Chapter 2

Background

"*To understand the limits and opportunities of algorithms in the context of artistic creation, we need to understand that the latter usually consists of three elements: discovery, production, and recommendation.*"

Evgeny Morozov

This chapter presents an overview of the state-of-the-art music recommender system research and various aspects of such systems. We mainly discuss music recommendation systems from two different perspectives; first: we discuss state-of-the-art music recommendation systems, where the model's primary goal is to generate only relevant recommendations for the user. Second, we discuss the state-of-the-art model for diversifying the recommendation system. We cover some prominent literature on music recommendation systems and diversification in the recommendation system related to our work. We also discuss some major evaluation metrics, some state-of-the-art methodologies, and the public datasets used in our work.

2.1 Literature Review in Accurate Music Recommendation

We provide here the necessary background for effective and diverse music recommender systems according to the taxonomy, as discussed in Section 1.2 and represented in the

Figure 2.1.

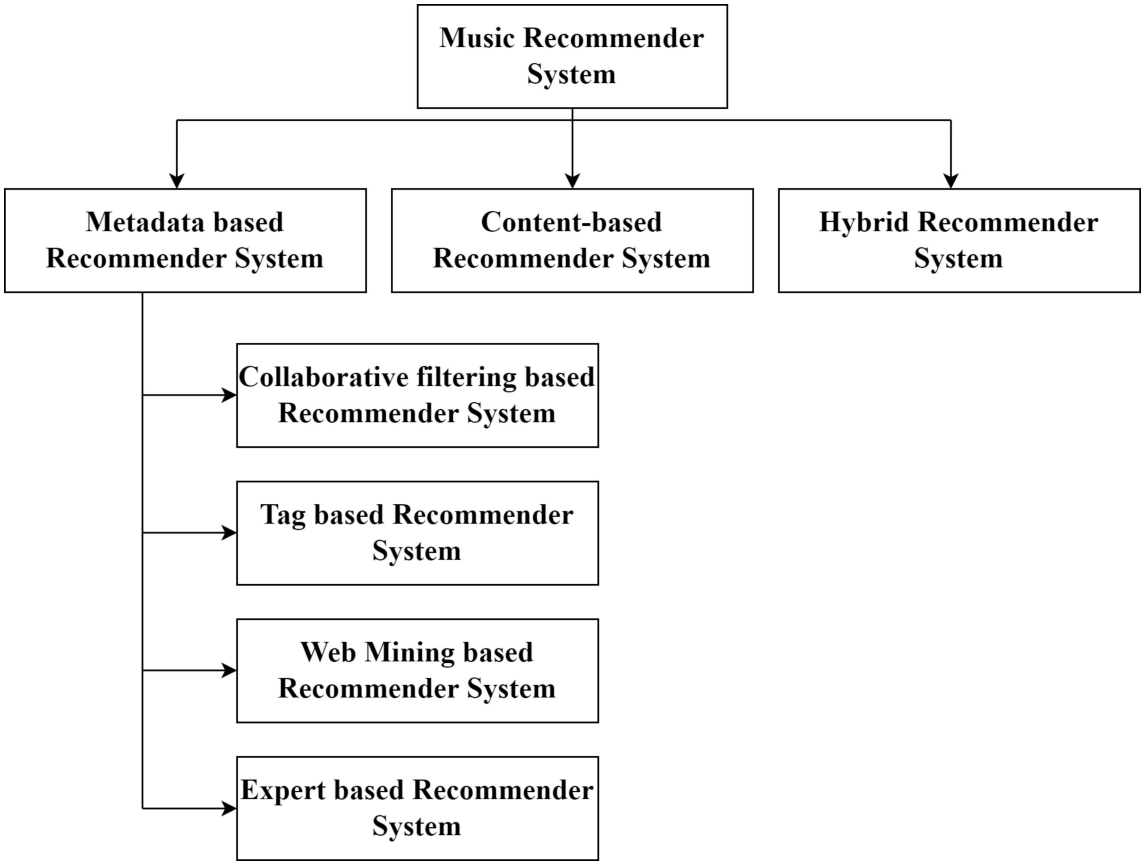

Figure 2.1: Classification of Music Recommendation System

2.1.1 Collaborative Music Recommender System

Recommender systems are programs that attempt to recommend the most relevant set of items to a particular user based on their past preferences. In today's era, recommendation systems are used in various e-commerce applications, like online shopping sites, and also and movie and music applications. The overall goal of these systems is to recommend items or movies to users based on their interests [45]. One major goal of a recommendation system is to overcome the information overload problem, where selecting relevant items from the giant stacks of items is difficult. A recommender system is a personalized assistance application where exploration and discovery of the items from the extensive collection of data become simpler and easier. It also increases

the user satisfaction and revenue of many e-commerce and media streaming platforms such as Amazon, Netflix, YouTube, or Spotify, and social networks such as Facebook or Twitter [4]. Collaborative filtering (CF) has been one of the most common algorithms in recommendation system development. In the collaborative filtering technique, the recommendation is generated based on the assumption that users who share common interests will receive common recommendations [46]. The CF-based techniques are further divided into two parts:

1. Neighbourhood-based models
2. Latent-factor based models

2.1.1.1 Neighbourhood-based models

In the user-based CF approach, the recommendation of items is generated based on the target user's similarity with other users [14]. In the item-based CF approach, item recommendations are generated based on the similarity of the items that target users have preferred in their past interactions with the system [47]. Many CF-based algorithms have been proposed based on machine learning, deep learning, and reinforcement learning methodologies. In CF-based algorithms, the model uses the rating history of users for recommendation generation. The rating information is in the form of a user-item matrix.

shown in the table 2.1.

Table 2.1: An illustration of User-item rating matrix

	I_1	I_2	I_3	I_4	I_5
U_1		2	3		5
U_2	2	3			4
U_3		4	2	2	
U_4	5		3		1

In the table 2.1, ratings are stored in a $m \times n$ matrix, where m is the total number of users present in the system, and n is the total number of items present in the dataset.

The rows of the matrix store rating information for each user, and the columns store rating information for items. The entry in the matrix shows the user's rating for an item i. If a user has rated a particular item, then the corresponding entry will contain rating information. This user-item matrix is considered an input for CF-based models. User-based CF approaches calculate rating predictions for a target user by using two sets of data: the ratings of the target user and the ratings of other like-minded users, i.e., users with similar patterns of ratings. There are many different algorithms for rating prediction in the user-based CF approach. A popular method of rating prediction is described by the following equation:

$$\hat{r_{ui}} = \bar{r_u} + \frac{\sum_{v \in N_i(u)} (r_{ui} - \bar{r}_v)}{\sum_{v \in N_i(u)} |Sim(u, v)|} \tag{2.1}$$

Here $\bar{r}_u$ denotes the user u's average rating value. $N_i(u)$ is number of users similar to the target user u who also rated the item i, and $Sim(u, v)$ is the similarity score between user u and user v. Here in the equation, the similarity between users is calculated using different methods like Pearson co-relation coefficient, cosine similarity, etc. The formula for the Pearson correlation coefficient is described by equation 2.2:

$$Sim(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{ui}} (r_{uv} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \tag{2.2}$$

In equation 2.2 I_{uv} are the sets of items rated by the user u.

2.1.1.2 Latent-factor based models

A latent factor model of CF-based methods differs from neighbourhood methods in rating prediction. The most popular way of rating prediction in latent factor models is matrix factorization [48]. In matrix factorization (CF) methods, we learn a vector of low-dimensional latent features or factors for each user and item. The representation of each user and item's latent factor shows how well a user and item possess particular

latent aspects. The latent factor representations can represent some structural and semantic information. The rating prediction in matrix factorization is made by splitting the original rating matrix (R) into two separate matrices S and M, in such a way that their product will approximate the original rating matrix:

$$R \approx SM^T \tag{2.3}$$

In equation 2.3 $S = |U| \times d$ matrix M is $M = |I| \times d$ and d is the number of factors (dimensions), and it is the parameter that must be optimized. Koren et al. proposed a method for matrix factorization technique by merging latent factors and neighbourhood information of users. The accuracy of the model is improved by using both explicit and implicit information of user and item [49]. Liang et al. proposed a method for music recommendation systems using the matrix factorization technique. They offered a methodology called CoFactor which jointly decomposes the user-item interaction matrix and the item-item co-occurrence matrix with shared item latent factors [50]. Vinagre et al. proposed a method for music recommendation by considering only positive feedback information in the matrix factorization methods. They also suggested some evaluation protocols for the online streaming recommender system [51].

2.1.2 Content-based Recommendation System

A cold start problem exists in the CF-based approach, where recommendation generation for the new user and item is complex. So content-based recommendation systems come into existence where the similarity between a user and an item is calculated using the user's and item's content feature information. That is how we overcome the issue of cold-start recommendations. In the CBF-based approach, the first step is to extract meaningful feature information from users and items, also called user and item profiling. After that, user and item content information are further used to represent comprehensive object representations, as shown in Figure 2.2.

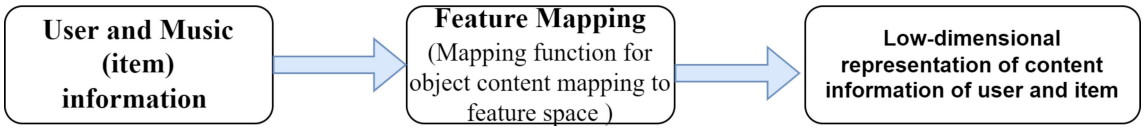

Figure 2.2: Object representation for content-based systems

In such music recommendation systems, content information is extracted first, in which user and item content information can be as described in table 2.2, 2.3, respectively.

Table 2.2: User content description for content-based music recommendation system

Data type	**Example**
Demographic	Age, marital status, gender etc.
Geographic	Location, city, country etc.
Psycho-graphic	Stable: interests, lifestyle, personality, etc. Dynamic: Mood, attitude, opinion

Table 2.3: Item content description for content-based music recommendation system

Data type	**Example**
Editorial metadata	Obtained from experts Cover name, title, composer, Genre etc.
Cultural metadata	Obtained from analysis of corpora Textual information
Acoustic metadata	Obtained from audio data beat, tempo, pitch, instrument, mood etc.

Many music recommendation applications such as Last.FM [1], Allmusic [2], Pandora [3], Spotify [4], and Apple [5] have aggregated millions of users and developed recommendation systems for the common user. Various algorithms have been proposed for music recommendation systems using content information of music to mitigate the cold start problem of recommendation and also for accuracy improvement in recommendation

[1]https://www.last.fm/
[2]https://www.allmusic.com/
[3]http://www.pandora.com
[4]https://www.spotify.com/in-en/
[5]https://music.apple.com/us/browse

systems. We review some relevant work in this area of research. Content information can consist of any information representing music items. In such a system, we use information about the music's content that can be extracted from audio, video, text (lyrics), and musical score information, such as that contained in a MIDI file. Other content information in music is used as implicit information like the artist's name, release date, and any additional information, which manual annotations can provide by music critics or experts [52].

Music psychology considerations show that music selection preferences are affected by factors like environment, current mood, and previous user activity, apart from their inclinations. These factors define users' short-term interest, which changes dynamically over a more extended period. Bogdanov et al. [53] proposed a model based on editorial metadata containing artists' information, release date, and record labels. They present a content-based recommendation model based on low-level music features like timbral, temporal, and tonal information and then derive high-level semantic descriptions of music. Zhu et al. [54] proposed an integrated music recommendation system based on functions of music genre classification, emotion classification, and similarity between music queries. Many machine learning methods are applied for content-based recommendation systems and they have achieved promising results using content information of music, social tags, manual annotations, and audio content information.

Moving to the use of deep neural networks in recommendation systems, Van den Oord et al. [37] introduced a deep content-based recommendation system using audio signals in terms of MFCC (Mel-frequency cepstral coefficients) values. This research's primary objective was to resolve the cold start problem of collaborative filtering model. The authors introduced a latent factor model for recommendation generation using music audio. They apply deep convolutional neural networks to predict latent factors from music audio. In the same direction, Wang et al. [55] scrutinized a two-stage process of content-based recommendation by suggesting that MFCC extraction does not

capture all the relevant music information at stage one. The authors proposed a model based on a deep belief network and a probabilistic graphical model. They consolidated this model into an automated process that simultaneously learns audio content features and generates personalized recommendations. Cano et al. [56] also used a metadata-free system named MusicSurfer. Instead of using MFCC, they automatically extract instrumentation, rhythm, and harmony information from music audio signals and generate recommendations using different similarity metrics.

In the past few years, Deep Learning has witnessed tremendous success in recommendation systems, as it has in other areas like computer vision, NLP, etc. The primary focus of a content-based recommendation system is to overcome the cold start problem of collaborative filtering. Lee et al. [57] applied a Deep Learning approach to overcome these problems, and then recommendation generation problems were formulated as a video content similarity learning problem. They use the visual and audio content to learn video embedding and use them to predict relationships between co-watch systems.

Zhong et al. [58] also used a CNN (Convolutional Neural Network) model for a content-based music recommendation system. They converted every music segment into spectrogram images using Fourier transformation and used this as an input for CNN. Barkan et al. [59] proposed a cold-start recommendation model that uses content and collaborative filtering. The proposed algorithm is used for learning item similarity relations utilizing the content of item information. The model uses CNN on top of the Word2Vec [60] embedding representation of an item that uses Bayesian personalized ranking for the item vector.

Several researchers have used Recurrent Neural Network (RNN) and its variants like Long Short Term Memory (LSTM) for recommendation generation [61–63]. RNN-based models use the sequential nature of the song with some explicit information like tags and sessions. Musto et al. [63] proposed a model based on a bidirectional RNN to learn the representation of the items to be recommended based on their textual description

and extracted content data using Linked Open Data.

2.1.3 Hybrid Recommendation System

In hybrid recommendation, we combine two different approaches to recommendation generation into one algorithm for more effective and stable recommendation generation. CF-based and CBF-based recommendation systems face difficulties when data is sparse, and so hybrid techniques are often used to overcome the limitations of both content-based and collaborative filtering methods. In a hybrid music recommendation system, we combine the CF approach with different music content information to overcome the cold-start problem and improve the model's accuracy. Wang et al. proposed a model for music recommendation generation using deep learning technology. They include the Mel frequency of the audio data with user preferences for music recommendations using the probabilistic matrix factorization model [64]. Yoshii et al. proposed a methodology that integrates rating and content data using a Bayesian network called an aspect model. They offered a hybrid three-way aspect model that includes unobservable users' preferences in latent factors using the Bayesian network [65]. Another model is proposed by Cheng et al. based on deep learning to learn the sequential properties of music with other metadata information of music. This hybrid model of music recommendation is a session-based CF model utilizing time sequence information of audio data with user preference information in recommendation generation [66]. Tahmasebi et al. proposed a hybrid model using the profile expansion technique. They include demographic information of the user with rating information for music recommendations to overcome the sparsity issue of recommendation system [67].

2.2 Literature Review in Diversification of Recommendation System

Users and their interaction history are generally used for recommendation generation in most recommender systems. The overall goal of these systems is to provide the most desirable personalised recommendation to a user. Traditional recommendation methods like collaborative filtering and content-based recommendation methods aim to learn a practical prediction function characterising user-item interaction history. Such a personalized approach can sometimes lead to monotonous recommendations with limited variety, and so can decrease user satisfaction over time. This has been realized for a long time in the literature and has led to making diversity one of the goals of recommendation systems. There are various ways of diversification in recommendation systems:

1. **User Preference-based Diversification**: User attribute preference-based diversification is based on user attributes. The user attributes in diversification are used to understand users and their preferences in the recommendations. For instance, in the case of a music or movie recommender system, the topic might mean genres and item diversification is done based on genre probabilities for the active user [68].
2. **Item Attribute-based Diversification**: Item attribute-based diversification in recommendations suggests generating a completely different recommendation for target users from their past preferences. In item attribute diversification, we introduce more varying items in a recommendation for users, as compared to systems centred only on personalization. This can be calculated by representing the items using attributes and then concentrating on understanding the dissimilarity of the attributes [69].
3. **Explanation-based Diversification**: In explanation-based item diversification,

we generate a recommendation to explicitly increase the diversity factor. The diversity factor is a dissimilarity score calculated for the item using Jaccard diversity distance. Roughly speaking, the terms explanation here refers to the strategy used for generating recommendations. Using this approach can lead to varied results in item diversity since it is possible to have different explanation sources for items even when the items are intrinsically similar [70].

2.2.1 Diversity in Recommendation System: State-of-the art Model based on Machine Learning and Deep Learning

These approaches have successfully altered the simple popularity-based item list traditionally used by many web applications like Amazon, Spotify, and Netflix. Also, such recommendation strategies focus only on accuracy in the design and evaluation of recommendation systems [71, 72]. Earlier approaches suffer from drawbacks like redundancy, where recommending items based on the user's past interaction history limits their interaction domains, i.e., a reduction of diversity in the model. Recent research has advanced from popularity-based recommender systems to ones that consider additional users and item information for a diversified recommendation generation [73]. Bradley and Smyth first introduced diversification in 2001 and opened the potential for new developments [74]. They also considered diversification as a possible solution for the over-fitting problem.

Earlier, many retrieval and mining algorithms were proposed for recommendation generation, which only considered the most relevant and top-rated items for a recommendation from a large set of items. For instance, suggesting highly rated movies to users, the most listened songs for the user and the most visited news article for the recommendation. Various ranking methodologies are proposed for diversity in these systems, which re-rank items for diversification in the recommendation list [75–77]. Re-ranking approaches are the most usual way of including diversity in the recommen-

dation system. This is a two-step approach in which a recommendation list is first retrieved. Then the re-ranking algorithm is run on that list to generate the diverse recommended list by optimising the objective function, which explicitly performs a trade-off between relevance and diversity.

Here we discuss some recent and promising approaches for diversity in the recommendation system. Apart from the greedy re-ranking algorithm, Intent-Aware Diversification is used in recommendation systems to re-rank the recommendation list using various aspects of users. These aspects can be defined explicitly or implicitly. Implicit aspects can be derived from the user's history, like latent factors, and explicit aspects are a set of features used to describe items and users. These are some machine learning-based approaches. Apart from these, some deep learning approaches are also used for diversity awareness in the recommender system. Esmeli et al. ([78]) proposed a session-based personalization for diverse recommendations. Diversity was included in the recommendation list by adding items that depended on the diversity level of the last interacted item of the session. Similarly, Hu et al., ([79]) also proposed a deep learning model for a session-based recommender system that uses the context information for personalized diversity. This approach is also a rearranging approach that re-ranks the recommendation list using users' relevance to the item based on some given context. Apart from single context consideration in RS, many authors incorporated multiple aspects in improving diversity in the recommendations. Oliveira et al. proposed a multiobjective method for diversity and accuracy consideration, which includes content information like contemporaneity, gender, genre, and locality. The multiobjective optimization is achieved using Pareto optimality ([80]). Nassife et al. proposed a diversification method in music recommendation using the Jaccard swap diversity and submodular diversity optimization method ([81]). Volokhin et al. proposed a diversity-infused recommendation model based on users' intent. The user's intent is obtained from a survey and used as a context value for music recommendations ([82]).

He et al. proposed multi-source subtopics, a framework for subtopic modelling that uses a random walk-based approach to estimate the similarity between subtopics extracted from multiple Web sources and then regularize the similarity relations to construct document content ([83]).

Recent works are using the traditional approach for a recommender system. Su et al. ([84]) propose a set-oriented framework for diversity using a matrix-factorization method based on users' context information explicitly for acquiring personalized diversity. Wang et al. ([85]) included personalized diversity using the similarity network for better user influence on the recommender system. This approach uses a similarity network to connect the similarity function and bipartite graph to improve the resource-allocation process. Chen et al. ([86]) proposed a deep learning approach for diversification. They used a sequential recommendation model with intent mining for diversity enhancement. The method uses an implicit intent mining approach to mine user intent automatically. Another concept introduced in the recommendation system for diversity using a machine learning algorithm is called DPP (Determinantal Point Process). DPP is a complementary and encouraging approach used as a probabilistic model for negative correlation, sampling, conditioning, marginalization, and many other inference tasks. Wilhelm et al. ([87]) included the DPP process in their work for YouTube video recommendation. DPP is a naive approach for learning diversity without using any side information, so the computational cost of this model grows exponentially because possible sets of items also increase exponentially. We also summarize related work in Table 2.4.

2.2.2 Diversity in Recommendation System: State of the art model based on GNN

Diversity in recommendation systems is well studied using reranking methods of various machine learning and deep learning-based techniques. For example, diversity-aware

Table 2.4: Summary of literature review

Ref	Category	Approach	Description
[88]	Machine learning	Reranking Method	Proposed a pairwise reranking model that learns user and item factors by minimizing an objective function that includes item dissimilarity irrespective of recommendation list size.
[89]	Graph Neural Network	Reranking method	The model builds a heterogeneous preference network to record user preferences and uses a heterogeneous graph attention network for node aggregation. Two different stages: Matching stage concentrates on accuracy, and the reranking module more concentrates on diversity.
[90]	Deep Learning	Classification	They proposed a method to quantify users based on their music consumption diversity, i.e., generalist and specialist. They investigate how algorithmic recommendations relate to consumption diversity.
[91]	Deep Learning	Reranking method	They proposed a method for fairness-aware variation of the Maximal Marginal Relevance (MMR), using a pre-trained CNN model with MMR to obtain fairness in the recommendation.
[92]	Machine learning	Reranking method	They proposed a xQuAD based model on the reranking method for long-tail items. The model is used for control of popularity bias in the recommendation system.
[93]	Deep Learning	Multi-aspect	This work is based on finding out the subgraph of a similar author for the target author using multiple aspects like temporal context (time, day, etc.), short text (tweets).
[94]	Deep Learning	Multi-aspect re-ranking	They proposed a re-ranking approach to a fairness-aware recommendation that learns individual preferences across multiple fairness dimensions for users to provide a fair recommendation.

Table 2.5: Summary of literature review (contd.)

Reference	Summary	Results	Diversity
[95]	The StartGCN model has been proposed to overcome the cold-start problem of the recommendation system. The proposed model uses a set of GCN encoders and decoders with enhanced intermediate monitoring for recommendation prediction.	The model performs best for inductive rating prediction while considering moderate neighbours for the target node.	The model does not focus on diversified recommendation generation.
[44]	This model follows a random walk-based technique for recommendation generation. They use a sublinear Vertex Reinforced Random Walk to achieve diversity in recommendations.	The model performance is evaluated on a real dataset, Meetup. The evaluation of the model is performed using Precision and NDCG.	The model includes diversity by learning the importance of relations among nodes.
[96]	The paper presents in-depth analysis of diversity in recommendation systems. They compared seven approaches to the recommendation system for absolute and relative diversity. They also compare the post-processing method and diversity optimization method.	Survey for various diversity methods.	-
[97]	The author proposed a model for a social recommendation based on three graphs: the user-item interaction graph using implicit feedback, the social graph of users and their friends, and the other graph is the user-item relation graph.	The model performs best for rating prediction while considering the social interaction of the target user.	The model is not concerned with diversified recommendation generation.

reranking can be achieved by incorporating diversity-promoting objectives into the optimization process. These objectives can be based on metrics such as novelty, coverage, or user preference diversity. However, these reranking methods limit the impact of recommendation quality and add a layer of complexity to the recommendation system, making it harder to explain and interpret the rationale behind the final recommendations [35, 73, 98].

In one of the earliest graph-based models, Zhu et al. [99] proposed a method named GRASSHOPPER, which ranks items according to diversity. The diverse item selection in this method is based on a sequential algorithm, and selected nodes are switched to the absorbing state. Nandanwar et al. [44] proposed a method named Div-HeteRec to evaluate the effect of each relation in the Heterogeneous Information Network (HIN). The authors use the Vertex Reinforced Random Walk method to learn the influence weights between the nodes. Xie et al. [89] proposed a method named GraphDR, which is a two-phased matching and ranking model. The matching module focuses on the coverage of items, while the ranking module generates specific ranks for these items. Sun et al. [43] proposed an approach to model the uncertainty in the user-item interaction graph using the Bayesian graph convolutional Neural Network framework using node copying graph generative model. They introduced a Bayesian probabilistic ranking training loss to introduce diversity and mitigate the data sparsity problem.

Apart from diversity concerns, some authors explicitly introduced a trade-off method between accuracy and diversity. Isufiet al. [42] proposed a technique based on the concept of collaborative filtering. They included a nearest-neighbour random walk and a farthest-neighbour walk in the node embedding generation phase; after that, they had a joint training model with a graph convolution network. Another trade-off is introduced by Joorabloo et al. [100], where a framework of recommendations with controlled levels of precision and diversity was introduced. The authors also conducted a user study to examine where the accuracy or diversity of recommendations is essential

for users. The user study helps decide whether users like diversity and, if yes, to what extent. The summary of the related research for GNN-based recommendation systems is described below in Table 2.5.

2.3 Preliminaries

In this segment, we discuss some theoretical and mathematical concepts which have been used in the rest of the thesis.

2.3.1 Similarity Measures

Similarity measures have been used for similarity calculation between two users and items. We primarily use Cosine Similarity for the user and item similarity calculation.

Cosine Similarity: Cosine similarity measures the similarity between two vectors. The vectors represent a low-dimensional representation of user and item information. It is measured by the cosine of the angle between the two vectors and determines whether the two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis.

$$Similarity(x, y) = cos(\theta) = \frac{x.y}{|x|\,|y|} \tag{2.4}$$

In Equation 2.4 x and y represent the two different vectors. In a recommendation system, it may be two different users or items representations.

2.3.2 Musical Instrument Digital Interface (MIDI)

In November 1981, a standard for digitally encoding music was initially introduced, and later MIDI (Musical Instrument Digital Interface) was designed in 1983. MIDI is a set of protocols that enables communication between computers, musical instruments,

and other music hardware. MIDI does not include any audio signals. Instead of audio signals, digital signals are sent, which carry information about the music. MIDI is a specification for a set of digital codes for transmitting musical score and timing control information by using a series of binary values (0 and 1's) in 8-bit size messages, which support data rates of up to 31,250 bits per second. Thus, MIDI messages are 8-bit long, which means they can carry 256-long values. A MIDI serial transmits one bit at a time. In a MIDI data stream of 10-bit words, the first bit is the start bit, and last is the stop bit, and the rest 8 are for the desired information. MIDI messages consist of a single status byte followed by 0, 1, and 2 data bytes. The status byte of the channel message contains 4 bits, which indicates the channel information. MIDI messages include Note On and Note Off messages, which are generated when the key is pressed and released sequentially [101–103]. Rajyashree et al. [104] proposed a methodology based on the jSymbolic library[6], which extracts features from the MIDI data. Then a machine learning technique like the Random Forest, Logistic Regression, etc., is used for recommendation generation [104].

2.3.3 Graph Neural Network

Recently, the advances in graph neural networks have proved instrumental in addressing the issues mentioned above for recommender systems. GNN adopts a node embedding representation technique using the neighbourhood aggregation method that includes high-level structural information of nodes. GNN-based methods have become the new state-of-the-art approaches in recommender systems with advantages in handling structural data [105]. The GNN models can be categorized into two kinds of models: spatial models and spectral models. Spectral models are based on the concept of signal processing and graph convolution in the spectral domain, and spatial models are based on the convolution of graph structure, which directly extracts the localized feature using

[6] http://www.music.mcgill.ca/~cmckay/NEMA/publications/ICMC_2006_jSymbolic.pdf

weighted aggregation, e.g., GCN (graph convolution network) [106]. Both models are based on the nearest neighbour concept for node representation to capture the high-order correlation between nodes and edges. GCN was introduced relatively recently for recommender systems and has been extended by many authors [107]. GCN is a spectral model that combines neural network and graph convolution to achieve semi-supervised classification.

2.3.4 Heterogeneous Graph Neural Network

Other popular variants of GNN are the graph representation techniques used to transform each node and edge into a low-dimensional vector while preserving their structural information. Recently a variety of methods have been used for graph embedding generation. These embeddings are generated on the node, sub-graph, metapath level or using random walk. Some popular techniques like node2vec, deepwalk, LINE (Large-scale Information Network Embedding), and SDNE (Structural Deep Network Embedding) are used for node embedding generation. These are conventional network embedding methods that apply to the homogeneous network [108–111].

Heterogeneous network embedding was first introduced by Yu et al.(2014), where they used different metapaths for latent representation of user and item [38, 112]. We also include the metapath2vec algorithm in our proposed approach for the latent representation of each user and item. Metapath2vec in a heterogeneous graph is used to form a link between different users and items that model their relationship. Zhao et al. (2020) use a metapath algorithm for embedding generation of users and items. They propose a model named HetNERec, which integrates various types of information extracted from heterogeneous networks to enhance recommendation performance using the matrix factorization technique [113]. Fu et al. (2020) use the metapath algorithm for heterogeneous node embedding generation. The concept called MAGNN is based on three phases of input node content transformation, intra-metapath aggregation to

incorporate intermediate semantic nodes and inter-metapath aggregation to combine messages from multiple metapaths [114]. Shi et al. (2018) propose a method named HERec where they use metapaths based random walk to generate node embedding, and further use these embeddings into matrix factorization framework for recommendation generation [115].

Feng et al. (2012) propose a method named optrank using social tag information. This method is used to alleviate the cold start problem using heterogeneous information of social networks [116]. Hu et al. proposed an approach for a new recommendations that considers users' long and short-term interests. They consider user interaction history and topic modelling with GNN and the attention-based LSTM network for recommendation generation [117]. Zheng et al. (2021) proposed an approach for user and item entity representation that considers information on product type and attribute data in their representation using the attentional attribute and interaction method. The overall concern of this proposed method is to overcome the sparsity problem [118]. Zheng et al. (2017) propose a method based on dual similarity regularization to impose constraints on users and items with varying similarities using meta paths. The authors manually set the metapaths for different domain-specific scenarios [119]. Wang et al. (2022) proposed an approach for personalized recommendation using multidimensional metapaths on a dimension-free temporal graph probabilistic spreading framework to automatically learn the priority and importance at the user level granularity [120].

2.4 Evaluation Metrics

We conduct extensive experiments to demonstrate our proposed model's accuracy and diverse embedding quality on real datasets. Our model's effectiveness is analyzed through various offline experiments measured by some well-known measures used for the recommender system. Our proposed approach focuses on recommending items to users with implicit feedback.

1. **RMSE**: Two main measures also evaluate recommender system accuracy: Root Mean Squared Error (RMSE) and Mean Absolute Error(MAE). MAE is defined as below:

$$MAE = \frac{1}{|R|} \sum_{u,i \epsilon R} |\hat{r}_{u,i} - r_{u,i}| \tag{2.5}$$

One characteristic of MAE is it does not allow biases to be extreme in error terms due to the presence of outlier (rating having a more significant error). It will consider those outliers evenly to the other predictions. Root Mean Squared Error is more likely to be influenced by outliers or wrong predictions because it tends to penalize significant errors by applying a square root to make it smaller. If the model generated predicted ratings $\hat{r}_{u,i}$ for a test set R of user-item pair (u, i) for which true rating $r_{u,i}$ are known:

$$RMAE = \sqrt{\frac{1}{|R|} \sum_{u,i \epsilon R} (\hat{r}_{u,i} - r_{u,i})^2} \tag{2.6}$$

2. **AUC Score**: We optimize the area under the ROC curve (AUC) metrics for experimental analysis. The AUC measure can measure the recommendation ability to distinguish the positive and negative items. For a user u the recommender AUC is defined as follows:

$$\text{AUC-Score} = \frac{1}{|I_u^+| \left| \frac{I}{I_u^+} \right|} \sum_{i \in I_u^+} \sum_{j \in \frac{I}{I_u^+}} f(r_{u_{ij}}) \tag{2.7}$$

Where I_u^+ is the list of positive items rated by user u, and $\frac{I}{I_u^+}$ are the list of negative items not-rated by the user u and $r_{u_{ij}} = r_{ui} - r_{uj}$.

3. **Precision**: In data mining, precision is an accuracy kind of measure to determine when the values of False Positive are important. Precision is the fraction of relevant documents with all retrieved documents. Precision is used to test the

relevance of the result set.

$$precision = \frac{relevant\ document \cap retrieved\ document}{retrieved\ document} \tag{2.8}$$

4. **Recall**: The recall is the proportion of actual positive predicted items to all positive items in the actual class:

$$recall = \frac{relevant\ document \cap retrieved\ document}{relevant\ document} \tag{2.9}$$

5. **Normalized Discounted Cumulative Gain (NDCG)**: The NDCG measure is used to evaluate the recommendation model's ranking quality. NDCG refers to the summation of the fraction of discounted system gain upon the discounted ideal gain for a rank p, where the summation of the relevance from rank 1 to rank p is known as the gain at a rank p:

$$\mathrm{DCG_{sp}} = rel_{s_1} + \sum_{j=2}^{p} \frac{rel_{sj}}{log_2(j)} \tag{2.10}$$

$$\mathrm{IDCG_{p}} = rel_{i_1} + \sum_{j=2}^{p} \frac{rel_{ij}}{log_2(j)} \tag{2.11}$$

$$\mathrm{NDCG_{p}} = \frac{\mathrm{DCG_{sp}}}{\mathrm{IDCG_{p}}} \tag{2.12}$$

where rel_{sj} refers to the relevance score attributed to item j by the user.

6. **Diversity**: Aggregate Diversity of a recommended list is calculated using Intralist Diversity (IL-D) measure defined in equation (4.10). IL-D is the average dissimilarity between items present in the recommendation list size of R for each user:

$$\text{IL-D} = \frac{2}{|R|\,|R-1|} \sum_{i \in R} \sum_{j \in R, i \neq j} 1 - sim(i,j) \tag{2.13}$$

where $sim(.,.)$ is the similarity function which measures the similarity value (conventionally normalized between 0 to 1) for a given pair of items. For example, similarity metrics will be cosine similarity or Pearson co-relation measure or any other similarity measure.

7. **Coverage**: In recommendation, system coverage is how many items appear in the recommended results or top-n recommended result. Coverage is defined as below:

$$\text{Coverage}(M) = \bigcup_{u} R_{M,k}(u) \tag{2.14}$$

Coverage for model M is $R_{M,k}$, which is a test set R for user u retrieved from model M. A higher value for coverage is better and shows that the model recommends a wide range of items.