67	10.07	13.89	6.8
Our model	**40.83**	**17.77**	**27.06**	**14.71**	**16.41**	**8.13**

Table V: Performance of our model when key components were removed on a single basis

Models/Dataset	Diginetica		TMall		Nowplaying	
	P@10	MRR@10	P@10	MRR@10	P@10	MRR@10
W/o global graph	**53.11**	**18.77**	**31.44**	**14.54**	**20.33**	**7.43**
W/o session graph	51.78	16.21	31.22	12.67	18.11	6.55

Table VI: Effects of different aggregation operations.

Models/Dataset	Diginetica		Tmall		Nowplaying	
Measures	**P@20**	**MRR@20**	**P@20**	**MRR@20**	**P@20**	**MRR@20**
Gate Mechanism	52.34	18.11	32.80	15.33	**22.47**	7.83
Max Pooling	45.39	16.44	31.87	15.39	19.13	6.71
Concatenation	50.22	17.03	31.55	14.89	19.88	7.93
Sum Pooling	**53.97**	**19.02**	**32.04**	**15.05**	21.30	**8.44**

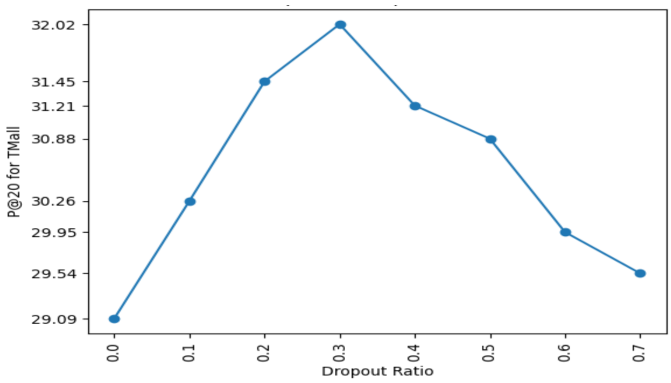


Figure 4.1: Impact of Dropout ratio

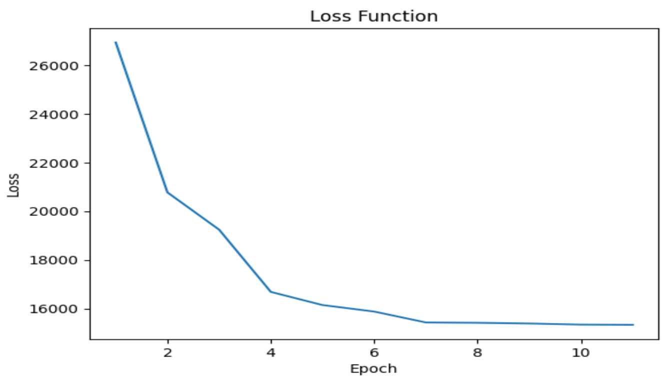


Figure 4.2: Stabilization of loss function with epoch for TMall dataset

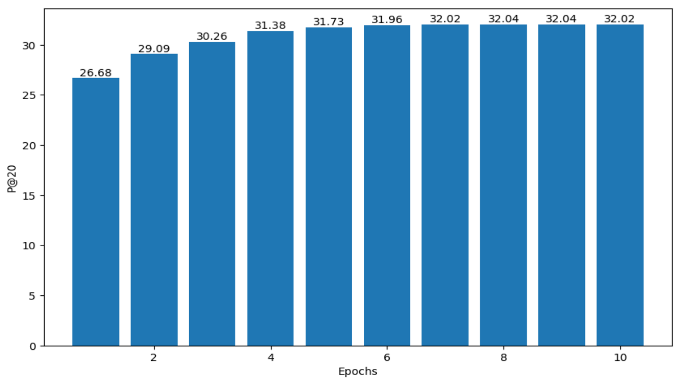


Figure 4.3: Trajectory of P@20 vs epochs for TMall dataset

4.8 Observations

With our experiments, we have observed the following conclusions-

- From the data presented presented in Table III and Table IV, it can be clearly seen that our model has given better results in comparison to models that do not use graphical neural networks. This clearly supports the fact that graphical neural networks are highly successful in the field of session-based recommender systems.

- From the data presented presented in Table III and Table IV, our model even gave superior results in comparison to every graphical neural network based session-based recommender systems that we took for our experiments.

- Our model used to get stabilize by the time it reaches tenth epoch which can be inferred from Fig. 4.1. From the Fig. 4.3, it can be seen that that the maximum performance for P@20 was obtained at ninth epoch for the TMall dataset. Same is the case with other datasets.

- The data presented presented in Table V, it is stating what will be the effect of removing any one of the two key components of our model. It can be inferred that the removal of the global graph does not have any drastic effect but removal of session graphs has a drastic effect on recommendations.

- From the data presented in Table VI, we can clearly infer that sum-pooling mechanism is the best performing for aggregating session and global information of the items. Gate mechanism also performed good and also outperformed sum-pooling in one instance (P@20 for Nowplaying).

- Effect of the dropout rate can be seen from the Fig. 4.1. Dropout is a regularization technique that we used for the global graph representation. The main principle of dropout is to use all neurons for testing while randomly dropping certain neurons with probability p during training. We can see that the model does not work well on both datasets when the dropout ratio is low since it is simple to overfit. When the dropout ratio is in between the extremes, it performs at its best.

  When the dropout ratio starts to hit the higher extreme end, the performance of the model starts to deteriorate since it is a tough ask from the model to learn from data when there are few neurons available.

CONCLUSION

The conclusion of our report focused on studying recommender systems, with a particular emphasis on session-based recommender systems. Throughout our research, we explored various techniques and models employed in this domain and evaluated their performance against baseline models.

One notable finding from our study is that our model based on Graph Neural Networks (GNN) outperformed all the baseline models in terms of recommendation accuracy. The GNN model showcased superior performance in capturing the complex relationships and patterns within session-based data, enabling it to provide more accurate and personalized recommendations.

By leveraging the power of GNNs, our model successfully incorporated the sequential and contextual information present in session data, allowing it to adapt to user preferences and make precise recommendations based on individual sessions. This improved accuracy and personalization offered by our model holds significant promise for enhancing user experience and engagement in real-world recommendation scenarios.

With a rigorous evaluation process and comparison against multiple baseline models, we can confidently assert that our GNN-based model has demonstrated its efficacy and potential in session-based recommender systems.

Overall, our findings contribute to the growing body of knowledge in the field of recommender systems, specifically in the context of session-based recommendations. The success of our GNN model highlights the importance of leveraging advanced techniques to effectively tackle the challenges associated with session-based data.

In conclusion, our research on session-based recommender systems, particularly our GNN-based model, showcases the significant potential for improving recommendation accuracy and personalization. We think that continued investigation and improvement of GNN-based methods will progress recommender systems and improve the user experience across a variety of fields.

REFERENCES

[1] Amr Abdelhamed and Hongning Wang. "S3rec: Self-supervised learning for sequential recommendation". In: *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*. 2020.

[2] Gediminas Adomavicius and YoungOk Kwon. "Improving aggregate recommendation diversity using ranking-based techniques". In: *IEEE Transactions on Knowledge and Data Engineering* 24.5 (2011), pp. 896–911.

[3] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. "Mining association rules between sets of items in large databases". In: *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*. 1993, pp. 207–216.

[4] Sameer Anand et al. "Integrating node centralities, similarity measures, and machine learning classifiers for link prediction". In: *Multimedia tools and applications* 81.27 (2022), pp. 38593–38621.

[5] Chumki Basu. "Recommendation as Classification: Using Social and Content-Based Information in Recommendation Chumki Basu". In: (1998).

[6] Robert M Bell and Yehuda Koren. "Scalable collaborative filtering with jointly derived neighborhood interpolation weights". In: *Seventh IEEE international conference on data mining (ICDM 2007)*. IEEE. 2007, pp. 43–52.

[7] Neeraj Bhat, Navneet Saggu, Sanjay Kumar, et al. "Generating visible spectrum images from thermal infrared using conditional generative adversarial networks". In: *2020 5th International Conference on Communication and Electronics Systems (ICCES)*. IEEE. 2020, pp. 1390–1394.

[8] Sergey Brin, Rajeev Motwani, and Craig Silverstein. "Beyond market baskets: Generalizing association rules to correlations". In: *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*. 1997, pp. 265–276.

[9] Robin Burke. "Hybrid recommender systems: Survey and experiments". In: *User modeling and user-adapted interaction* 12 (2002), pp. 331–370.

[10] Robin Burke. "Hybrid web recommender systems". In: *The adaptive web: methods and strategies of web personalization* (2007), pp. 377–408.

[11] Shi-Yong Chen et al. "Stabilizing reinforcement learning in dynamic environment with application to online recommendation". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 1187–1196.

[12] Xu Chen et al. "Personalized key frame recommendation". In: *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*. 2017, pp. 315–324.

[13] Mark Claypool et al. "Combing content-based and collaborative filters in an online newspaper". In: *Proc. of Workshop on Recommender Systems-Implementation and Evaluation*. 1999.

[14] Paul Covington, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations". In: *Proceedings of the 10th ACM conference on recommender systems*. 2016, pp. 191–198.

[15] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. "Are we really making much progress? a worrying analysis of recent neural recommendation approaches". In: *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys)*. 2019.

[16] Mostafa Dehghani et al. "Neural personalized ranking for item cold-start recommendation". In: *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 2018.

[17] Shuiguang Deng et al. "On deep learning for trust-aware recommendations in social networks". In: *IEEE transactions on neural networks and learning systems* 28.5 (2016), pp. 1164–1177.

[18] C Desrosiers and G Karypis. "Methods, A comprehensive survey of neighborhood-based recommendation". In: (2011).

[19] Xin Dong et al. "A hybrid collaborative filtering model with deep structure for recommender systems". In: *Proceedings of the AAAI Conference on artificial intelligence*. Vol. 31. 1. 2017.

[20] Travis Ebesu and Yi Fang. "Neural citation network for context-aware citation recommendation". In: *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*. 2017, pp. 1093–1096.

[21] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. "Collaborative filtering recommender systems". In: *Foundations and Trends® in Human–Computer Interaction* 4.2 (2011), pp. 81–173.

[22] Nir Friedman, Dan Geiger, and Moises Goldszmidt. "Bayesian network classifiers". In: *Machine learning* 29 (1997), pp. 131–163.

[23] Yuyun Gong and Qi Zhang. "Hashtag recommendation using attention-based convolutional neural network." In: *IJCAI*. 2016, pp. 2782–2788.

[24] Will Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*. 2017.

[25] Jiawei Han, Jian Pei, and Yiwen Yin. "Mining frequent patterns without candidate generation". In: *ACM sigmod record* 29.2 (2000), pp. 1–12.

[26] Balázs Hidasi et al. "Session-based recommendations with recurrent neural networks". In: *Proceedings of the 4th International Conference on Learning Representations (ICLR)*. 2015.

[27] Balázs Hidasi et al. "Session-based recommendations with recurrent neural networks". In: *arXiv preprint arXiv:1511.06939* (2015).

[28] Yogesh Jhamb, Travis Ebesu, and Yi Fang. "Attentive contextual denoising autoencoder for recommendation". In: *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*. 2018, pp. 27–34.

[29] Wang-Cheng Kang and Julian McAuley. "Self-attentive sequential recommendation". In: *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys)*. 2018.

[30] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[31] Yehuda Koren. "Factorization meets the neighborhood: a multifaceted collaborative filtering model". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008, pp. 426–434.

[32] Yehuda Koren, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems". In: *Computer* 42.8 (2009), pp. 30–37.

[33] Akshi Kumar, Nipun Aggarwal, and Sanjay Kumar. "SIRA: a model for propagation and rumor control with epidemic spreading and immunization for healthcare 5.0". In: *Soft Computing* 27.7 (2023), pp. 4307–4320.

[34] Sanjay Kumar et al. "Movie genre classification using binary relevance, label powerset, and machine learning classifiers". In: *Multimedia Tools and Applications* 82.1 (2023), pp. 945–968.

[35] Sanjay Kumar et al. "OptNet-Fake: Fake News Detection in Socio-Cyber Platforms Using Grasshopper Optimization and Deep Neural Network". In: *IEEE Transactions on Computational Social Systems* (2023).

[36] Urszula Kużelewska. "Advantages of information granulation in clustering algorithms". In: *Agents and Artificial Intelligence: Third International Conference, ICAART 2011, Rome, Italy, January, 28-30, 2011. Revised Selected Papers 3*. Springer. 2013, pp. 131–145.

[37] Jing Li et al. "Neural attentive session-based recommendation". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017, pp. 1419–1428.

[38] Sheng Li, Jaya Kawale, and Yun Fu. "Deep collaborative filtering via marginalized denoising auto-encoder". In: *Proceedings of the 24th ACM international on conference on information and knowledge management*. 2015, pp. 811–820.

[39] Xiaopeng Li and James She. "Collaborative variational autoencoder for recommender systems". In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, pp. 305–314.

[40] Yang Li et al. "Hashtag recommendation with topical attention-based LSTM". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2016, pp. 3019–3029.

[41] Yujia Li et al. "Gated graph sequence neural networks". In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. 2015.

[42] Defu Lian et al. "GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 831–840.

[43] Qiao Liu et al. "STAMP: short-term attention/memory priority model for session-based recommendation". In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018, pp. 1831–1839.

[44] Abhishek Mallik, Anavi Khetarpal, and Sanjay Kumar. "ConRec: malware classification using convolutional recurrence". In: *Journal of Computer Virology and Hacking Techniques* 18.4 (2022), pp. 297–313.

[45] Benjamin M Marlin. “Modeling user rating profiles for collaborative filtering”. In: *Advances in neural information processing systems* 16 (2003).

[46] Prem Melville and Vikas Sindhwani. “Recommender systems.” In: *Encyclopedia of machine learning* 1 (2010), pp. 829–838.

[47] Aditya Miglani et al. “A Literature Review on Brain Tumor Detection and Segmentation”. In: *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE. 2021, pp. 1513–1519.

[48] Hanh TH Nguyen et al. “Personalized deep learning for tag recommendation”. In: *Advances in Knowledge Discovery and Data Mining: 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, May 23-26, 2017, Proceedings, Part I 21*. Springer. 2017, pp. 186–197.

[49] Deuk Hee Park et al. “A literature review and classification of recommender systems research”. In: *Expert systems with applications* 39.11 (2012), pp. 10059–10072.

[50] Michael J Pazzani and Daniel Billsus. “Content-based recommendation systems”. In: *The adaptive web: methods and strategies of web personalization* (2007), pp. 325–341.

[51] Ruihong Qiu et al. “Rethinking the item order in session-based recommendation with graph neural networks”. In: *Proceedings of the 28th ACM international conference on information and knowledge management*. 2019, pp. 579–588.

[52] Ruihong Qiu et al. “Rethinking the item order in session-based recommendation with graph neural networks”. In: *Proceedings of the 28th ACM international conference on information and knowledge management*. 2019, pp. 579–588.

[53] Massimo Quadrana et al. “Personalizing session-based recommendations with hierarchical recurrent neural networks”. In: *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys)*. 2017.

[54] Steffen Rendle. “Factorization machines”. In: *2010 IEEE International conference on data mining*. IEEE. 2010, pp. 995–1000.

[55] Steffen Rendle. “Factorization machines with libfm”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 3.3 (2012), pp. 1–22.

[56] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. “Factorizing personalized markov chains for next-basket recommendation”. In: *Proceedings of the 19th international conference on World wide web*. 2010, pp. 811–820.

[57] Steffen Rendle et al. “Factorizing personalized markov chains for next-basket recommendation”. In: *Proceedings of the 19th International Conference on World Wide Web (WWW)*. 2010.

[58] Yash Saini et al. “Abusive text examination using Latent Dirichlet allocation, self organizing maps and k means clustering”. In: *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE. 2020, pp. 1233–1238.

[59] Yash Saini et al. “Abusive text examination using Latent Dirichlet allocation, self organizing maps and k means clustering”. In: *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE. 2020, pp. 1233–1238.

[60] Ruslan Salakhutdinov and Andriy Mnih. "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 880–887.

[61] Badrul Sarwar et al. "Item-based collaborative filtering recommendation algorithms". In: *Proceedings of the 10th international conference on World Wide Web*. 2001, pp. 285–295.

[62] Badrul Sarwar et al. "Item-based collaborative filtering recommendation algorithms". In: *Proceedings of the 10th international conference on World Wide Web*. 2001, pp. 285–295.

[63] Sandeep Singh Sengar et al. "Bot detection in social networks based on multilayered deep learning approach". In: *Sensors & Transducers* 244.5 (2020), pp. 37–43.

[64] Sandeep Singh Sengar et al. "Bot detection in social networks based on multilayered deep learning approach". In: *Sensors & Transducers* 244.5 (2020), pp. 37–43.

[65] Barry Smyth and Paul Cotter. "A personalised TV listings service for the digital TV age". In: *Knowledge-Based Systems* 13.2-3 (2000), pp. 53–59.

[66] Yong Kiam Tan, Xinxing Xu, and Yong Liu. "Improved recurrent neural networks for session-based recommendations". In: *Proceedings of the 1st workshop on deep learning for recommender systems*. 2016, pp. 17–22.

[67] Jiaxi Tang et al. "Personalized top-n sequential recommendation via convolutional sequence embedding". In: *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys)*. 2018.

[68] Trinh Xuan Tuan and Tu Minh Phuong. "3D convolutional networks for session-based recommendation with content features". In: *Proceedings of the eleventh ACM conference on recommender systems*. 2017, pp. 138–146.

[69] Bartłomiej Twardowski. "Modelling contextual information in session-aware recommender systems with neural networks". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. 2016, pp. 273–276.

[70] Jun Wang et al. "Irgan: A minimax game for unifying generative and discriminative information retrieval models". In: *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 2017, pp. 515–524.

[71] Meirui Wang et al. "A collaborative session-based recommendation approach with parallel memory modules". In: *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 2019, pp. 345–354.

[72] Qinyong Wang et al. "Neural memory streaming recommender networks with adversarial training". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 2467–2475.

[73] Shoujin Wang et al. *A Survey on Session-based Recommender Systems*. 2021. arXiv: `1902.04864 [cs.IR]`.

[74] Suhang Wang et al. “What your images reveal: Exploiting visual contents for point-of-interest recommendation”. In: *Proceedings of the 26th international conference on world wide web*. 2017, pp. 391–400.

[75] Xiang Wang et al. “Item silk road: Recommending items from information domains to social users”. In: *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 2017, pp. 185–194.

[76] Xinxi Wang and Ye Wang. “Improving content-based and hybrid music recommendation using deep learning”. In: *Proceedings of the 22nd ACM international conference on Multimedia*. 2014, pp. 627–636.

[77] Xinxi Wang et al. “Exploration in interactive personalized music recommendation: a reinforcement learning approach”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 11.1 (2014), pp. 1–22.

[78] Ziyang Wang et al. “Global Context Enhanced Graph Neural Networks for Session-Based Recommendation”. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’20. Virtual Event, China: Association for Computing Machinery, 2020, pp. 169–178. ISBN: 9781450380164. DOI: 10.1145/3397271.3401142. URL: https://doi.org/10.1145/3397271.3401142.

[79] Caihua Wu et al. “Recurrent neural network based recommendation for time heterogeneous feedback”. In: *Knowledge-Based Systems* 109 (2016), pp. 90–103.

[80] Chao-Yuan Wu et al. “Recurrent recommender networks”. In: *Proceedings of the tenth ACM international conference on web search and data mining*. 2017, pp. 495–503.

[81] Shu Wu et al. “Session-based recommendation with graph neural networks”. In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*. 2019.

[82] Shu Wu et al. “Session-based recommendation with graph neural networks”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 346–353.

[83] Chengfeng Xu et al. “Graph Contextualized Self-Attention Network for Session-based Recommendation.” In: *IJCAI*. Vol. 19. 2019, pp. 3940–3946.

[84] Carl Yang et al. “Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation”. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, pp. 1245–1254.

[85] Fajie Yuan et al. “Simple and effective next item recommendation algorithms for sequential data”. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 2019.

[86] Chen Zhang et al. “SSUM: Self-Supervised User Embedding for Session-based Recommendation”. In: *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*. 2021.

[87] Fuzheng Zhang et al. “Collaborative knowledge base embedding for recommender systems”. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 353–362.

[88] Shuai Zhang et al. "Deep learning based recommender system: A survey and new perspectives". In: *ACM computing surveys (CSUR)* 52.1 (2019), pp. 1–38.

[89] Yongfeng Zhang et al. "Joint representation learning for top-n recommendation with heterogeneous information sources". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017, pp. 1449–1458.

[90] Lei Zheng, Vahid Noroozi, and Philip S Yu. "Joint deep modeling of users and items using reviews for recommendation". In: *Proceedings of the tenth ACM international conference on web search and data mining*. 2017, pp. 425–434.

[91] Wentao Zhu et al. "Next item recommendation with self-attention". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*. 2018.