Determinantal Point Processes for Producing Diversified and Personalized Predictions

Yuli Liu

A thesis submitted for the degree of
Doctor of Philosophy at
The Australian National University

October 2023

Except where otherwise indicated, this thesis is my own original work.

Yuli Liu
17 October 2023

Acknowledgments

I would like to express my most sincere gratitude and appreciation to the people who have supported me in making this thesis possible.

I would first like to thank my supervisor, Prof. Lexing Xie, for giving me the opportunity to join the doctoral journey and helping me to establish good research habits. Her kindness and thoughtfulness always make me feel comfortable and confident. I also would like to thank my primary supervisor, Dr. Christian Walder, who is one of the excellent researchers and one of the coolest people I have ever met in my life. He can always enlighten me using outstanding skills in mathematics when I get stumped by a research question. I would then like to thank my co-supervisor, Dr. Cheng Soon Ong, for providing me useful advice and treating me kindly. It was an honor to work and communicate with them.

I would like to extend my sincerest thanks to my lab mates at the ANU Computational Media Lab, especially Quyu Kong, Siqi Wu, Alasdair Tran, Rui Zhang, Shidi Li, Mengyan Zhang, and many others, for their encouragement and insightful comments.

I am grateful to the Australian Government Research Training Program and Data61, CSIRO for providing my scholarships, and I thank the National Computational Infrastructure (NCI) for providing computational resources.

Finally and most importantly, I would like to express deep thanks to my wife Changting Sheng, my parents Qingxiang Liu and Fuxiang Wang, and my elder sister Yule Liu, for their unreserved love and help.

Abstract

Producing personalized predictions, *i.e.*, filtering redundant information and locating desired parts as predictions of a specific user's information needs according to his/her historical interactions, plays an indispensable role in online applications. In related literature, personalized predictions are popularly produced by modeling within the framework of recommender systems such as traditional recommendation and sequential recommendation that model users' static and dynamic preferences towards items, respectively. Recently, personalized predictions of individual elements have been extended to the set level where subsequent sets (each with arbitrary number of elements) are predicted given previous sets, namely Temporal Sets Prediction (TSP). However, most of the existing models that propose to provide personalized predictions are built with the purpose of shrinking the discrepancy between predictions and targets (*i.e.*, improving the accuracy of predictions), leading the provided suggestions to be monotonous in terms of topics or categories. To enrich the suggestions and broaden users' horizons, diversifying personalized predictions has become a trending research topic. This thesis presents our efforts in incorporating diversification and personalization by means of the widely adopted elegant probabilistic models — Determinantal Point Processes (DPPs).

First, an intuitive design of adopting the maximum a posteriori (MAP) inference over DPP distribution for generating diverse and relevant suggestions is implemented in traditional recommender systems. We directly formulate the inherent goal of diversity-promoting recommendation as a two-objective optimization problem by simultaneously minimizing recommendation errors and maximizing diversity, which are integrated into Generative Adversarial Nets (GAN). In addition to the common practice of generating DPP-distributed diverse item subsets using MAP inference in the training process, a conditional DPP samples generation algorithm is applied in the serving (evaluation) procedure to ensure that the learned diversity-promoting recommendation model is capable of providing users with expected services.

Next, the dependencies among items that constitute the DPP-distributed subsets are noticed and explored. Based on which, we propose two objective functions for sequential recommendation that are both dependency-aware and diversity-aware, utilizing DPP-distributed likelihoods of standard DPPs and conditional DPPs, respectively. In contrast to commonly used objective functions (*e.g.*, binary cross-entropy

and pairwise ranking BPR), which primarily prioritize accuracy and are usually formulated under the assumption of independence among estimated items, our proposed objective functions for sequential recommendation stand out because they explicitly consider dependencies and incorporate diversity as a guiding factor.

Then, an interesting discovery of ranking interpretation behind the probabilities of k-DPP distributed subsets inspires the proposal of set-level k-DPP probability based ranking optimization for recommendation. In this thesis, we formalize the set-level relevance and diversity ranking objectives on basis of DPP kernel decomposition. The probabilities of subsets with cardinality k learned from the specialized k-DPP distribution over a fixed size ground set enable us to compare and rank multiple items at the set level with consideration of both relevance and diversity. We implement the k-DPP based ranking in the context of both Matrix Factorization (MF) and neural networks approaches on three real-world datasets, obtaining improved relevance and diversity performances. The k-DPP ranking approach is broadly applicable, and when applied to seminal CF models it also yields strong performance improvements, suggesting that this technique holds significant value to the field of recommender systems.

Last, associating the set-level predictions in the field of TSP with structures of structured DPP (SDPP), which are both comprised of arbitrary number of elements, we present a *Structured Temporal Sets Prediction* framework (STSP), which explores and exploits three types of underlying relationships (*i.e.*, preference correlation, co-occurrence pattern, and temporal sets dependency). STSP makes two significant contributions to this field. First, elements in a set of temporal sequence are directly modeled as a SDPP structure, in which a DPP distribution is modeled over collections of structures (temporal sets) instead of over subsets of individual items. Second, all trainable representations and underlying relationships are integrated into a dedicated sets-level objective function with the intention of enhancing the probability of selecting the desired subset of structures as a DPP. Accordingly, we treat temporal sets in a broader perspective compared to existing combination manners, enabling the underlying relationships to be explicitly represented.

Altogether, we explore an interesting research task of producing diversified and personalized predictions, which is implemented in varied contexts of item-level recommendation and set-level TSP. To achieve diversification and meanwhile boost accuracy performance, the repulsion character of DPP is exploited from multiple aspects, as well as dependencies among elements are captured. The intuitive design of dual consideration of diversity in both training and serving procedures, the noticed dependency across training sequence instances, the interesting discovery of k-DPP probability specified ranking implication, and the association between set-level predictions and SDPP structures hold significant value for further studies.

Publications and Software

Published and under-reviewed papers are summarized as follows. A list of corresponding software packages is also provided.

Major Publications in Thesis

- **Yuli Liu**, Christian Walder, Lexing Xie. "Determinantal Point Process Likelihoods for Sequential Recommendation." *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022. (Full paper, in Chapter 4)
- **Yuli Liu**, Christian Walder, Lexing Xie. "Learning k-Determinantal Point Processes for Personalized Ranking." *In submission.* (In Chapter 5)
- **Yuli Liu**, Christian Walder, Lexing Xie. "Structured Determinantal Point Processes for Temporal Sets Prediction." *In submission.* (In Chapter 6)
- **Yuli Liu**, Christian Walder, Lexing Xie. "Promoting Recommendation Diversity: Dual Consideration in Both Training and Serving Procedures." *In submission.* (In Chapter 3)

Other Publications

- **Yuli Liu**. "Recommending Inferior Results: A General and Feature-Free Model for Spam Detection." *Proceedings of the 29th ACM International Conference on Information and Knowledge Management* (*CIKM*), 2020. (Full paper)
- **Yuli Liu**. "Signed Latent Factors for Spamming Activity Detection." https://arxiv.org/pdf/2209.13814.pdf. *In submission.*
- **Yuli Liu**, Christian Walder, Lexing Xie. "Joint Modeling of Generative and Discriminative Models for Diversity-Promoting Recommendation." *In submission.*

Software

- DPPLikelihoods4SeqRec Public: source code. (In Chapter 4)
  https://github.com/l-lyl/DPPLikelihoods4SeqRec
- k-DPP4Ranking Public: source code. (In Chapter 5)
  https://github.com/l-lyl/k-DPP4Ranking
- SDPP4TSP Public: source code. (In Chapter 6)
  https://github.com/l-lyl/SDPP4TSP

Contents

List of Figures

List of Tables

Chapter 1

Introduction

To avoid users being overwhelmed by vast amounts of information, numerous online platforms, such as e-commerce, web logs, and social networking, have invested in producing personalized predictions, serving to filter redundant information and locate desired parts to satisfy the information needs of a specific user. For instance, on Netflix, the personalized recommendation system tailors movie suggestions based on a user's viewing habits, favoring genres they frequently enjoy, like action and thrillers, while avoiding genres they rarely explore, such as romantic comedies or documentaries. This customization enhances the user's streaming experience. In this sense, the task of recommender systems can be viewed as producing personalized predictions for users, as recommendations are usually provided by modeling users' preferences towards items on the basis of historical interactions between users and items [Fang et al., 2020; Dai et al., 2021].

Existing studies in this domain focus predominantly on providing accurate results to match user preferences [Bai et al., 2019; Kang and McAuley, 2018; He et al., 2017b; Tang and Wang, 2018], underscoring the significance of tailoring predictions to individual preferences. However, this approach harbors inherent flaws, as an exclusive focus on personalization can inadvertently result in offering monotonous and analogous suggestions that [Wu et al., 2019b; Liang and Qian, 2021], while aligning with users' interests, encompass a limited breadth of topics. Two major factors are linked with this situation. First, *prediction methodology*, which is usually designed following the strategy of selecting top elements (*e.g.*, items, products, and blogs) ranked by their match to users' personal preferences and interests, often yields suboptimal recommendations [Liu et al., 2014b; Hu et al., 2014]. For example, collaborative filtering favors popular items and thus easily recommends popular items that are already known to the user [Ashkan et al., 2015]. Content-based filtering may provide items that match user interests but cover a very narrow scope of topics [Qin and Zhu, 2013; Wu et al., 2019b]. Sequential models (*e.g.*, Markov Chain and RNNs) usually learn temporal preferences based on general elements, *i.e.*, popular or similar items that are frequently exposed

to customers, causing a rare exposure of potential but unpopular items [Kim et al., 2019]. Second, *objective function* used for training and guiding personalized prediction models, such as binary cross-entropy [He et al., 2017b], Bayesian Personalized Ranking (BPR) [Rendle et al., 2012], and listwise ranking [Wang et al., 2020], is generally confined in the direction of shrinking the discrepancy between predictions and targets (*i.e.*, improving the accuracy/relevance of predicted results), which may cause a bottleneck in producing accuracy but monotonous results [Liu et al., 2022].

To better serve and satisfy users, an important metric, *diversity*, has been proposed to cover a wider range of users' interests by providing predictions from various categories or topics. Diversity plays an important role in broadening users' horizons and helping online service providers to explore users' potential interests [Cheng et al., 2017]. Three general groups of efforts have been made to promote the diversity of predictions, *i*) adopting post-processing heuristics that generate a candidate prediction set based on the accuracy metric at first and then selects a few elements by maximizing the diversity metric [Ashkan et al., 2015; Qin and Zhu, 2013]; *ii*) incorporating extra operations into end-to-end learning tasks, such as adding the randomness [Sun et al., 2020b] and introducing the normalized topic coverage metric [Cheng et al., 2017] in the learning process; *iii*) using DPP-based algorithms to introduce diversity in predictions, *e.g.*, the maximum a posteriori (MAP) inference for generating diverse samples [Chen et al., 2018; Wu et al., 2019b] and the probabilistic diversity measurement of sets for estimating specific items' diversity during the training process of recommendation models [Liu et al., 2020; Gan et al., 2020]. Although these approaches provide some insights into understanding and promoting diversity for predictions, they still suffer from inherent limitations. The applicability of the first (second) group of studies is limited, as extra data processing (parameter turning) is needed, and in most cases, diversity is achieved by sacrificing accuracy (*i.e.*, the so-called *trade-off strategy*) [Wu et al., 2019b]. The DPP-based approaches in the third group have been highlighted due to the efficient and exact probabilistic algorithms for modeling repulsive correlations. However, most of existing studies only dedicate themselves to simply incorporating DPP algorithms (*e.g.*, MAP and marginalization) into mature recommendation models or frameworks like Graph Neural Networks (GNN) [Gong et al., 2022] or Reinforcement Learning (RL) [Liu et al., 2021a; Weng et al., 2021] for diversity-aware prediction.

These approaches have made limited progress in effectively balancing personalization and diversification, as they have not significantly transformed existing models, and there remains untapped potential in fully exploring the capabilities of DPPs [Liu et al., 2022]. Therefore, in this thesis, we engage in targeted research on the attributes of the DPP models, including diversity measurement, conditional DPP distribution, and marginal DPP probability. These attributes are closely associated with recommenda-

tion tasks at both the individual and set levels. Consequently, this strategic approach enables the simultaneous realization of dual objectives, ensuring the relevance and enhancing the diversity of recommendation results.

1.1 Research Questions

In this thesis, we explore more potential and deeper exploitation of DPPs compared to previous work on diversity promotion. The following content presents our efforts in guaranteeing both diversification and personalization of produced predictions, where new perspectives, as well as new insights are provided into related research fields. In an effort to provide clarity, we highlight that our research questions have been inspired by the identified limitations in existing studies.

The research topic of producing diversity-aware predictions in the recommendation field is generally named diversity-promoting recommendation task, which has been viewed as a two-objective optimization problem, *i.e.*, to minimize the recommendation errors (irrelevant items) of a recommendation list and to maximize the diversity within items in a list. To achieve diversity-promoting recommendation, an intuitive design, applying MAP inference to generate a diverse and relevant DPP subset as top-n recommendations, is commonly employed [Chen et al., 2017; Gan et al., 2020]. However, in previous studies, there have been separate approaches for conducting MAP inference during generation, with some focusing on the training procedure alone [Wu et al., 2019b], while others only consider it during serving or evaluation [Chen et al., 2018]. However, none of these approaches have effectively modeled the two-objective optimization problem. Our first question is therefore inspired: **can we directly formulate the two-objective optimization problem and consider diversity in both training and serving procedures to guarantee the overall performance of diversity-promoting recommendation?**

Sequential recommendation, a particular personalization prediction task that holds real-world scenarios, has been a trending research topic. Extensive efforts have been made in designing sequence process models or exploiting powerful deep learning based frameworks [Kim et al., 2019; Tang and Wang, 2018; Kang and McAuley, 2018] for sequential recommendation, which directly use common loss functions like BPR and binary cross entropy with the independent assumption of items when calculating the loss. There is little work focusing on proposing dedicated objective functions, acknowledging naturally existing dependencies in sequences. Naturally, the second research question is proposed **can we shed new insights into the objective function of sequential recommendation to take the dependency across the training sequence into account by means of DPP likelihoods?**

Producing personalized predictions is naturally a ranking problem, as highly relevant items can be selected for a specific user when the entire item list is ranked according to their relevance to the user's preferences. To achieve this ranking task, ranking optimization such as BPR [Rendle et al., 2012] and listwise ranking [Xia et al., 2008; Rahimi et al., 2019] has been widely employed, especially in collaborative filtering based recommender systems [He et al., 2017b; Covington et al., 2016]. However, existing algorithms are accuracy-oriented (*i.e.*, solely comparing relevance) and are still constrained in the item-level comparison without achieving real set-level ranking that is capable of capturing dependencies among items in a set. We then can ask the third question: **can we introduce DPP into optimization criterion to implement real set-level diversity and accuracy ranking comparisons by exploring the ranking interpretation behind DPP distributions?**

Temporal sets prediction is a challenging research problem considering the complex relationships within the set and across the set sequence. The common approaches of addressing this task are still following the popular sequence models (*e.g.*, RNN [Hu and He, 2019] or transformer [Wu et al., 2019a]) with the manner of combining latent representations of items in a set to represent the set. There are no real set-level representations and objective functions dedicated to temporal sets. We therefore ask the fourth question: **can we design an integrated framework with set-level representation and set-level objective function to fit into the temporal sets prediction by treating the temporal set as an integrated element (structure) of structured DPP?**

In summary, four research questions are put forward, driven by the need to address existing issues across distinct but interconnected research topics. These research questions illuminate unexplored areas and offer novel viewpoints that suggest the imperative for a more profound and expansive exploration of DPP than is currently undertaken in existing studies.

1.2 Thesis Overview

This thesis primarily focuses on investigating the research questions and elaborates on their solutions in the following structure. Figure 1.1 summarizes this thesis using a roadmap diagram.

Chapter 2 presents introduction and summary of studies that are closely related to this thesis. The most common applications of producing personalized predictions, *i.e.*, recommender systems including traditional and sequential recommendation, are reviewed at first. Subsequently, we discuss the recommendation approaches that consider diversity metric, namely, diversity-promoting recommendation. We then review

the temporal sets prediction tasks. This chapter ends with elegant probabilistic models — DPPs, which are popular in diversification tasks.

In Chapter 3, we detail the answer to the first research question mainly from two aspects: *i*) two objective functions corresponding to the two objectives of diversity-promoting recommendation are proposed to promote diversity and meanwhile to boost accuracy, which are integrated into the GAN framework [Goodfellow et al., 2014]; *ii*) in addition to the commonly used MAP inference [Chen et al., 2018; Gan et al., 2020; Cheng et al., 2017] for DPP generation in the training process, a new conditional DPP distribution-based MAP inference is designed to serve users with diversified and personalized recommendations. These recommendations are produced on the basis of diversity-aware representations trained via the two proposed objective functions.

Chapter 4 is dedicated to address the second question. We first analyze the characteristics of sequential recommendation, which inspire us to find that taking dependencies across training sequences (with previous items and subsequent targets) into loss calculation is considerable. Two likelihoods calculated via the definitions of standard DPP distribution and conditional DPP distribution are then proposed to formalize objective functions oriented towards sequential recommendation tasks, which capture the dependencies among targets and across the entire sequence of training instances, respectively. To demonstrate the importance of two formulated dependencies, specific experimental settings are designed by varying the length of the training sequence and the target number in real-world datasets.

In Chapter 5, we answer the third research question by constructing an overall (based on the kernel decomposition of quality *vs.* diversity) DPP kernel at first, which facilitates assigning high DPP probabilities to subsets that contain relevant and diverse elements. We then design a specialized DPP distribution on the cardinality k (*i.e.*, k-DPP [Kulesza and Taskar, 2011a]), which is endowed with the ranking interpretation. A new ranking optimization can be implemented by directly comparing the probabilities of any two k-DPP distributed subsets based on the overall kernel. In this setting, not only the diversity comparison, but also the relevance comparison is considered in the ranking procedure. What is more, the formulated probabilities of k-DPP subsets compared for ranking really capture the dependencies among the items in the subsets.

Chapter 6 sets out a detailed answer to the fourth research question. First, three types of specific relations are extracted from the complex relationships within temporal set sequence. To formulate these relations, in addition to the common sequence preference representation [Fang et al., 2020; Yu et al., 2020a], a new type of co-occurrence representation is defined. We then represent the structure of SDPP [Gillenwater et al., 2012a] using these two types of representations, which means that the arbitrary number of items in a temporal set is explicitly represented as an integrated structure. Taking

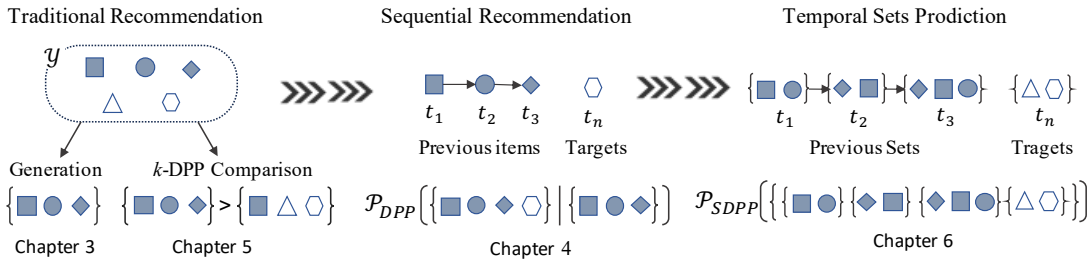


Figure 1.1: The solid shapes represent observed items of interest, while hollow shapes represent unobserved items. The ground set is denoted as $\mathcal{Y}$, comprised of items under consideration. Each chapter of this thesis contributes to specific domains. Chapter 3 utilizes the DPP MAP algorithm to generate diversified and personalized items within a GAN framework. Chapter 5 achieves ranking optimization through the comparison of k-DPP probabilities for k-sized item subsets. Chapter 4 introduces a novel set of DPP-based objective functions for sequential recommendation by treating it as the selection of a completed true sequence based on known previous items. Chapter 6 represents temporal sets as SDPP structures and predicts them under the SDPP distribution.

a broad view of structures instead of individual items, the dedicated set-level objective function is finally proposed, which is naturally applicable to the temporal sets prediction task and directly combines three types of relations.

Finally, Chapter 7 summarizes this thesis along with several future research directions in this area.

1.3 Key Contributions and Impact

The main contributions of this thesis include new designed frameworks for producing personalized and diversified predictions (*e.g.*, DDPR in Chapter 3 and STSP in Chapter 6), as well as new learning objectives in three research fields, *i.e.*, traditional recommendation (Chapter 5), sequential recommendation (Chapter 4), and temporal sets prediction (Chapter 6).

- **A diversity-aware GAN framework** is built, which integrates two objectives of diversity-promoting recommendation into the adversarial training. The diversity of generated items is maximized by formulating the DPP kernel and quantifying the repulsion strength. The relevance of recommendations is further boosted through minimizing the errors between predictions and true labels. To guarantee the diversity performance of recommendation, we propose to consider diversity not only in the training process, but also in the serving procedure. Therefore, a conditional DPP distribution-based MAP inference is constructed, which considers the correlations between historical interactions and predictions

when providing users with recommendations. The goal of producing relevant and diverse predictions was successfully attained by means of diversity-aware adversarial training and conditional DPP generation. (Discussed in Chapter 3)

- **Defining and exploring the importance of two types of dependencies among training instances for sequential recommendation.** This thesis introduces the first work that is aware of the importance of dependencies within the observed interaction sequences (training instances) in the calculation of loss function for sequential recommendation. Two types of dependency relationships, *i.e.*, dependency within targets and dependency across training sequences (with previous items and subsequent targets), are defined and captured, whose considerable importance is demonstrated by experimental results on real-world datasets. (Discussed in Chapter 4)

- **Two dependency-considered objective functions dedicated to sequential recommendation.** We formulate the two defined dependencies via the likelihoods of standard DPP and conditional DPP. Based on which, two dedicated objective functions that respectively optimize the probabilities of selecting target set and true interaction sequence as DPP-distributed subsets, are developed for sequential recommendation. (Discussed in Chapter 4)

- **Endowing tailored k-DPP probabilities with ranking interpretation.** The ranking implication behind specialized DPP distributions conditioned on the cardinality k is first discovered and demonstrated. By constructing an overall DPP kernel that balances diversity and relevance, we are able to formulate the optimization criterion for personalized ranking from a brand new perspective that considers two types of set-level ranking comparisons, *i.e.*, topic coverage ranking and relevance ranking. (Discussed in Chapter 5)

- **Designing k-DPP based probabilistic ranking optimization approaches.** Using the probability of k-DPP subset distribution over specialized ground set as a comparison metric, we create two new ranking optimization methods: one solely considers rankings from the perspective of target subset; the other takes both inclusion of target set and exclusion of negative set into account. Significant superiority is achieved by our ranking approaches on both accuracy and diversity performances. (Discussed in In Chapter 5)

- **Explicitly representing temporal sets with arbitrary number of items as SDPP structures.** The complex and arbitrary construction of temporal sets is first explicitly represented via the structure formulation of SDPP, enabling researchers to easily calculate and employ the set sequence from a broad view.

A new type of representation of individual elements dedicated to capturing the co-occurrence patterns of the challenging temporal sets prediction task is proposed. Comprehensive evaluation is performed to demonstrate the effectiveness of explicit structure representation and co-occurrence patterns. (Discussed in Chapter 6)

- **A structured temporal sets prediction framework** is developed, which focuses directly on structure-level representations by incorporating three types of well defined and formulated correlations in temporal sets prediction task. The SDPP based objective function for this integrated framework is also proposed. (Discussed in Chapter 6)

Related Work

Producing personalized predictions is a general research topic, with recommender systems being the most prevalent approach. For example, traditional recommender systems utilize each user's observed interactions that are treated equally to model static preferences of users via content-based [Deshpande and Karypis, 2004] and collaborative filtering-based [Hu et al., 2008] techniques for anticipating which unobserved items are likely to be relevant. Sequential recommender systems aim to predict the next items under the condition of a known previous temporal sequence of items. In this setting, users' dynamic preferences are modeled by capturing dependencies among sequence-specific items using sequence processing approaches such as Markov Chain [Davidson et al., 2010] and Recurrent Neural Networks (RNNs) [Hu and He, 2019; Choi et al., 2016]. These methods can dynamically measure the importance of items (that are involved into the sequence) to subsequent items. This type of dynamic prediction makes sequential recommendations more transferable to real-world scenarios, as users' preferences are easily influenced by previously acquired information. In addition to these tasks of recommending individual elements as predictions, producing subsequent sets (each with an arbitrary number of elements) as personalized predictions given previous sets, namely Temporal Sets Prediction (TSP) [Hu and He, 2019; Sun et al., 2020c], has recently been a trending and challenging research topic.

We introduce the studies related to recommender systems, including two popular tasks, *i.e.*, traditional recommendation and sequential recommendation, in Section 2.1. Ranking optimization approaches for collaborative filtering (a popular technique of traditional recommendation) and common loss functions for sequential recommendation are respectively introduced in Section 2.1.1 and Section 2.1.2, as we propose new ranking optimization insights in Chapter 5 and new objective functions for sequential recommendation in Chapter 4. Since this thesis is motivated by diversifying personalized predictions, we subsequently review the closely related research topic, namely diversity-promoting recommendation in Section 2.2. Section 2.3 discusses the set-level personalized prediction, *i.e.*, temporal sets prediction, which is explored in Chapter 6.

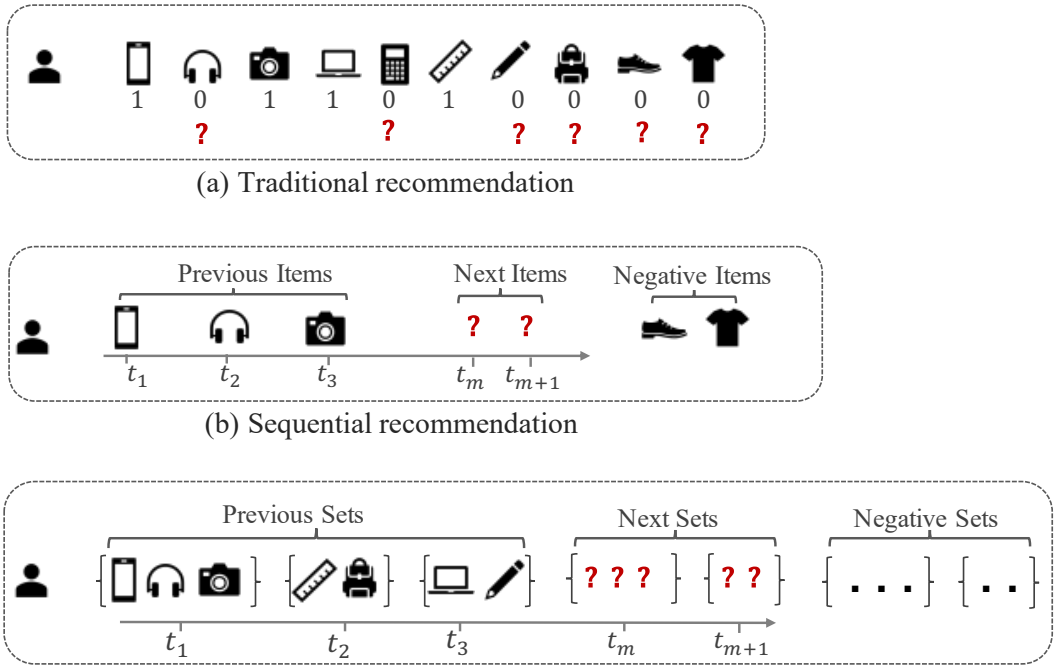
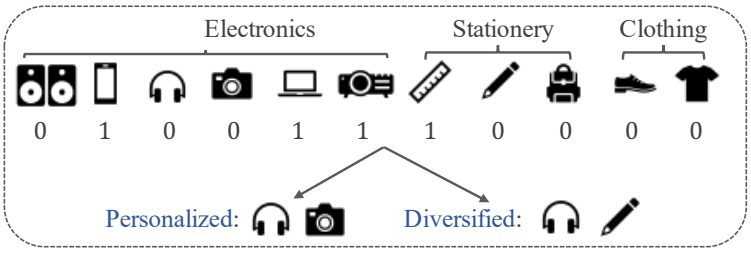


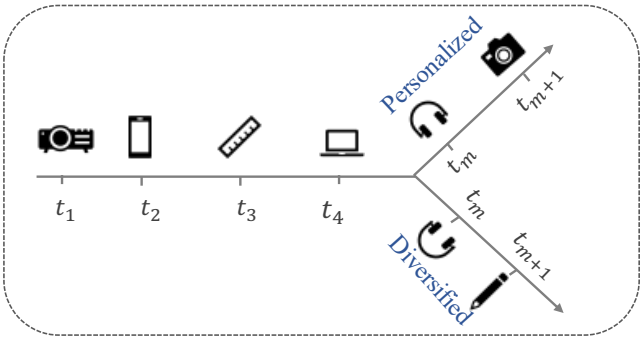
Figure 2.1: Illustration of three popular personalization prediction tasks. The user's implicit feedback is defined as 1 for item i when there is an interaction between the user and item i and 0 for unobserved interactions.

Lastly, we not only introduce related studies of determinantal point process (DPP) in this chapter, but also present its preliminaries, as DPP (the main focus of this thesis) is not as well known as the recommendation concept.

Before individually introducing studies of different fields related to this thesis, we present a general illustration of three popular applications of personalization prediction in Figure 2.1 to intuitively summarize connections and differences of these three fields. Aligning with three subfigures of Figure 2.1, traditional recommendation (Figure 2.1a) conspicuously differs from other two tasks (sequential recommendation in Figure 2.1b and temporal sets prediction in Figure 2.1c), as there is no timestamp involved. In terms of traditional recommendation, user preferences are usually reflected by implicit feedbacks (*e.g.*, purchasing products and clicking items), where 1 denotes observed interactions and 0 means other cases (*e.g.*, a user does not like the item or the user is not aware of it). Traditional recommender systems then try to capture users' preferences for providing recommendations based on observed implicit interactions with the assumption that all user-item interactions are equally important. The timestamp information that marks dynamic influences is therefore not involved. We can treat traditional recommendation as a matrix completion task, *i.e.*, predicting which 0 entries should be filled with 1 given the current 1/0 matrix with observed users-items interactions.



(a) Diagram of diversity for traditional recommendation



(b) Diagram of diversity for sequential recommendation

Figure 2.2: Illustration of diversification on traditional recommendation and sequential recommendation. Listed items are grouped into three categories. A common metric of diversity is to provide a prediction list with items in various categories.

Essentially, sequential recommendation and temporal sets prediction are both sequential prediction tasks, which focus on modeling users' dynamic preferences based on known temporal data. The main difference between them is the level of predicted elements. For sequential recommendation, the predicted elements are individual items. As for temporal sets prediction, the subsequent sets (each with arbitrary number of items) are needed to be predicted. During the training process, it is common practice to select negative samples (items or sets) that users have not interacted with in order to calculate the loss measure. As the basic element of the temporal sets prediction task is a set, the relationships within the set sequence is more complex compared to sequential recommendation [Hu and He, 2019].

We display a comparison example between personalized recommendation and diversity-aware recommendation in Figure 2.2. There are three categories of products: electronics, stationery, and clothing. According to the user's observed interactions, traditional recommender systems (Figure 2.2a) can easily find that he/she prefers electronics over items of the other two categories, and then recommend closely similar items (such as headset and camera in electronics), *i.e.*, matching the user's interests but covering a very narrow scope of topics/categories [Qin and Zhu, 2013]. This type of recommendation (solely pursues the match of preferences) is inclined to produce monotonous

results and confines the user's exploration to specific topics. On the other hand, if the diversity is considered, corresponding recommender systems will try to broaden the coverage of predictions' category and then facilitate mining users' potential preferences in a broadened horizon. Consequently, a pencil that belongs to stationary category and happens to be of interest to the user is likely to be recommended, as there is also an observed interaction in this category. In this sense, diversifying the predictions offers a chance to satisfy users' potential interests. We need to notice that camera and pencil are likely to be both relevant to the user's preferences, and the reason that pencil rather than camera is selected by diversity-aware recommendation is due to the diversity factor brought by the pencil. This means that diversification is not unselectively pursuing diversity, but promoting the diversity of personalized results. The meaning of diversification in temporal prediction tasks (*e.g.*, sequential recommendation shown in Figure 2.2b) is to diversify subsequent suggestions that are likely relevant to the user's preferences, predicted based on previous interactions. In sequential recommendation, as illustrated in Figure 2.2b), the objective is to introduce diversity in subsequent suggestions while ensuring their relevance to the user's preferences, which are predicted based on their previous interactions.

2.1 Recommender Systems

Recommender systems have long been acknowledged as an effective and indispensible tool for addressing the problem of information overload [Fang et al., 2020]. They enable users to easily filter abundant information and locate desired results in terms of their preferences, and allow online platforms to widely publicize their promotional literature. In this section, the widely studied research topics, traditional recommendation and sequential recommendation, are reviewed.

2.1.1 Traditional Recommendation

There are mainly two lines of traditional recommendation techniques: collaborative filtering-based methods and content-based methods. In this subsection, only the most related and popular line, Collaborative Filtering (CF), is reviewed. In addition, the widely employed ranking optimization approaches for training CF based recommendation models are also introduced.

Collaborative Filtering. The main meaning of CF is to model users' preferences on items based on historical interactions [He et al., 2017b; Wang et al., 2019a; Liu et al., 2010]. Among the various collaborative filtering techniques, matrix factorization (MF) and neural CF are widely recognized as the most popular approaches. Models based on MF learn user and item embeddings based on low-rank decomposition of the

user-item interaction matrix [Koren, 2008; Koren et al., 2009]. The huge success of neural networks encourages researchers to represent the inner product with a neural architecture [He et al., 2017b]. Some studies treat user-item behaviors as a user-item bipartite graph, and design neural graph models [Wang et al., 2019c; He et al., 2020; Chen et al., 2020], in which the crucial collaborative signals are distilled and employed.

VAECF [Liang et al., 2018] is a unique model that first extends the generative model — variational autoencoders (VAE) to collaborative filtering. Unlike most of the CF models, there are no learned embeddings of all users and items for relevance prediction in VAECF. Another generative model, Generative Adversarial Nets (GAN), has also been an effective and popular methodology for recommender systems. IRGAN [Wang et al., 2017b] is a seminal work that applies GAN to recommendation tasks, which unifies generative and discriminative retrieval models to improve relevance capacity through adversarial training. To address the cold start problem, a GAN-based recommendation model combines multiple generators to generate users from multiple perspectives of items' attributes [Sun et al., 2020a]. Some studies [Gao et al., 2019; Tong et al., 2019] incorporate different neural models (*e.g.*, graph neural network and variational auto-encoder) into the GAN framework for improving the ability of learning user preferences. GAN has also been applied to social recommendations [Krishnan et al., 2019] based on a modular adversarial framework.

The work of diversity-aware GAN framework in Chapter 3 can be viewed as a collaborative filtering approach, as it relies on aggregated user historical data. Furthermore, our optimization criterion approaches proposed in Chapter 5 is also implemented in the CF context.

Ranking Optimization. Bayesian Personalized Ranking (BPR) is the most popular pairwise approach for implicit feedback-based recommendation [Wan et al., 2022], which treats each pair of elements independently, as well as ignores the correlations of multiple observed items and multiple unobserved items. Listwise approaches measure the ranking loss by calculating the distance between predicted list and true list [Xia et al., 2008; Cao et al., 2007; Shi et al., 2010]. SetRank [Wang et al., 2020] provides a new research perspective for listwise learning with implicit feedback by using the permutation probability to encourage one observed item to rank in front of a set of unobserved items in each list. As our ranking optimization based on k -DPP probability comparison in Chapter 5 focuses on implicit feedback, and SetRank has achieved state-of-the-art ranking results [Chen et al., 2021; Yu et al., 2020b], other list-based ranking-based methods are not compared in experiments. Set2SetRank [Chen et al., 2021] attempts to construct ranking pairs at the set level by comparing the individual items of a set, which still uses the BPR optimization criterion.

Chapter 5 of this thesis provides a brand new perspective into ranking-based op-

timization approaches, which goes beyond the confines of the BPR criterion and individual comparison. It is achieved by comparing the probabilities of subsets being DPP-distributed, *i.e.*, real set-level ranking comparison.

2.1.2 Sequential Recommendation

Models in Sequential Recommendation. The main factor that sets the sequential recommendation apart from conventional recommender systems (*e.g.*, *Top-N* recommendation and MF-based recommendation) is the consideration of sequential dependency. To capture the sequential dependency for completing the user interaction sequence, Markov Chain based [Rendle et al., 2010; He and McAuley, 2016a] and Recurrent Neural Networks based [Tang and Wang, 2018; Kang and McAuley, 2018; Sun et al., 2019] methods have dominated the literature and real-word applications. In Chapter 4, we propose DSL and CDSL loss functions targeting the next item(s) estimation. The next-item(s) recommendation [Yuan et al., 2019; Tang and Wang, 2018] is a common task that calculate a loss based on the next one item (or multiple items) in the training process. The 'next item(s)' in Chapter 4 actually means the next T targets for estimation in training. By preparing user sequences with L previous items and T targets for training, we aim to introduce the targets dependency into the loss formulation, and then facilitate recommendation models providing users with a set of relevant results. The session-based recommendation [Hidasi et al., 2015; Li et al., 2017] is a kind of non-personalized sequential recommendation, which uses sequential session data without user profile/identifier for recommendation.

We believe that our proposed objective functions (in Chapter 4) are also adaptive to session-based recommendations, as the sequence dependency is also considered in their field. This will be further explored in the future.

Loss Functions in Sequential Recommendation. In the sequential recommendation literature, two types of loss functions are widely employed, *i.e.*, binary cross-entropy and pairwise ranking (BPR). Many different variations are also developed based on these two loss functions, *e.g.*, softmax cross-entropy [Covington et al., 2016] and pairwise hinge loss [Hsieh et al., 2017]. BPR is usually limited to having only one target and one negative sample at each time step [Tang and Wang, 2018; Yu et al., 2019; Socher et al., 2013], but some studies have expanded the BPR formulation to a set-wise Bayesian approach [Wang et al., 2020; Rendle et al., 2010] to consider more comparisons. The seminal session-based recommendation model GRU4Rec [Hidasi et al., 2015] proposes the TOP1 loss, which is similar to the BPR loss, and the improved version GRU4Rec$^+$ [Hidasi and Karatzoglou, 2018] further boosts the recommendation performance using a special case of binary cross-entropy, *i.e.*, categorical cross-entropy.

Recently, inspired by the widely used contrastive loss [Hadsell et al., 2006; Yang et al., 2019] in the computer vision tasks, a line of work [Xie et al., 2020, 2021] employs the contrastive loss in sequential recommendation models, which is defined similarly to the softmax cross entropy loss. The nature of above-mentioned loss functions is still relevance-based ranking [Hidasi and Karatzoglou, 2018], and the dependencies in sequence and among targets are ignored. In addition, the temporal dependency is complex [Wu et al., 2019a], merely considering the dependency of items' ranking positions in a list (*e.g.*, list-wise ranking) is not enough [Chen et al., 2021]. Therefore, we treat sequential recommendation as a task of set selection in Chapter 4 using the elegant probabilistic model — DPP, instead of directly comparing the rankings of items.

2.2 Diversity-promoting Recommendation

Diversity-promoting recommender systems with the goal of recommending diverse and relevant results to users have received significant attention. Early work in this line of research [Ashkan et al., 2015; Carbonell and Goldstein, 1998; Zhang and Hurley, 2008] often adopts a post-processing strategy that generates a set of candidate elements based on the accuracy metric at first and then selects a few items by maximizing the diversity metric. This type of two-step process needs extra parameter tuning to balance diversity and relevance. To consider both quality and diversity in a unified learning task, some methods improve recommendation diversity in the manner of end-to-end supervised learning, *e.g.*, the diversified collaborative filtering (DCF) [Cheng et al., 2017] that needs to choose a trade-off parameter for promoting diversity.

Some recent approaches employ the negative correlation of DPP to reduce the chance of recommending similar items. DPP generates diverse lists through the maximum a posteriori (MAP), which is NP-hard and computationally expensive [Liang and Qian, 2021]. To address this issue, a novel fast greedy algorithm is developed [Chen et al., 2018] to accelerate the MAP inference for DPP. Based on this algorithm, more recent studies [Wu et al., 2019b; Gan et al., 2020] employ DPP to improve diversity in different recommendation tasks. These studies mainly focus on the construction of a DPP kernel over the ground set (entire items) for MAP inference generation, which is inspired by the quality *vs.* diversity decomposition of DPP. The diversity component of DPP kernel is usually pre-learned based on the observed diverse subsets that users interact with [Wu et al., 2019b].

2.3 Temporal Sets Prediction

Unlike widely studied time series forecasting [Zhou et al., 2021; Wu et al., 2021] and sequence prediction [Bengio et al., 2015; Chiu et al., 2018], temporal sets prediction is proposed to untangle complicated relationships among items in the set and dependencies across temporal sets for predicting subsequent sets. Hu et al. [Hu and He, 2019] first formulate this task and propose a set-to-set method for temporal sets prediction. To capture the dependencies across temporal sets, popular sequence-related models are employed, *e.g.*, multi-heads self-attention [Sun et al., 2020c] and RNN [Hu and He, 2019]. Recently, some studies have attempted to propagate information between sets to model the complicated relationships. For example, Yu et al. [Yu et al., 2020a] perform graph convolutions on dynamic relationships graph to learn comprehensive set representations. Another study [Yu et al., 2022] attempts to integrate collaborative signals into temporal sets by propagating element-guided message. Although this work claims that the propagation is on set-level, the user-set interaction graph is actually constructed based on interactions between individual items and users instead of directly via set to set. In addition, previous TSP studies still use the individual item-level loss function (*e.g.*, weighted mean square loss [Hu and He, 2019] and binary cross-entropy loss [Yu et al., 2022; Sun et al., 2020c]) for training, in which complicated relationships are ignored in loss formulation.

Compared to previous related studies, this thesis (Chapter 6) formulates the temporal set as an element (structure) and designs a structured probabilistic distribution based objective function to fit the set-level prediction.

2.4 Determinantal Point Processes based Methods

Determinantal point process (DPP) offers a promising and complementary approach, focusing on measuring negative correlations, which has been widely employed to improve diversity in different tasks, such as recommendation[Gan et al., 2020; Chen et al., 2018] and machine translation [Song et al., 2018; Dhoolia et al., 2021]. The extensions of DPP, *i.e.*, k-DPP and SDPP that are implemented in this thesis, have received relatively less attention compared to the standard DPP.

A DPP is a distribution over subsets of a fixed ground set, for instance, sets of items provided by recommender systems. Formally, a DPP $\mathcal{P}$ on a discrete set $\mathcal{I} = \{i_1, i_2, \ldots, i_N\}$ is a probability measure on $2^{\mathcal{I}}$, the set of all subsets of $\mathcal{I}$. When $\mathcal{P}$ gives nonzero probability to the empty set, there exists a matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ such that for

every subset $Y \subseteq \mathcal{I}$, the probability of Y is

$$\mathcal{P}(Y) \propto \det\left(\mathbf{L}_Y\right), \tag{2.1}$$

where $\mathbf{L}$ is a real, positive semidefinite (PSD) kernel matrix indexed by the elements of $\mathcal{I}$, and $\mathbf{L}_Y$ is obtained by retaining the rows and columns of $\mathbf{L}$ that corresponds to elements in Y. Following Equation 2.1, we can observe that if given a random subset Y drawn according to $\mathcal{P}$, the marginal probability of including one element i is $\mathcal{P}\left(i \in Y\right) = L_{ii}$, and two elements i and j is

$$\begin{aligned} \mathcal{P}(i, j \in Y) &= \begin{vmatrix} L_{ii} & L_{ij} \\ L_{ji} & L_{jj} \end{vmatrix} \\ &= L_{ii}L_{jj} - L_{ij}L_{ji} \\ &= \mathcal{P}(i \in Y)\mathcal{P}(j \in Y) - L_{ij}^2 \end{aligned} \tag{2.2}$$

If we think of the entries of the kernel as measurements of similarity between pairs of elements in $\mathcal{I}$, then highly similar elements are unlikely to appear together. Thus, the negative correlations between pairs of elements are determined by the off-diagonal elements. That is, a large value of L_{ij} reduces the likelihood of both elements coexisting in a diverse subset.

Besides the DPP-based approaches for diversity-promoting recommendation mentioned in Section 2.2, extensive studies have focused on learning parameters of DPP kernels learning [Affandi et al., 2014; Gillenwater et al., 2014] and sampling DPP-distributed subsets [Calandriello et al., 2020; Li et al., 2016]. Standard DPP is also applied to basket completion task [Warlop et al., 2019], which views the basket completion task as a multi-class classification problem. It leverages ideas from tensor factorization and multi-class classification to design the multi-task DPP model. In Chapter 3, diversification of traditional recommendation in the training process is mainly achieved by directly maximize the diversity metric (measured by the standard DPP) of item subsets from the generator component of GAN. In terms of applying DPP to sequential recommendation (Chapter 4), we directly design two objective functions that not only consider diversity, but also capture dependencies in the calculation of loss.

In the standard setup of DPP, a probability is assigned to every subset (including the empty set and the entire elements). This is not desirable in practical problems that require explicit control over the number of items selected by the model, such as image search engines that provide a fixed-sized array of results on a page [Kulesza et al., 2012]. If a model based on a standard DPP gives some probability of suggesting no or all images, this is not ideal. Given this situation, an extension of DPP, *i.e.*,

k-DPP [Kulesza and Taskar, 2011a], has been proposed, which conditions a DPP on the cardinality k of the random set. This extension solves an important practical problem, and necessitates new algorithms for efficient inference based on recursions for the elementary symmetric polynomials [Kulesza et al., 2012].

Sampling from k-DPP holds important practical implications, and some efficient sampling algorithms have been proposed. α-DPP [Calandriello et al., 2020] is a recent k-DPP sampling algorithm that adaptively builds a sufficiently large uniform sample of data that is then used to efficiently generate a smaller set of k items. Meanwhile, it ensures that this set is drawn exactly from the target distribution defined on all items. There are a few studies that directly employ k-DPP for diversification aware tasks, *e.g.*, video summarization [Zheng and Lu, 2021] and diverse mini-batch selection [Huang et al., 2019]. In this thesis (Chapter 5), we explore the ranking interpretation behind the tailored k-DPP distribution for ranking optimization, instead of focusing on sampling or learning.

To handle a complex situation where each element of a set is actually a structure, the structured determinantal point process (SDPP) is proposed [Kulesza and Taskar, 2010]. Structured DPP opens up new possibilities for applications; we are allowed to model diverse sets of essentially any structured objects. For instance, we could find not only the accurate translation but also a diverse set of high-quality translations for a sentence, serving as alternatives for human translators. SDPP offers a natural probabilistic model over sets of structures (*i.e.*, temporal sets). In SDPP case, elements of $\mathcal{Y}$ are structures, which means that we will no longer think of $\mathcal{Y} = \{1, 2, \ldots, M\}$; instead, each element $\boldsymbol{y} \in \mathcal{Y}$ is a structure given by a sequence of R parts $\left(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(R)}\right)$, each of which takes a value from a finite set of N possibilities, where $N = M^R$.

An immediate challenge is that the kernel $\mathbf{L}$ of SDPP, which has N^2 entries, can no longer be written down explicitly. The previous work therefore defines its entries using the quality/diversity decomposition [Kulesza et al., 2012]. In Chapter 6, the SDPP ground set is selected according to a temporal sequence instead of building over entire items, making the SDPP kernel efficiently computable. The original SDPP is proposed to select salient and diverse treads in document collections [Gillenwater et al., 2012a]. Previous studies that employ SDPP for diversification are not abundant; they focus on tasks of bundle list recommendation [Bai et al., 2019] and answer summarization [Pande et al., 2013]. We first introduce SDPP into temporal sets prediction (in Chapter 6) and a structured framework with SDPP level objective function is built.

Chapter 3

Generating Diverse and Relevant Top-n Recommendations using Determinantal Point Processes

As one of the most common implementation of producing personalized predictions, traditional recommender systems primarily make efforts on producing relevant results for users at the beginning. Diversifying recommendations to broaden user horizons and explore potential interests has emerged as a prominent research area. Although numerous efforts have been made to enhance the diversity of recommendations, the trade-off between diversity and accuracy remains a significant challenge. The primary causes for this problem lie in the following two aspects: (*i*) the inherent goals of diversity-promoting recommender systems, which are to simultaneously deliver accurate recommendations and cater to a broader spectrum of user interests, have not been adequately explored; (*ii*) considering diversity factor only in model training procedure can not guarantee the provision of diversification service in recommender systems.

In this chapter, we directly formulate the inherent goals of diversity-promoting recommendation as a two-objective optimization problem by simultaneously minimizing the recommendation error and maximizing diversity. These two proposed objectives are integrated into Generative Adversarial Nets (GAN) to guide the training process toward the orientation of boosting both diversification and recommendation accuracy. We draw inspiration from Determinantal Point Process (DPP) to maintain diversity by maximizing DPP log-determinant over generated subsets. In addition, we propose a conditional DPP samples generation algorithm through the MAP inference in the serving (evaluation) procedure to ensure that the learned diversity-promoting recommendation model is capable of providing users with expected services. Experimental results demonstrate that our model outperforms existing methods in terms of both diversity and relevance in recommendation results.

3.1 Introduction

Traditional recommender systems generally fall into two groups, *i.e.*, collaborative filtering (CF) based and content based methods [Ashkan et al., 2015; Liang and Qian, 2021]. They strive to match items' relevance to the preferences of a specific user via developed models. This match adjustment is guided by objective functions, with the purpose of reducing the discrepancy between expected scores and true labels that are actually observed. These relevance-oriented objectives naively recommend top-ranked relevant items but neglect the diversity among them, making recommender systems tend to produce items that match a user's interests but covering a very narrow scope of topics (categories) [Qin and Zhu, 2013]. To surprise users with potential topics instead of making them get bored with monotonous suggestions, the diversity-promoting recommendation has been proposed and becomes a leading research topic in the recommendation field [Kunaver and Požrl, 2017], which aims not only to maintain the relevance of a recommendation list, but also to diversify the categories/topics of items in the list, *i.e.*, achieving two goals.

Early work in this field often adopts a post-processing heuristic. Two types of heuristic methods, *i.e.*, tuning a parameter to diversify the final recommendations [Zhang and Hurley, 2008; Carbonell and Goldstein, 1998] and defining a submodular objective function to learn the diversified model [Agrawal et al., 2009; Qin and Zhu, 2013], are widely used to maintain the trade-off between relevance and diversity of recommended items. Recently, some studies have tried to model the diversity of recommendation sets depending on repulsive correlations of Determinantal Point Process (DPP) [Chen et al., 2018; Wu et al., 2019b; Warlop et al., 2018]. While these methods made progress to some extent, there remain limitations: (*i*) tuning trade-off parameters of the heuristic methods is not effective and efficient enough [Cheng et al., 2017], and other heuristic methods with submodular functions end up being a suboptimal solution [Ashkan et al., 2015]; (*ii*) Most DPP-based methods first learn the DPP **L**-kernel on the entire set of items and then define a personalized user kernel by trading off items' relevance for diversity among items [Wu et al., 2019b; Warlop et al., 2018]. Learning the extra $N \times N$ **L**-kernel, where N represents the number of entire items, is computationally expensive. Furthermore, existing approaches fail in considering two crucial factors that contribute to balancing diversity and accuracy in recommendation systems: (*i*) The inherent goals of diversity-promoting recommendations, *i.e.*, a two-objective optimization problem (to minimize recommendation errors and to maximize diversity [Liang and Qian, 2021; Liu et al., 2019]), are not well formulated; (*ii*) They usually directly employ pre-learned diversity-promoting recommendation models, which may cause losing part of diversification ability in practical service, as the

diversity factor is only considered in the model design (training) procedure. To overcome these limitations, a **d**ual consideration for **d**iversity-**p**romoting **r**ecommendation framework (DDPR) is proposed, in which the diversity factor is considered in both training and serving procedures.

In the training procedure, we formulate the two inherent goals of diversity-promoting recommendation as two optimization objectives, *i.e.*, maximizing diversity and minimizing recommendation errors, which are incorporated into Generative Adversarial Nets (GAN). The adversarial training setting of GAN has been widely employed in different recommendation tasks [Krishnan et al., 2019; Wang et al., 2017b; Zhang et al., 2021]. Nevertheless, most GAN-based works are relevance-oriented, which means that the generator learns to generate relevant items to fool the discriminator, and the discriminator tries to draw a clear distinction between the ground-truth relevant documents and the generated ones. In this common adversarial mechanism, the generator is gradually guided to generate personalized samples from a few modes (*i.e.*, relevant items that cover limited categories) for each user. As a result, the generator is likely to rotate through a small set of output types, causing a mode collapse problem [Arjovsky et al., 2017]. In this work, we refine the generator component by designing two particular objective functions to guide the training process not only to the orientation of accuracy but also to diversification. Unlike existing approaches that formulate DPP kernel over the entirety of item catalog, we measure the DPP diversity based on trainable representations of item subset that is related to a specific user. In this way, the diversity of item subset can be directly calculated in the learning process based on corresponding items' embedding vectors, which means that: (*i*) The computation efficiency is promoted, as the diversity is measured across item subset instead of across entire items; (*ii*) The extra pre-learned parameters commonly used for calculating DPP kernel [Wu et al., 2019b; Warlop et al., 2018] over entire items are not needed. As we diversify items and pursue recommendation accuracy through optimizing two objectives simultaneously, the trade-off parameter that is often employed in previous methods [Chen et al., 2018; Wu et al., 2019b] is not required as well.

In the serving procedure, the maximum a posteriori (MAP) inference is performed over the conditional DPP distribution based on the trained and diversity-aware item representations, which takes the correlations between observed interactions and potential interests of users into account. Drawing upon the dual consideration (training and service) of diversity, users are further guaranteed with relevant and diversified results in practical applications.

This chapter makes the following main contributions:

- We propose a dual consideration diversity-promoting recommendation framework that first considers diversity in both training and serving procedures.

Table 3.1: The Key Symbols Used in DDPR Framework.

Notation	Interpretation
$I_{\mathcal{T}}$, $I_{\mathcal{G}}$	true item set, generated item set
$p_{\text{true}}(u)$, $\mathcal{G}_\theta(u)$	true data distribution, generator distribution
$\mathbf{e}^u_{\mathcal{G}}$, $\mathbf{e}^i_{\mathcal{G}}$	user and item representation from generator
$\mathbf{e}^u_{\mathcal{D}}$, $\mathbf{e}^i_{\mathcal{D}}$	user and item representation from discriminator
$\mathbf{L}_{(u,I_{\mathcal{G}})}$	DPP kernel on generated items of user u
$\mathcal{J}_{Div}$	diversity objective function
$\mathcal{L}_{Sup}$	supervised loss
$\mathcal{L}_{\mathcal{D}}$	discriminator objective function
$\mathcal{L}_{Gen}$	generator loss function
$\mathbf{L}_u$	personalized kernel for MAP generation

- We incorporate two proposed objectives into the GAN learning mechanism to formulate the inherent goals of diversity-promoting recommendation.
- Instead of using the common naive Top-N selection to evaluate the recommendation model, we propose a conditional generation method, *i.e.*, recommended lists are generated through DPP MAP inference given the known positive feedbacks.
- Experimental results show that DDPR is superior both in terms of quality and diversity compared to several state-of-the-art baselines.

3.2 Methodology

This section first presents the preliminaries of DPP and GAN framework (customized to the recommendation task), and then explains how to formulate the two-objectives optimization on diversity and quality, as well as how to generate items through MAP inference based on personalized DPP kernel. The important notation used in this chapter is summarized in Table 3.1.

3.2.1 Preliminaries: Properties of DPPs

DPPs are elegant probabilistic models that arise in quantum physics. They model the likelihood of selecting a subset with diverse items as the determinant of a kernel matrix. As DPPs offer efficient and exact algorithms for sampling, marginalization, conditioning, and other inference tasks [Kulesza and Taskar, 2011a], they have been a promising and popular approach for years. Two main essential characteristics of a DPP, *i.e.*, capturing negative correlations and providing a way to elegantly model the

trade-off between quality and diversity, have made DPP an appealing tool for modeling diversity in various applications such as video and text summarization [Kulesza and Taskar, 2011b; Gong et al., 2014], image search and ranking [Kulesza and Taskar, 2011a], or diversity-promoting recommendation [Wu et al., 2019b; Chen et al., 2018; Chen and Ahmed, 2020].

For the purpose of modeling real data, the most common way to construct DPPs is through **L**-ensembles [Kulesza et al., 2012]. An **L**-ensemble defines a DPP via a positive semi-definite matrix $\mathbf{L}$ indexed by the elements of ground set $\mathcal{Y}$. In the discrete case, let $\mathcal{Y} = \{1, 2, \cdots, N\}$ be a ground set of N items (*e.g.*, products or movies). A point process is simply a probability measure on $2^{\mathcal{Y}}$. Let $\mathcal{P}$ be a determinantal point process. When $\mathbf{Y}$ is a random subset drawn according to $\mathcal{P}$, we have, for every $Y \subseteq \mathcal{Y}$,

$$P(\mathbf{Y} = Y) = \frac{\det(\mathbf{L}_Y)}{\det(\mathbf{L} + \mathbf{I})}. \tag{3.1}$$

The kernel $\mathbf{L} \in \mathbb{S}_+^{N \times N}$ represents the parameter of DPP and $\mathbf{I}$ is the $N \times N$ identity matrix. $\mathbf{L}_Y$ denotes the principal minor (sub-matrix) with rows and columns selected according to the indices in Y. In the following, we will use $\mathcal{P}(Y)$ as a shorthand instead of $\mathcal{P}(\mathbf{Y} = Y)$ when the meaning is clear.

By virtue of the determinant function $\det(\cdot)$, the interesting property of pairwise repulsion is contributed. To see that, we can refer to Equation 2.2. If items i and j are the same, then $P(Y = \{i, j\}) = 0$ (because $L_{ij} = L_{ii} = L_{jj}$) [Gong et al., 2014]. This means that identical items should not appear together in the same set. In addition, if i and j are similar to each other, then the probability of i and j coexisting in a subset is going to be less than that of observing either one of them alone. Therefore, the most diverse subset of $\mathcal{Y}$ is the one that attains the highest probability,

$$Y^* = \arg\max_Y P(Y) = \arg\max_Y \det(\mathbf{L}_Y), \tag{3.2}$$

where Y^* results from maximum a posteriori (MAP) inference. Although MAP inference is NP-hard, there are several approaches to obtain approximate solutions [Chen et al., 2018; Gillenwater et al., 2012b].

3.2.2 Generator *vs.* Discriminator Minimax Game

In the recommendation scenario, the generative model (generator) is formulated to generate (or recommend) items that look like the ground-truth (positive) items and therefore could fool the discriminative model (discriminator), whereas the discriminator tries to perform classification, *i.e.*, to distinguish between true and generated data.

Formally, we have the overall objective:

$$J^{\mathcal{G}^*,\mathcal{D}^*} = \min_{\theta} \max_{\phi} \sum_{u=1}^{N} \Big(\mathbb{E}_{I_{\mathcal{T}} \sim p_{\text{true}}(u)} \left[\log \mathcal{D}_{\phi} \left(u, I_{\mathcal{T}} \right) \right] + \mathbb{E}_{I_{\mathcal{G}} \sim \mathcal{G}_{\theta}(u)} \left[\log(1 - \mathcal{D}_{\phi} \left(u, I_{\mathcal{G}} \right)) \right] \Big), \tag{3.3}$$

where $I_{\mathcal{T}}$ and $I_{\mathcal{G}}$ represent user u's ground-truth browsing history and generated item set, sampled from true data distribution $p_{\text{true}}(u)$ and generator distribution $\mathcal{G}_{\theta}(u)$, respectively. θ and ϕ denote the parameters in generator and discriminator. As this equation shows, the optimal parameters of the generator and the discriminator can be learned iteratively by maximizing and minimizing the same objective function.

3.2.2.1 Discriminator

The objective of discriminator is to maximize the log-likelihood of correctly distinguishing the positive and generated items. With the observed browsing history of users, and the ones generated from the current optimal generative model $\mathcal{G}_{\theta^*}$, The corresponding loss function used for training the discriminator is defined as:

$$\mathcal{L}_{\mathcal{D}} = - \sum_{u=1}^{M} \Big(\mathbb{E}_{I_{\mathcal{T}} \sim p_{\text{true}}(u)} \left[\log \mathcal{D}_{\phi} \left(u, I_{\mathcal{T}} \right) \right] + \mathbb{E}_{I_{\mathcal{G}} \sim \mathcal{G}_{\theta^*}(u)} \left[\log(1 - \mathcal{D}_{\phi} \left(u, I_{\mathcal{G}} \right)) \right] \Big). \tag{3.4}$$

$\mathcal{D}_{\phi} \left(u, I_{\mathcal{G}} \right)$ and $\mathcal{D}_{\phi} \left(u, I_{\mathcal{T}} \right)$ evaluate the overall relevance of the generated set $I_{\mathcal{G}}$ and the true interaction set $I_{\mathcal{T}}$ for user u, respectively. For any item i in $I_{\mathcal{T}}$ or $I_{\mathcal{G}}$, we use the following formulation to represent the personalized relevance,

$$q(u, i) = \sigma \left(\mathbf{e}_{\mathcal{D}}^{u} {\mathbf{e}_{\mathcal{D}}^{i}}^{\mathrm{T}} \right), \tag{3.5}$$

where σ is the sigmoid function, and $\mathbf{e}_{\mathcal{D}}^{u}$ and $\mathbf{e}_{\mathcal{D}}^{i}$ are $1 \times D$ embedding vectors from discriminator for user u and item i.

3.2.2.2 Generator

By contrast, the generator tries to *fool* the discriminative model by fitting the underlying relevance distribution over the user's browsing history $p_{\text{true}}(u)$. That is, while keeping the current discriminator $\mathcal{D}_{\phi}$ fixed after its maximization. According to some studies [Wang et al., 2017b], the generative model $\mathcal{G}_{\theta}$, as part of the original GAN formulation, cannot be optimized directly using gradient descent because the samples it generates for recommendation purposes are discrete. A common approach is to

use policy gradient-based reinforcement learning [Wang et al., 2017b; Yu et al., 2017], formulated as follows:

$$\mathcal{L}_{Gen} = -\sum_{u=1}^{M} \mathbb{E}_{I_{\mathcal{G}} \sim \mathcal{G}_\theta(u)} \left[\mathcal{R}\left(u, I_{\mathcal{G}}\right) \log \mathcal{Q}(u, I_{\mathcal{G}}) \right]. \tag{3.6}$$

$\mathcal{R}(u, I_{\mathcal{G}})$ denotes the reward function that in general reflects how much the generator can deceive the discriminator, formally:

$$\mathcal{R}\left(u, I_{\mathcal{G}}\right) = -\log\left(1 - \mathcal{D}_\phi\left(u, I_{\mathcal{G}}\right)\right). \tag{3.7}$$

$\mathcal{Q}(u, I_{\mathcal{G}})$ is a relevance-evaluating function that evaluates the overall relevance of generated item set $I_{\mathcal{G}}$ from u, which can be calculated through the similar method as Equation 3.5, *i.e.*, utilizing the representations of $\mathbf{e}^u_{\mathcal{G}}$ and $\mathbf{e}^i_{\mathcal{G}}$ from generator to evaluate the predicted relevance of each item in $I_{\mathcal{G}}$ to user u.

3.2.3 Diversity and Quality Optimization

One of DPP's significant advantages is that an intuitive trade-off between quality and diversity through the decomposition of DPP kernel $\mathbf{L}$ whose entries can be written as $L_{ij} = q_i \phi_i^\top \phi_j q_j$. We can think that $q_i \in \mathbb{R}^+$ measures the quality (relevance) of an item i, and $\phi_i^\top \phi_j \in [-1, 1]$ denotes a signed measure of similarity between items i and j. Unlike previous work that employs additional diversity-aware parameters or tunable trade-off weight to pursue diversity, we directly measure and maximize the diversity of generated item sets, and meanwhile promote recommendation quality (relevance) via minimizing the difference between predicted preferences and true data distribution.

In the recommendation scenario, diversity and relevance are two types of destinations, *i.e.*, to expand the scope of recommendations' topics and to recommend accurate results. Existing work (not only related to the recommendation task [Wu et al., 2019b; Chen and Ahmed, 2020], but also to text summarization [Gong et al., 2014] or Web search [Kulesza et al., 2012]) has suggested that diversification is usually improved by sacrificing accuracy, as these two objectives are often mutually exclusive, and the focus of recommendation models tends to be placed on just one of them. In this work, we aim to directly formulate both objectives, allowing them to harmoniously coexist and simultaneously contribute to the overall performance of the system. This strategy seeks to strike a balance between accuracy and diversification, maximizing the utility and effectiveness of our approach.

3.2.3.1 Maximize Diversity

Equation 2.2 demonstrates why DPP is 'diversifying'. If we think of the entries of the $\mathbf{L}$ kernel as measurements of similarity between pairs of elements in $\mathcal{Y}$, then highly similar elements are unlikely to appear together. To capture the repulsion correlations, previous related studies usually learn the diverse kernel based on observed diverse sets from all user-item interaction history at first and then construct the personalized user kernel via the quality (predicted scores) *vs.* diversity decomposition [Wu et al., 2019b; Gartrell et al., 2017]. In this work, we directly treat the trainable parameters of $\mathcal{G}_\theta$ (*i.e.*, item embedding vectors from generative model) as the feature representations of items, which are used to calculate the similarity between any pair of items (*e.g.*, item i and j) by Gaussian kernel, formally:

$$\mathbf{L}\left(\mathbf{e}_{\mathcal{G}}^{i}, \mathbf{e}_{\mathcal{G}}^{j}\right) = \exp\left\{-\frac{1}{2}\left(\mathbf{e}_{\mathcal{G}}^{i} - \mathbf{e}_{\mathcal{G}}^{j}\right)^{\top} \Sigma^{-1}\left(\mathbf{e}_{\mathcal{G}}^{i} - \mathbf{e}_{\mathcal{G}}^{j}\right)\right\}. \tag{3.8}$$

We then can formulate the diversity of generated item set $I_{\mathcal{G}}$ as,

$$\mathcal{J}_{Div} = \log\left(\det(\mathbf{L}_{(u,I_{\mathcal{G}})})\right), \tag{3.9}$$

where $\mathbf{L}_{(u,I_{\mathcal{G}})}$ denotes DPP kernel on the generated item set of user u. This formulated set diversity is maximized when the item representations are orthogonal, and therefore it promotes diversity. Through directly using trainable representations of items to calculate the DPP diverse kernel on subsets, we can maximize the diversity of generated items in the training process of GAN. This demonstrates that our approach eliminates the need for the additional trade-off parameter [Chen et al., 2018], and the pre-learned diverse kernel on the entire item sets [Wu et al., 2019b; Gartrell et al., 2017], both of which are commonly utilized in most existing works. This highlights the uniqueness and novelty of our methodology.

3.2.3.2 Minimize Recommendation Error

To ensure that diversity does not come at the cost of quality, we suggest the introduction of a complementary objective that aims to complete the two-objective optimization, specifically, by minimizing the discrepancy between generated items and true labels. Inspired by a supervised version of GAN [Luc et al., 2016; Zhao et al., 2019], we introduce a supervised loss $\mathcal{L}_{Sup}$ to calculate the distance between predicted scores $\mathcal{Q}(u, I_{\mathcal{G}})$ of generated item set ($I_{\mathcal{G}}$) and true labels $\mathcal{T}(u, I_{\mathcal{G}})$, which is defined as follows:

$$\mathcal{L}_{Sup} = \sum_{u=1}^{N} \mathbb{E}_{I_{\mathcal{G}} \sim \mathcal{G}_\theta(u)} \left\| \mathcal{Q}(u, I_{\mathcal{G}}) - \mathcal{T}(u, I_{\mathcal{G}}) \right\|_2^2. \tag{3.10}$$

Based on this function, we can improve the "faking" ability of generator and avoid the recommendation diversity being unduly promoted (*i.e.*, sacrificing quality). In practical experiments, we randomly sample Z negative items for each user, which are added into the item set $I_{\mathcal{G}}$ when calculating $\mathcal{L}_{Sup}$, to ensure that there are enough negative items for comparison.

We now can obtain the final loss function for generator by incorporating the two objectives *w.r.t.* relevance and diversity,

$$\mathcal{L}_G = \mathcal{L}_{Gen} - \mathcal{J}_{Div} + \mathcal{L}_{Sup}. \tag{3.11}$$

By applying this equation, we directly integrate the two objectives of diversity-promoting recommendation into the GAN framework. This incorporation boosts the generator's ability to diversify while also improving the relevance of its recommendations.

3.2.4 Generating items by DPP

To make the generative model synthesize diverse and relevant recommendations that could fool the discriminator, employing the DPP maximum a posteriori (MAP) inference to generate the set of items with the highest DPP probability [Wu et al., 2019b; Liu et al., 2019] instead of leveraging importance sampling [Wang et al., 2017b], is an intuitive practice. However, this common approach is not adequate because of two limitations: *i*) DPP MAP inference is performed to generate plausible results for discriminator in the training procedure, but eventually provides top-scored items (*Top-N* recommendation) for specific users (in testing process or real-world recommendation scenario). This means that the corresponding recommendation model might fail to suggest diversified results for users even when the diversity is considered in the training process; *ii*) The correlations (dependencies) between observed interactions (positive items in the training dataset) and potential items (*i.e.*, items that users interacted with but are divided into the test dataset or recommended items that are of interest to users in practical application) are not considered, as during the generative model training process, MAP inference directly returns a fixed-sized subset with the highest probability, thus overlooking these dependencies.

We therefore propose a new DPP generation approach that considers conditional distribution in the testing (evaluation) process to guarantee that the reworked generator (diversity-aware GAN) that considers diversification will eventually provide users with diversified and relevant suggestions.

3.2.4.1 Generating for Training

The common practice for generating DPP-distributed diverse item subset is based on Equation 3.2 [Chen et al., 2018; Kulesza et al., 2012]. In this work, the diverse kernel on item subset is constructed by calculating the similarity of items' embedding vectors, which means that the personalized preferences of users are not taken into consideration. If we directly draw samples depending on the diverse kernel $\mathbf{L}$, the recommendation model can only provide the most diversified item set for all users. To incorporate the user's preferences and items' diversity into a unified DPP kernel, we employ the personalized kernel $\mathbf{L}_u$ for each user according to the quality *vs.* diversity decomposition of DPP kernel, formally:

$$\mathbf{L}_u = \text{Diag}(\exp(\mathcal{Q}(u))) \cdot \mathbf{L} \cdot \text{Diag}(\exp(\mathcal{Q}(u))), \tag{3.12}$$

where $\mathcal{Q}(u)$ is the relevance-evaluating function of user u on entire items. It is crucial to note that in our approach, the diverse kernel $\mathbf{L}$ is over entire items' trainable embeddings from the generator during the training process, calculated following Equation 3.8. This differs from previous methods [Chen and Ahmed, 2020; Warlop et al., 2018], which rely on a pre-learned diverse kernel. We then can perform MAP inference [Chen et al., 2018] based on the personalized DPP kernel to generate diverse and plausible item set for discriminator during the learning process.

3.2.4.2 Generating for Serving

In addition to the standard use of commonly DPP set generation within the generative model, we also draw samples for evaluation by considering conditional distribution in the MAP inference, *i.e.*, using the personalized DPP kernel to generate a diversity-quality balanced item set as final recommendations instead of directly selecting top-scored items. We partition the item ground set $\mathcal{Y}$ into two disjoint segments $\mathcal{Y}_o$ (observed positive items in training dataset) and $\mathcal{Y}_g$ (unobserved items). We regard the observed items as the previous selected subset Y_o. The conditional distribution of Y_g (generated item subset) is thus given by,

$$P\left(Y_g \mid Y_o\right) = \frac{\det \mathbf{L}_{Y_o \cup Y_g}}{\det\left(\mathbf{L} + \mathbf{I}_g\right)}, \tag{3.13}$$

where $\mathbf{I}_g$ is a diagonal matrix, the same size as $\mathbf{L}$. However, the elements corresponding to Y_o are zeros and the elements corresponding to $\mathcal{Y}_g$ are 1 in $\mathbf{I}_g$. The conditional probability is defined in such a way, the selected subset should be diverse among $\mathcal{Y}_g$ as well as be diverse from previously observed $\mathcal{Y}_o$.

The MAP inference for conditional DPP distribution is as hard as the standard DPP model [Chen et al., 2018; Gong et al., 2014], and new recommended subset can be generated following the formulation:

$$Y_g^* = \arg\max_{Y \in \mathcal{Y}_g} P\left(Y \mid Y_o\right). \tag{3.14}$$

In this way, given observed browsing histories, the final recommendations for serving users in evaluation are provided using MAP inference instead of directly selecting top-scored items.

Applying MAP inference for DPP in recommendation systems is a challenging task, especially when there is a large pool of items. The problem lies in the NP-hard nature of MAP inference itself, which can reach a complexity of $O(N^4)$ using known exact implementations of the greedy algorithm [Gillenwater et al., 2012b]. Here, N is the number of entire items. Even with the latest advanced algorithms [Chen et al., 2018; Hemmi et al., 2022] utilizing Cholesky decomposition, the complexity remains around $O\left(M^2N\right)$ (generating M elements from ground set with N items). In this scenario, directly rendering existing methods is impractical for recommendation systems. Motivated by the need for a more efficient solution, we propose a scaled-down DPP MAP inference strategy (applied in both training and serving). We first sort and select the top K items. The value of K is equal to the number of items present in the longest interaction sequence across all users in experimental dataset. Then, we use this subset with K items to form a new ground set and corresponding kernel matrix. As a result, the DPP MAP inference problem is simplified to selecting M items from a considerably smaller pool of K items. This is because the number of items that a user interacts with is typically much less than the total number of items (N) available in the system. The corresponding complexity transforms into sorting computations with a time complexity of $O(N \log N)$, coupled with scaled DPP MAP inference calculations taking $O\left(M^2K\right)$. This approach drastically reduces computational complexity, thereby enhancing efficiency and practicality for large-scale recommendation systems. Apart from these, DDPR is iteratively learned following the general adversarial training procedure.

3.3 Experiments

In this section, we present the experimental setup and provide a comprehensive analysis of the experimental results.

Table 3.2: Statistics of the datasets for experiments (MovieLens and Anime).

Dataset	#Users	#Items	#Interactions	#Categories
MovieLens	0.9k	1.6k	100k	18
Anime	73.5k	12.2k	1M	44

3.3.1 Experimental Settings

We start by introducing the settings of experiments, including datasets, evaluation metrics, baselines, and implementation configurations.

3.3.1.1 Datasets

To evaluate the effectiveness of DDPR, we conduct experiments on two benchmark datasets: *i*) **MovieLens** is a movie rating dataset that has been widely used to evaluate traditional recommendation algorithms. We use the version containing 100K ratings (1 to 5). There are 18 movie categories, and each movie may belong to more than one categories; *ii*) **Anime** dataset consists of 1 million ratings (1 to 10) from users to anime. It contains 44 explicit categories and each anime may belong to more than one categories.

The statistics of the two datasets are summarized in Table 3.2. The main reasons for selecting these two datasets are: *i*) they are common public datasets in the recommendation field. In particular, they are also chosen as experimental datasets in competitive baselines [Wu et al., 2019b; Wang et al., 2017b] analyzed in this work. Selecting them helps to keep the comparison as fair as possible; *ii*) the category coverage of recommended items is an important evaluation metric for diversity-promoting recommendation [Zhang and Hurley, 2008; Puthiya Parambath et al., 2016], and items of these two datasets contain the category attribute. For each dataset, we use the 10-core setting to ensure that each user and item have at least ten interactions. We randomly select 80% of historical interactions in both datasets to constitute the training set, and treat the remaining as the test set, as in [Wang et al., 2017b; Zhang et al., 2013].

3.3.1.2 Evaluation Metrics

MAP inference for conditional DPP is used for evaluating DDPR. Other methods directly apply top-scored selection. To comprehensively evaluate the relevance and diversity of recommended items, three groups of evaluation metrics are adopted: *i*) **Accuracy**. We evaluate the accuracy of ranking list using three top-N related metrics, *i.e.*, Precision@N (P@N), Recall@N (Re@N) and NDCG@N (ND@N); To check the relevance of suggestions in different ranking positions of recommendation list, we test N=3 and 20. *ii*) **Diversity**. Category coverage (CC) [Wu et al., 2019b; Puthiya Parambath

et al., 2016] is an widely adopted diversity evaluation metric, which is appropriate to the diversity concept in diversity-promoting recommendation. A higher value of CC means the recommendations are more diverse; *iii*) **Trade-off**. To evaluate the balance performance between accuracy and diversity, the harmonic metric F-score (F1) is often used [Cheng et al., 2017; Liu et al., 2020], where F1@N = $2 \times quality@N \times diversity@N$ / ($quality@N + diversity@N$). And $quality@N$ represents the mean of P@N, NDCG@N, and Recall@N, and $diversity@N$ denotes CC@N.

3.3.1.3 Baselines

To demonstrate the effectiveness of our DDPR model, we compare it with the following state-of-the-art methods.

- **BPRMF**: This method optimizes the Matrix Factorization (MF) model with Bayesian Personalized Ranking (pairwise ranking loss) [Rendle et al., 2012].
- **NGCF**: This is a competing GCN-based recommender model considering multi-hop neighbors based on a stack of GCN layers [Wang et al., 2019c], which has shown to outperform several types of methods including other GCN-based models GC-MC [Berg et al., 2017] and PinSage [Ying et al., 2018], and state-of-the-art neural network-based models NeuMF [He et al., 2017b] and CMN [Ebesu et al., 2018].
- **DGCN**: Through some extra designs, *e.g.*, re-balancing neighbor discovering, adjusting negative sampling, and distilling category preferences in representation space, this method [Zheng et al., 2021] pushes the diversification process upstream into the matching stage of GCN.
- **IRGAN**: This seminal GAN-based information retrieval method [Wang et al., 2017b]. improves relevance of suggestions via adversarial training in a minimax game between generator and discriminator.
- **PDGAN**: Diversity is achieved by adopting MAP inference using pre-learned DPP kernel over entire items to generate diverse results for discriminator in training procedure [Wu et al., 2019b].
- **DPPMAP**: This is a state-of-the-art method [Chen et al., 2018] using MAP inference to generate diversified recommendations.

3.3.1.4 Configuration

To keep the comparison fair, we set the trainable embedding vectors of all models in this work with the same dimension (64), and report the best results of each model by

Table 3.3: Overall performance comparison on two datasets. The bold value represent the best result among all methods *w.r.t.* a specific metric on a dataset. The improvement achieved by our approach over each baseline is denoted by the subscript.

Datasets	Metrics	BPRMF	NGCF	DGCN	IRGAN	DPPMAP	PDGAN	DDPR
MovieLens	P@3	$0.3468_{+34.08}$	$0.4414_{+5.35}$	$0.4055_{+14.67}$	$0.4570_{+1.75}$	$0.3882_{+19.78}$	$0.4080_{+13.97}$	**0.4650**
	P@20	$0.2267_{+38.73}$	$0.3124_{+0.67}$	$0.2853_{+10.23}$	$0.3119_{+0.83}$	$0.2718_{+15.71}$	$0.2784_{+12.97}$	**0.3145**
	ND@3	$0.3649_{+31.35}$	$0.4486_{+6.84}$	$0.4303_{+11.39}$	$0.4729_{+1.35}$	$0.4061_{+18.03}$	$0.4214_{+13.74}$	**0.4793**
	ND@20	$0.4364_{+14.00}$	$0.4892_{+1.70}$	$0.4560_{+9.10}$	$0.4891_{+1.72}$	$0.4590_{+8.39}$	$0.4678_{+6.35}$	**0.4975**
	Re@3	$0.0573_{+8.03}$	$0.0605_{+2.31}$	$0.0584_{+5.99}$	$0.0607_{+1.98}$	$0.0569_{+8.79}$	$0.0592_{+4.56}$	**0.0619**
	Re@20	$0.2170_{+24.10}$	$0.2681_{+0.45}$	$0.2381_{+13.10}$	$0.2676_{+0.64}$	$0.2436_{+10.55}$	$0.2457_{+9.61}$	**0.2693**
	CC@3	$0.3615_{+9.41}$	$0.3271_{+20.91}$	$0.3650_{+8.36}$	$0.3577_{+10.57}$	$0.3511_{+12.65}$	$0.3689_{+7.21}$	**0.3955**
	CC@20	$0.6665_{+5.49}$	$0.6871_{+2.33}$	$0.6968_{+0.90}$	$0.6901_{+1.88}$	$0.7019_{+0.17}$	$\mathbf{0.7097}_{-0.93}$	0.7031
	F1@3	$0.3000_{+21.00}$	$0.3219_{+12.77}$	$0.3282_{+10.60}$	$0.3434_{+5.71}$	$0.3138_{+15.68}$	$0.3286_{+10.47}$	**0.3630**
	F1@20	$0.4074_{+16.99}$	$0.4695_{+1.51}$	$0.4446_{+7.20}$	$0.4599_{+3.63}$	$0.4440_{+7.34}$	$0.4511_{+5.65}$	**0.4766**
Anime	P@3	$0.3941_{+22.74}$	$0.4293_{+12.67}$	$0.4096_{+18.09}$	$0.4670_{+3.58}$	$0.4017_{+20.41}$	$0.4560_{+6.07}$	**0.4837**
	P@20	$0.2645_{+25.33}$	$0.2988_{+10.94}$	$0.2877_{+15.22}$	$\mathbf{0.3369}_{-1.60}$	$0.2782_{+19.16}$	$0.3322_{-0.21}$	0.3315
	ND@3	$0.4062_{+19.50}$	$0.4323_{+12.28}$	$0.4174_{+16.29}$	$0.4751_{+2.19}$	$0.4129_{+17.56}$	$0.4608_{+5.34}$	**0.4854**
	ND@20	$0.4569_{+9.59}$	$0.4985_{+0.44}$	$0.4805_{+4.20}$	$0.4970_{+0.74}$	$0.4665_{+7.33}$	$0.4962_{+0.91}$	**0.5007**
	Re@3	$0.0251_{+39.84}$	$0.0331_{+6.04}$	$0.0320_{+9.69}$	$0.0340_{+3.24}$	$0.0318_{+10.38}$	$0.0332_{+5.72}$	**0.0351**
	Re@20	$0.1033_{+45.89}$	$\mathbf{0.1526}_{-1.25}$	$0.1480_{+1.82}$	$0.1503_{+0.27}$	$0.1016_{+48.33}$	$0.1471_{+2.45}$	0.1507
	CC@3	$0.2202_{+17.30}$	$0.2185_{+18.22}$	$0.2433_{+6.17}$	$0.2326_{+11.05}$	$0.2363_{+9.31}$	$0.2427_{+6.43}$	**0.2583**
	CC@20	$0.4529_{+20.87}$	$0.4529_{+20.87}$	$0.5192_{+5.43}$	$0.5130_{+6.71}$	$0.5216_{+4.95}$	$0.5326_{+2.78}$	**0.5474**
	F1@3	$0.2446_{+19.22}$	$0.2522_{+15.62}$	$0.2631_{+10.83}$	$0.2713_{+7.48}$	$0.2572_{+13.37}$	$0.2748_{+6.11}$	**0.2916**
	F1@20	$0.3421_{+19.82}$	$0.3727_{+9.98}$	$0.3846_{+6.58}$	$0.4002_{+2.42}$	$0.3662_{+11.93}$	$0.4038_{+1.51}$	**0.4099**

varying the corresponding hyperparameters. As mentioned earlier, Z negative items are selected for supervised loss in Equation 3.10. We set Z equal to the number of observed positive interactions following the common practice of negative sampling [Berg et al., 2017; Zhang et al., 2013]. Adam optimizer is used for all methods.

3.3.2 Performance Comparison

We start by comparing the performance of all the methods, and then explore the attributes of our model.

3.3.2.1 Overall Comparison

The overall comparison between our models and baselines is summarized in Table 3.3. The subscript shows the relative improvement of our model's performance over a baseline *w.r.t.* a specific metric. The best performance among all methods (each line) is in bold. We have the following main observations:

- Two relevance-oriented models, NGCF and IRGAN, outperform the other factorization-based baseline BPRMF in accuracy-related metrics (Precision, NDCG, and Recall), indicating the effectiveness of the neural graph technique and GAN methodology,

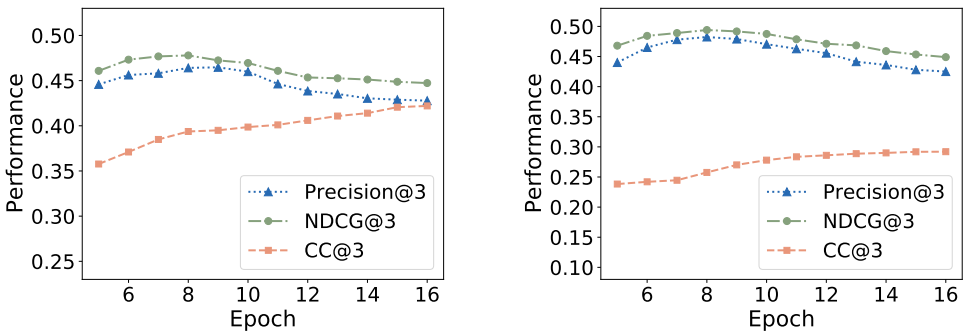
respectively. NGCF outperforms IRGAN in terms of most Top-20 related accuracy metrics (*e.g.*, NDCG@20 on MovieLens and Recall@20 on Anime), which suggests that exploiting the high-order connectivity from user-item interactions helps to find more latent preferences of users. As for other accuracy and diversity metrics, IRGAN shows advantages over NGCF. This might be attributed to the adversarial training, which improves the ability of IRGAN to provide accurate sets that are also diversified to some extent by the importance sampling of the generator.

- Compared to competitive relevance-oriented recommendation models (IRGAN and NGCF), three state-of-the-art diversity-promoting methods (DPPMAP, PDGAN, DGCN) achieve higher performance in diversity (CC), but are worse in accuracy results. This demonstrates that they all end up with sacrificing accuracy for diversity (trade-off), which is also the common strategy that most diversity-promoting recommendation models adopt [Chen et al., 2018; Chen and Ahmed, 2020; Ashkan et al., 2015]. On the contrary, our DDPR method demonstrates varying degrees of improvement in both accuracy and diversity compared to relevance-oriented and diversity-promoting approaches. This indicates that our method has the capability to simultaneously consider both accuracy and diversity, rather than making a trade-off between the two.

- Comparing the balance performance (F1) of all methods, our DDPR achieves significant improvements (>5% in most cases) against baselines. This validates that the formulated two optimization functions incorporated into the GAN framework facilitate meeting the two primary objectives of diversity-promoting recommendation.

- In comparison to the competitive diversity-promoting recommendation models (*e.g.*, PDGAN and DGCN), DDPR achieves superior diversity performance with a substantial margin (around 5%), except for the CC@20 on MovieLens. This demonstrates the efficacy of our dual consideration approach as it ensures the production of diverse predictions to the greatest possible extent, offering a variety of options to cater to users' diverse interests. Simultaneously, our model also displays a significant advantage in terms of accuracy metrics compared to the noteworthy relevance-oriented models (*e.g.*, NGCF and IRGAN), indicating that the achievements of diversity do not compromise the recommendation quality. This can be primarily attributed to two factors: *i*): our two-objective optimization approach adeptly balances the learning of item embeddings; *ii*) this balanced learning allows the personalized DPP kernel to generate both diverse and relevant results.

As shown in Equation 3.10, the loss function of minimizing distance between true labels and predictions is inspired by the squared loss. We also tried the commonly



(a) MovieLens (b) Anime

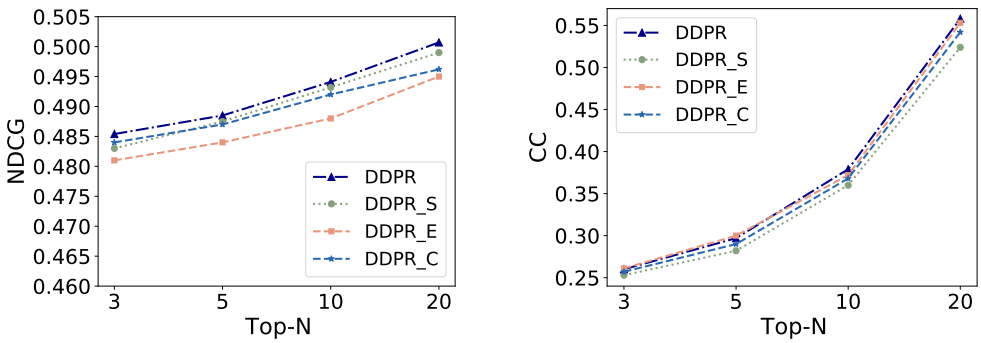
Figure 3.1: The learning curves of DDPR *w.r.t.* quality (Precision@3 and NDCG@3) and diversity (CC@3) performances, (a) on MovieLens and (b) on Anime

used binary cross-entropy [He et al., 2017b] as the supervised loss, which produces slightly weaker performance in experiments. The reason might be that the cross-entropy cost is applied to the minimax game between generator and discriminator, and using it again for supervision brings difficulty for the generator to find extra insights in training. Therefore, all the experimental results of our models are obtained using the supervised loss as in Equation 3.10. Unlike previous trade-off diversity-promoting recommendation studies [Wu et al., 2019b; Chen et al., 2018; Liu et al., 2020], there is no experiment focusing on studying the effects of trade-off parameter tuning in this work, as we directly use two optimization objectives to boost both diversity and quality rather than trading off one goal against another.

3.3.2.2 Learning Trends of Relevance and Diversity

To demonstrate the correlations between relevance and diversity in the adversarial training process, we illustrate the performance comparison (Precision@3, NDCG@3, and CC@3) over training epochs on Movielens and Anime datasets in Figure 3.1. In the earlier training period, the performance usually increases gradually (similar to the other machine learning models). To better indicate the trends of our model, the learning results of early stages (epochs) are omitted.

A common observation on two datasets is that through adversarial training, the diversity performance increases monotonously as training epochs grow, while the relevance increases at first and then decreases. As we directly maximize diversity and minimize relevance loss, with the learning of generator and discriminator, both relevance and diversity are improved. After the relevance performance reaches its full capacity (*i.e.*, samples from generator can fool the discriminator), it starts to degenerate. At this stage, the diversity continues to grow due to the maximization of diversity.



(a) NDCG (b) CC

Figure 3.2: NDCG and CC performance with different N on Anime. DDPR-S, DDPR-E, and DDPR-C represent ablation variants of DDPR that ablate diversity consideration in serving, recommendation error minimization loss, and conditional dependency in MAP inference of serving process, respectively.

Because of the supervision loss in generator, the degeneration of relevance performance in the later stages of training process is relatively gradual (as shown in Figure 3.1 after around 10 epochs), which means that if the diversity factor is preferred by a specific recommender system, we can train DDPR with more epochs to achieve higher diversity and acceptable relevance. The results listed in Table 3.3 are selected according to the NDCG@3 performance, *i.e.*, selecting the results of an epoch that achieves the best NDCG@3 result.

3.3.2.3 Ablation Analysis

We perform ablation studies to show how the techniques of DDPR affect its performance. In Figure 3.2, -S, -E, and -C represent DDPR variants that are deprived of diversity consideration in serving (directly employing top-N selection in evaluation), recommendation error minimization loss (without using Equation 3.10), and conditional dependency in MAP inference (using standard DPP generation in serving rather than adopting Equation 3.2), respectively. The corresponding experiments are conducted on MovieLens. Aligning with the two subfigures, we can see that:

- Our MAP inference on conditional DPP distribution takes a step further compared to previous common DPP generation methods. Specifically, it incorporates known user interactions into the inference process, hence providing a more personalized and context-aware recommendation (as shown in the comparison between DDPR and DDPR-C). This approach not only makes our recommendations more relevant to individual users, but also enhances the diversity of the recommendations;

- Although DDPR-E slightly outperforms DDPR in terms of diversity, when N is relatively small (*e.g.*, CC@3) in the Top-N evaluations. However, across a broader array of metrics, our model demonstrates significant advantages. This indicates that incorporating the recommendation error loss to fulfill the two-objective problem formulation of diversity-promoting recommendation plays a key role in creating a balance between diversity and accuracy. Specifically, by integrating this additional loss, our model not only seeks to maximize diversity among the recommended items but also strives to minimize the discrepancy between the predicted and true user preferences. This dual-goal orientation ensures that our model can generate diverse recommendations without sacrificing the quality and relevance of the recommendations, thereby enhancing the overall user satisfaction;

- Taking diversity into account during both the training and evaluation phases indeed improves the accuracy and diversity of the recommendation results, (DDPR *vs.* DDPR-S). Specifically, by maintaining a focus on diversity throughout the entire model lifecycle, from the learning phase to the application stage, our approach ensures a consistent emphasis on broadening the scope of recommendations. These findings underscore the significant contributions of our DDPR framework in effectively promoting diversity and maintaining accuracy in recommendation systems.

In examining DDPR's performance with respect to training epochs, we gain a clear understanding of how the model evolves and improves over training time. This offers insights into the learning process of our model, providing us valuable cues for fine-tuning and model optimization. Through the ablation analysis, we are able to identify the specific impact of each component of DDPR on the overall performance. This process not only underscores the contributions of individual model elements but also offers a path to potential enhancements by highlighting the areas that could benefit from further refinement.

3.4 Conclusion

This chapter presents a fresh perspective on diversity-promoting recommendation, moving away from the traditional focus on the trade-off between quality and diversity. Experimental results conducted on real-world datasets demonstrate that the direct formulation of the inherent goals of diversity-promoting recommendation using two optimization objectives can significantly enhance both the diversity and relevance of recommendations. To fully harness the potential of diversity-aware item embeddings, we introduce conditional DPP MAP inference during the testing/serving procedure. This approach takes into consideration the correlations and dependencies between ob-

served interactions and potential user interests, leading to further improvements in recommendation quality.

3.4.1 Limitations

In this chapter, the proposed model presents limitations primarily in two aspects. First, the proposed diversity maximization method can be considered a generic approach applicable to diversification in various domains, including Web search [Rafiei et al., 2010] and machine translation [Roberts et al., 2020]. However, within the GAN framework, the measure of relevance, as indicated in Equation 3.5, relies on the dot product between user and item embeddings. This operational approach essentially underlies traditional recommendation through matrix factorization (MF). Additionally, our model's implementation of the DPP kernel in the context of DPP MAP generation is also defined using Equation 3.5, emphasizing that our model is confined to traditional recommendation tasks at its core. Second, due to the inherent nature of GANs, involving the adversarial training of two models and the item generation module of the generator model, the training efficiency of the model is constrained.

3.4.2 Future Work

Addressing the two limitations mentioned above, we envision two future directions. First, we plan to design a unique GAN framework tailored for recommendation tasks, aiming to enhance the speed of adversarial training and the item list provision by the generator. Second, we intend to introduce more generalized quality measurement metrics and calculation methods to elevate the practicality of the DDPR model.

Chapter 4

Determinantal Point Processes Likelihoods for Sequential Recommendation

In the previous chapter, we introduced the effort to exploit MAP inference for generating DPP subsets, using these as diversified recommendations. This is an intuitive attempt to apply DPP for producing personalized and diversified predictions in the context of traditional recommendation. Although improvements are achieved by the DPP-based training and serving settings deployed within GAN, the diversity factor in DDPR is considered by incorporating additional components into the existing recommendation framework. This means that the DPP-based designs are an auxiliary operation instead of the essential part for recommendation models. We therefore propose to explore more significant roles that DPP can play in this chapter, such as the objective functions that steer the learning of recommendation models. Considering that dependency is an important factor to be modeled in sequential recommendation, and given that DPP distributions formulate the probability of subsets (thereby capturing the dependency among elements within subsets), we are motivated to propose DPP-based objective functions for sequential recommendation.

Sequential recommendation is a popular task in academic research and close to real-world application scenarios, where the goal is to predict the next action(s) of the user based on his/her previous sequence of actions. In the training process of recommender systems, the loss function plays an essential role in guiding the optimization of recommendation models to generate accurate suggestions for users. However, most existing sequential recommendation techniques focus on designing algorithms or neural network architectures, and few efforts have been made to tailor loss functions that fit naturally into the practical application scenario of sequential recommender systems.

Ranking-based losses, such as cross-entropy and Bayesian Personalized Ranking (BPR), are widely used in the sequential recommendation area. We argue that such

objective functions suffer from two inherent drawbacks: *i*) the dependencies among elements of a sequence are overlooked in their loss formulations; *ii*) instead of balancing accuracy (quality) and diversity, only generating accurate results has been over emphasized. We therefore propose two new loss functions based on the Determinantal Point Process (DPP) likelihood, that can be adaptively applied to estimate the subsequent item or items. The DPP-distributed item set captures natural dependencies among temporal actions, and a quality *vs.* diversity decomposition of the DPP kernel pushes us to go beyond accuracy-oriented loss functions. Experimental results using the proposed loss functions on three real-world datasets show marked improvements over state-of-the-art sequential recommendation methods *w.r.t.* both accuracy and diversity metrics.

4.1 Introduction

Traditional recommender systems, *e.g.*, *Top-N* recommendation [Hu et al., 2008; Deshpande and Karypis, 2004], strive to model users' preferences towards items based on historical interactions between users and items. The modeling process only depends on static interactions and ignores sequential dependencies with the assumption that all user-item interactions are equally important. However, this might not hold in real-world scenarios [Fang et al., 2020], where the next action(s) of a user are generated according to his/her previous behavioral sequence. To model users' evolving interests and adjust recommendation methods to real applications, the customized time-relevant recommender system (*i.e.*, sequential recommendation) is proposed and has become a popular topic in academic research and online applications.

Sequential dependencies among items are relevant for next item(s) prediction. For example, if it is known that a user has browsed "*shirt*, *hat* and *watch*" (as shown in Figure 4.1), the conventional recommender typically provides content-similar or category-similar items [Liang and Qian, 2021], such as more watches or clothes. As for sequential recommendation, considering dependencies and the temporal order of a previous behavioral sequence, the recommendation model tends to capture the temporal dependency of previous actions on target items (*i.e.*, *sequence dependence*), and thereby determines that the user is actually trying to find clothing accessories (*e.g.*, *headphones* and *glasses*) in this period of time.

The core consideration in recommending an item or items is the user preference-based ranking of items. Similar to most learning-to-rank tasks (*e.g.*, Web search and question answering), commonly used ranking-based loss functions such as pointwise ranking (cross-entropy) and pairwise ranking (BPR) are widely deployed for the training of recommendation models. In this chapter, we argue the insufficiency of applying

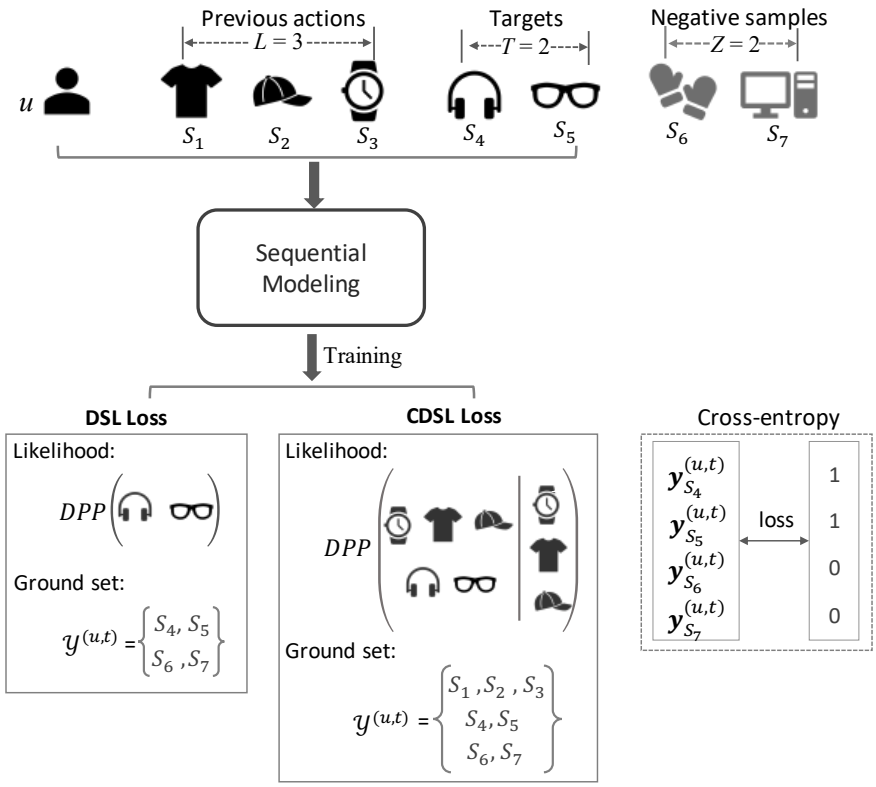


Figure 4.1: A simplified diagram comparing the loss functions based on DPP-distributed set likelihoods with a basic cross entropy loss function.

ranking-based loss functions in sequential recommendation from three aspects.

Firstly, we regard estimating the next T items (*i.e.*, target items in Figure 4.1) as a task of predicting the temporally subsequent set, according to the historical sequence. In this scenario, the dependency among targets is noteworthy. However, the ranking-based objective function independently compares each target to its true label (pointwise ranking), to a negative item (pairwise ranking), or to a perfect ordering list (listwise ranking). This manipulation neglects the dependency among targets (*targets dependency*), thereby affecting the set prediction capacity of sequential recommendations. For example, based on the previous sequence in Figure 4.1, the loss calculated by assuming independent targets (*headphones* or *glasses*) merely enables sequential models to capture the dependencies in two sequences {*shirt*, *hat*, *watch*, *headphones*} or { *shirt*, *hat*, *watch*, *glasses*}. This means that part of the *sequence dependency* is neglected, due to the omission of *targets dependency*. On the other hand, the inherent goal of next-item recommender systems is to select or generate personalized item sets for specific users. However, ranking-based loss functions enforce the rank or estimated score of targets, leading sequential recommendations to provide high-scored items (*i.e.*, *top-N* prediction), without considering the feasibility of selecting whole

targets as a generated set. In this work, instead of stressing the ranking of target items in loss functions, we propose to focus on enhancing the likelihood of target sets, *i.e.*, endowing the target set with a high probability of being selected/recommended.

Based on the Determinantal Point Process (DPP) [Kulesza et al., 2012; Kulesza and Taskar, 2011a], we propose the DPP Set Likelihood-based loss function (DSL). We regard the temporal prediction task as drawing a subset from a user-sequence specified ground set (denoted as $\mathcal{Y}^{(u,t)}$ in Figure 4.1) according to the DPP probability measure. $\mathcal{Y}^{(u,t)}$ is the combination of target items and negative samples for each user's sequence at time step t. It means that the drawn subset represents the probability of inclusion for user-preferred targets and exclusion for non-relevant items. In this way, the set likelihood-based loss function considers the correlations among observed targets, and enforces that the chance of suggesting the target set should be higher than that of selecting the negative one.

Secondly, existing sequential recommendations that try to capture the *sequence dependency* (on the entire sequence) *e.g.*, {*shirt*, *hat*, *watch*, *headphones*, *glasses*} in Figure 4.1, mainly focus on modeling the dependency in neural layers [Tang and Wang, 2018; Kang and McAuley, 2018] or Markov chains [Rendle et al., 2010; He and McAuley, 2016a]. However, in a more challenging part, *i.e.*, the training loss, the dependencies among temporal actions are neglected. While DSL takes into account the *targets dependency* when calculating loss, it does not directly model the relationship between previous actions and the targets. Specifically, DSL is proposed for the scenario of predicting the next items (*i.e.*, $T > 1$ in Figure 4.1).

To model the entire dependency of a sequence and generalize DPP likelihood-based loss to next item(s) estimation (*i.e.*, predicting next one or more targets in training), a new Conditional DPP Set Likelihood (CDSL) objective is designed. We condition a DPP on the event that elements (items) in the previous set are observed. Specifically, given the previous actions, the target item(s) subset is selected (distributed as a DPP) and negative samples are excluded. In this case, the user-sequence specified ground set contains items of the entire sequence ($L + T$) and corresponding negative samples Z. Conditioning a DPP on observed previous items helps recommendation models capture more correlations and is more powerful for subset drawing. Experiments on different datasets demonstrated these intuitions.

Thirdly, ranking-based losses mainly aim at improving the relevance score of target items, whereas they account little for the diversity that is an important factor for avoiding users being bored with the narrow scope of topics of recommendations. This causes recommender systems to unduly lean on accuracy recommendations [Wu et al., 2019b] while ignoring the diversity.

The Determinantal Point Process (DPP), as an elegant probabilistic model, is en-

Table 4.1: The Key Symbols Used in DDPR Framework.

Notation	Interpretation
$\mathbf{K} \in \mathbb{R}^{\lvert\mathcal{Y}\rvert \times \lvert\mathcal{Y}\rvert}$	diverse kernel, $\mathcal{Y}$ denotes ground set
$\mathcal{Y}^{(u,t)}$	sequence specified ground set
$T^{(+)}$, $T^{(-)}$	observed diverse set, unobserved item set
$\mathbf{K}^{(u,t)} \in \mathbb{R}^{\lvert\mathcal{Y}^{(u,t)}\rvert \times \lvert\mathcal{Y}^{(u,t)}\rvert}$	submatrix of diverse kernel
$\mathbf{L}^{(u,t)} \in \mathbb{R}^{\lvert\mathcal{Y}^{(u,t)}\rvert \times \lvert\mathcal{Y}^{(u,t)}\rvert}$	submatrix of DPP kernel
$\mathbf{R}^{(u,t)}$	user u's preference scores on specific items

dowed with the property of modeling the repulsive correlations (dissimilarity) of elements in a subset. This property makes DPP a natural choice for diversity-promoting recommender systems, as it facilitates providing items that are diversified. Unlike previous studies that apply DPP for diversification in the process of building model architectures [Gartrell et al., 2019; Warlop et al., 2019; Wilhelm et al., 2018] or generating diverse and relevant items by maximum a posteriori (MAP) inference [Wu et al., 2019b; Chen et al., 2018], we embed the recommendation diversity in the set likelihood-based loss functions by employing the quality (accuracy or relevance) *vs.* diversity decomposition of DPP kernel.

We summarize the contributions of this chapter as follows:

- A new perspective (*i.e.*, set selection/prediction instead of ranking items) on loss functions for sequential recommendations is contributed.
- Two objective functions, DSL and CDSL bring *targets dependency* and *sequence dependency* into training loss and fit naturally into the next items and item(s) prediction.
- The diverse kernel $\mathbf{K}$ learned from the observed item sets and personalized relevance scores predicted by the sequential model play roles of diversity and quality in the diversity-aware loss functions, respectively.

4.2 Proposed Loss Functions

The simplified illustration of DPP set likelihood-based loss functions (DSL and CDSL) is shown in Figure 4.1. The most apparent difference between the commonly used pointwise ranking loss (binary cross entropy) and the proposed loss functions (DSL and CDSL) is that the cross entropy is a type of accuracy comparison loss [Ruby and Yendapalli, 2020], *i.e.*, measuring the probability error between estimated scores and true labels, whereas DSL and CDSL calculate the loss measure by DPP set distribution

comparison, *i.e.*, the comparison between inclusion probability for targets and exclusion probability for negative samples. In addition, the proposed DSL and CDSL directly apply the output results of sequential models to calculate the proposed losses and no specific changes to the model framework are required. This indicates that DSL and CDSL can be flexibly employed in different sequential models. We summarize notations used in this chapter in Table 4.1.

4.2.1 Preliminaries: Common Loss Functions and DPP

Before we present the proposed loss functions, we briefly review the preliminaries of ranking-based losses. We reintroduce DPPs that have been mentioned in Section 2.4 and Section 3.2.1, in order to provide a cohesive explanation of our motivation.

4.2.1.1 Ranking-based losses

Similar to other learning to rank tasks, the core of recommender systems is the relevance-based ranking of items[Hidasi et al., 2015]. Ranking can be pointwise, pairwise, or listwise.

There are different types of pointwise-based losses, such as cross-entropy, squared loss, and MRR optimization [Steck, 2015]. In this section, we only present the binary cross-entropy loss, as different pointwise-based losses share the similar concept and cross-entropy is the most common and representative one [He et al., 2017b].

The binary cross-entropy loss for next items (T targets) prediction is defined as

$$\ell = \sum_{u} \sum_{t \in C^u} \sum_{i \in \mathcal{D}_t^u} -\log\left(\sigma\left(\boldsymbol{r}_i^{(u,t)}\right)\right) + \sum_{j \notin \mathcal{D}_t^u} -\log\left(1 - \sigma\left(\boldsymbol{r}_j^{(u,t)}\right)\right), \tag{4.1}$$

where $\boldsymbol{r}_i^{(u,t)}$ is the estimated preference score of user u on target item i in the sequence of time step t, and σ usually represents the *sigmoid* activation function. C^u is a collection of specific time steps of user u. To make it applicable to next items prediction, $\mathcal{D}_t^u$ is used to denote the next T target items. The cross-entropy format can be derived from the likelihood function

$$p(\mathcal{S} \mid \Theta) = \prod_{u} \prod_{t \in \mathcal{C}^u} \prod_{i \in \mathcal{D}_t^u} \sigma\left(\boldsymbol{r}_i^{(u,t)}\right) \prod_{j \notin \mathcal{D}_t^u} \left(1 - \sigma\left(\boldsymbol{r}_j^{(u,t)}\right)\right) \tag{4.2}$$

by taking the negative logarithm. From this equation we see that the cross-entropy loss independently estimates the ranking or score, ignoring dependency among target items.

Pairwise ranking (BPR) [Rendle et al., 2012] is also widely used in learning-to-rank

tasks, formulated as:

$$\ell = \sum_{u} \sum_{t \in C^u} \sum_{i \in \mathcal{D}_t^u} - \ln \sigma \left(\boldsymbol{r}^t_{(u,i,j)} \right), \tag{4.3}$$

where $\boldsymbol{r}^t_{(u,i,j)} = \boldsymbol{r}^t_{(u,i)} - \boldsymbol{r}^t_{(u,j)}$. Item j is an item that user u shows no interest. This formulation is derived from the maximum a posteriori (MAP) estimator of the model parameters (Θ):

$$\underset{\Theta}{\operatorname{argmax}} \prod_{u} \prod_{t \in C^u} \prod_{i \in \mathcal{D}_t^u} p\left(i >_{u,t} j \mid \Theta\right) p(\Theta), \tag{4.4}$$

based on the Bayesian formulation and the independence assumption of targets and users. This means that pairwise ranking loss also takes no account of targets dependency and sequence dependency. The above formulations are defined in the next T targets scenario, which can be easily transformed for the next item task.

Instead of treating either each rating (cross-entropy) or each pairwise comparison (BPR) as an independent instance, listwise-based ranking [Cao et al., 2007] directly assigns a loss to an entire item list (or a *Top-N* list) by employing the cross-entropy to measure the distance between a predicted list and perfect ordering. The listwise learns a ranking dependent on the position of items in the relevance score-ordered lists. However, as interactions in recommender systems are usually of binary relevance (with no perfect ordering list) and the permutation probabilities are computationally expensive, listwise ranking is not used often [Hidasi et al., 2015] in sequential recommendations. Therefore, listwise ranking is not compared with here.

4.2.1.2 Determinantal Point Process

In this section, we revisit and provide a succinct explanation of Determinantal Point Processes (DPPs) to facilitate further understanding and familiarity with DPPs. DPP is known and widely used within the machine learning community, *e.g.*, diversity-promoting recommendations [Wu et al., 2019b] and neural conversation [Song et al., 2018]. It provides tractable and efficient means to balance quality and diversity within a subset. Formally, a determinantal point process $\mathcal{P}$ on a discrete set $\mathcal{Y} = \{1, 2, ..., M\}$ is a probability measure on $2^{\mathcal{Y}}$, the set of all subsets of $\mathcal{Y}$). When $\mathcal{P}$ gives nonzero probability to the empty set, there exists a matrix $\mathbf{L} \in \mathbb{R}^{M \times M}$, such that for every subset $Y \subseteq \mathcal{Y}$, the probability is $\mathcal{P}(Y) \propto \det\left(\mathbf{L}_Y\right)$, where $\mathbf{L}$ is a real, positive semi-definite kernel matrix indexed by the elements of $\mathcal{Y}$. In the context of sequential recommendation, $\mathcal{Y}$ is the entire item (movies or products) catalog, and Y is the subset of items that users interact with. The notation $\mathbf{L}_Y$ denotes the submatrix of kernel $\mathbf{L}$ restricted to only those rows and columns which are indexed by Y.

The marginal probability of including one element S_i is $\mathcal{P}(S_i) = \mathbf{L}_{ii}$. That is, the diagonal of $\mathbf{L}$ gives the marginal probabilities of inclusion for individual elements of $\mathcal{Y}$.

The probability of selecting two items S_i and S_j is $\mathbf{L}_{ii}\mathbf{L}_{jj} - \mathbf{L}_{ij}^2 = \mathcal{P}(S_i)\mathcal{P}(S_j) - \mathbf{L}_{ij}^2$. The entries of the kernel $\mathbf{L}$ usually measures similarity between pairs of elements in $\mathcal{Y}$. Thus, highly similar elements are unlikely to appear together. In this way, repulsive correlations (*i.e.*, diversity) with respect to a similarity measure are captured. However, an important factor — personalized relevance to items in recommendations are not reflected in the similarity-based kernel matrix. To adapt DPP to user preferences- and diversity-aware sequential recommendation tasks, we bring a quality *vs.* diversity decomposition of the DPP kernel by independently modeling diversity and quality.

4.2.2 Quality *vs.* Diversity Kernel

Similarity measures reflect the primary qualitative characterization of the DPP as diversifying processes. However, in practical recommendations, diversity needs to be balanced against underlying user preferences (quality) for different items. The decomposition of the DPP kernel $\mathbf{L}$ offers an intuitive trade-off between quality of items in $\mathcal{Y}$ and a global model of diversity, whose entries can be written $\mathbf{L}_{ij} = q_i \phi_i^\top \phi_j q_j$. $q_i \in \mathbb{R}^+$ is a quality term of item i and $\phi_i \in \mathbb{R}^D$ is a D-dimensional vector of normalized diversity features (in practice, it is usually represented by feature vectors) [Kulesza et al., 2012]. In sequential recommendations, user preferences on items of a sequence can be naturally treated as the quality measure. On the other hand, we need a diversity kernel to capture the diversity of items, which can be learned from the observed diverse item sets [Gartrell et al., 2017]. To reduce the computational complexity of calculating a $\mid \mathcal{Y} \mid \times \mid \mathcal{Y} \mid$ matrix, the diversity kernel is represented using the low-rank factorization as $\mathbf{K} = \mathbf{V}^\top \mathbf{V}$, where $\mathbf{K} \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$, $\mathbf{V}$ is a $M \times D$ matrix and $D << \mid \mathcal{Y} \mid$. From now on we use $\mathbf{K}$ to denote the *diversity kernel*, and $\mathbf{L}$ represents the final kernel that balance by quality and diversity.

We first learn *diversity kernel* $\mathbf{K}$ from observed interactions (instances in training datasets) before training sequential models. To capture the co-occurrence of diverse items, we use the following diverse sets generation process to diversify item sets: *i)* before sequentially selecting items for a diverse set, we assign each item in a user's interaction list with the same probability of being selected; *iii)* every time an item i is selected. Its category c_i will be stored. The probability of all items in c_i will decay ($decay = 0.5$). Because items are randomly selected according to their corresponding probabilities, the decay setting facilitates generating item sets with different categories (*i.e.*, diversity). Five distinct items are selected for each subset; *iii)* repeat the last step until the observable positive items of the user have all been selected at least once.

For each positive item set, a corresponding negative set is sampled, which contains negative items that share the same categories with items of the positive set, but have

no interaction with the user.

Based on these ground-truth diverse sets, the diversity kernel $\mathbf{K}$ can be learned following [Warlop et al., 2019], with the log-likelihood formulation:

$$\ell = \sum_{\left(T^{(+)},T^{(-)}\right)\in\mathcal{T}} \log\det\left(\mathbf{K}_{T^{(+)}}\right) - \log\det\left(\mathbf{K}_{T^{(-)}}\right), \tag{4.5}$$

where $T^{(+)}$ is an observed diverse set and $T^{(-)}$ represents the corresponding set that contains negative items and $\mathcal{T}$ denotes the set of paired sets used for training.

In sequential recommendation, the training loss is calculated by accumulating the difference comparison between targets and predictions from each sequence of users in a batch. To compare the likelihood between the true set and negative set in each sequence, we define the following user-sequence specified kernel:

$$\mathbf{L}^{(u,t)} = \mathrm{Diag}\left(\mathbf{R}^{(u,t)}\right)\cdot\mathbf{K}^{(u,t)}\cdot\mathrm{Diag}\left(\mathbf{R}^{(u,t)}\right), \tag{4.6}$$

where $\mathbf{R}^{(u,t)}$ represents user u's preferences on specific items (*sequence ground set*, *i.e.*, $\mathcal{Y}^{(u,t)}$ in Figure 4.1) depending on the observed sequence of time step t, and $\mathbf{K}^{(u,t)} \in \mathbb{R}^{l\times l}$ (l denotes the number of elements in the sequence ground set, which can be $L+T$ or $L+T+Z$ as shown in Figure 4.1) is the submatrix of diversity kernel indexed by the *sequence ground set*. Similarly, $\mathbf{L}^{(u,t)} \in \mathbb{R}^{l\times l}$ is used to denote the submatrix of $\mathbf{L}$ DPP kernel. We name it as *sequence kernel*, as it is confined to a user sequence. As a shorthand, we will remove the superscript (u,t) that represents a specific ground set associated to a user sequence when the meaning is clear.

4.2.3 DPP Set Likelihood-based Loss

We first introduce the intuitive idea of using DPP set likelihood comparison as a loss function for sequential recommendations. Given a sequence, the targets and negative items can be regarded as the user-sequence specified ground set, and we can treat the measurement of loss as the negative log probability of selecting targets as a DPP. This means that the target set is supposed to receive a higher probability of being drawn by $\mathcal{P}$ (mentioned in Section 4.2.1.2) than that of negative items. The normalization constant for $\mathcal{P}$ follows from the observation that $\sum_{Y'\subseteq\mathcal{Y}}\det\left(\mathbf{L}_{Y'}\right) = \det(\mathbf{L}+\mathbf{I})$, where $\mathbf{I}$ is the identity matrix [Gartrell et al., 2017]. The probability of $\det\left(\mathbf{L}_Y\right)$ is normalized by all possible item sets $Y'\subseteq\mathcal{Y}$. Therefore, we have the following probability of inclusion for target set Y_T in a user sequence:

$$\mathcal{P}(Y_T) = \frac{\det\left(\mathbf{L}_{Y_T}\right)}{\det(\mathbf{L}+\mathbf{I})}, \tag{4.7}$$

The ground set of this sequence is $\mathcal{Y}^{(u,t)}$ as shown in the DSL part of Figure 4.1. Y_T denotes target items set in a sequence. More precisely, $\mathbf{I}$ is identity matrix of the *sequence ground set*, *i.e.*, with ones in the diagonal positions. It represents a $\mid \mathcal{Y}^{(u,t)} \mid \times \mid \mathcal{Y}^{(u,t)} \mid$ matrix. In DSL, *sequence ground set* contains targets and negative samples in a user's sequence instance.

Based on the introduction above, we can see that DSL is expected to capture the dependency among targets. In this work, DSL is only applied to the next items (target items $T > 1$) estimation. This is to the fact that if only one item is in the target set, the significance of considering *targets dependency* is lost. Besides, DSL ignores the *sequence dependency* (correlations between previous items and targets). To make the DPP-based loss function more viable, we further propose CDSL.

4.2.4 Conditional DPP Set Likelihood-based Loss

To capture entire *sequence dependency* in loss functions, we try to formulate the motivation of drawing a DPP set (T targets) conditioning on the observed previous set (L items). Given the previous items (Y_L) in a user-specified sequence, we want to obtain the probability of drawing an entire sequence ($Y_{L \cup T}$), *i.e.*, L previous items and T targets, which is formulated as:

$$\mathcal{P}(Y_{L \cup T} \mid Y_L) = \frac{\det\left(\mathbf{L}_{Y_{L \cup T}}\right)}{\det\left(\mathbf{L} + \mathbf{I}_{\overline{Y}_L}\right)}, \tag{4.8}$$

where $\mathbf{L}$ is actually the *sequence kernel* on $\mathcal{Y}^{(u,t)}$ that is composed of previous items, targets and negative samples (as shown in the CDSL part of Figure 4.1). $\mathbf{I}_{\overline{Y}_L}$ is the matrix with ones in the diagonal entries indexed by elements of $\mathcal{Y}^{(u,t)} - \overline{Y}_L$ and zeros everywhere else. This conditional distribution is derived according to [Borodin and Rains, 2005; Kulesza et al., 2012]. If readers are interested, you can learn more from the literature. From this equation, we can see that even if T equals 1 (next item scenario), CDSL can still be applied to utilize the dependency by calculating DPP set ($Y_{L \cup T}$) probability. This means that CDSL is more adaptable.

Based on Equation 4.7 and Equation 4.8, DSL and CDSL loss functions for next T targets and next target(s) estimations can be obtained by taking the negative log-likelihood.

Table 4.2: Statistics of the datasets (ML-1M, Anime, and Beauty).

Dataset	#Users	#Items	#Interactions	#Categories
ML-1M	6.0k	3.4k	1.0M	18
Anime	73.5k	12.2k	1.0M	43
Beauty	52.0k	57.2k	0.4M	213

4.3 Experiments

We comprehensively evaluate the proposed DSL and CDSL loss functions in terms of *i*) recommendation quality, *ii*) diversification, and *iii*) training efficiency.[1]

4.3.1 Datasets

We evaluate our methods on three benchmark datasets from three real world applications. **MovieLens**[2] is a widely used benchmark movie rating dataset. We use the version (*ML-1M*) that includes one million user ratings to 18 categories of movies. **Amazon** contains a series of datasets introduced in [McAuley et al., 2015], comprised of a large corpora of product reviews crawled from *Amazon.com*. We consider *Beauty* category in this work, which is notable for its high sparsity and variability. There are 213 categories of *Beauty* products. **Anime**[3] consists of user ratings to anime in *myanimelist.net*. All items in *Anime* are grouped into 43 categories. The statistics of the datasets are shown in Table 4.2. The reasons we select these three datasets are: *i*) they are common public datasets in the sequential recommendation field. In particular, *Beauty* and *ML-1M* are also chosen as experimental datasets in two state-of-the-art baselines [Tang and Wang, 2018; Kang and McAuley, 2018] analyzed in this work. Selecting them helps to keep the comparison as fair as possible; *ii*) the diversity (category coverage of recommended items) is also considered in the proposed loss functions. To evaluate the diversity metric, we need to choose datasets that contain the category attribute of items. As the category attribute is not considered in the selected baselines, we process the datasets by closely following their settings, instead of directly requiring the processed datasets from the authors. For all datasets, we use timestamps to determine the sequence order of actions. We use the 10-core setting for experiments to ensure the quality of the dataset.

Referring to previous work [Kang and McAuley, 2018; He et al., 2017a; Rendle et al., 2010], the most recent T actions (last T items) in each user's sequence are used for testing. The remaining actions (except the T items) are split into two parts: *i)* first 90% of them as the training set and *ii)* the next 10% of actions as the validation

[1] Our source code can be found at https://github.com/l-lyl/DPPLikelihoods4SeqRec.
[2] https://grouplens.org/datasets/movielens/1m/
[3] https://www.kaggle.com/CooperUnion/anime-recommendations-database

set. To better imitate the real-world sequential recommendation scenario and evaluate DSL and CDSL more comprehensively, the test set size of each dataset is not fixed but changes with the number of targets (*i.e.*, T in Figure 4.1) in training sequences. For instance, when considering the single next target ($T = 1$) in training, we only leave the last item of each user's interactions for testing, and the remaining actions are used for training and validating. Similarly, when $T = 3$, the test set contains each user's last three items.

4.3.2 Baselines

As mentioned in Section 4.2, the proposed loss functions can be flexibly deployed in various sequential recommendation methods without modifying their model architecture. Under the same experimental settings, if the performance of reworked models (replacing a baseline's original loss function with DSL or CDSL) is improved, the superiority of our loss functions is demonstrated. In this work, three baselines are taken into consideration:

Caser [Tang and Wang, 2018]: A CNN-based method that applies convolutional operations on the embedding matrix of some most recent items, achieves the *state-of-the-art* sequential recommendation performance. The *binary cross-entropy loss* is used for one or more targets.

MARank [Yu et al., 2019]: A multi-order attentive neural model to extend user's general preference by fusing individual- and union-level temporal item-item correlations is designed for next-item recommendation. BPR loss for one target is adopted.

SASRec [Kang and McAuley, 2018]: This is a *state-of-the-art* self-attention based next-item recommendation model, which adaptively assigns weights to previous items at each time step. The *binary cross-entropy loss* is calculated for each fixed-length sequence. If the length of a user's sequence is less than the maximum length n, *'padding'* items will be added. This means that the targets and previous items in a fixed-length sequence are both considered when calculating loss.

The above description provides three reasons for the selection of these baselines: *i*) they are recently proposed state-of-the-art sequential recommendation methods; *ii*) they are representative works, and many new models are built based on them [Sun et al., 2019; Xu et al., 2019; Li et al., 2020, 2021]; and *iii*) different settings of loss function are used. Since our loss functions are specifically designed for sequential recommendation scenarios to capture sequential dependencies, we do not include general recommendation models as baselines. Furthermore, it has been empirically shown that non-sequential models, which disregard the sequential order of actions, perform worse compared to sequential models in handling sequential recommendation tasks [Tang and

Wang, 2018; Yu et al., 2019].

To explore the influences of *sequence dependency* and *targets dependency*, as well as to analyze performances of the proposed loss functions in coping with different length of targets (next item and next items), we set different numbers of last items (*i.e.*, T) of the training sequence instances as targets that receive estimated relevance scores for loss calculation. The target length T is in $\{1, 2, 3, 5\}$. As mentioned before, we divide datasets according to the target length. For each real-world dataset (*ML-1M*, *Anime*, or *Beauty*), the last 1, 2, 3, and 5 items of each user's temporal interactions are left for testing, and remaining actions for training and validation. Therefore we obtain four groups (1, 2, 3, and 5 unobserved items) of divided data (test, training, and validation) from each real-world dataset. Correspondingly, four groups of diverse item sets are observed from different training data of each dataset, which are separately used to learn the corresponding diverse kernel matrix $\mathbf{K}$. In this setting, we imitate the scenarios of next item and next T items recommendation, as well as test the ability of DSL and CDSL to cope with different sizes of missing actions.

Three groups of evaluation metrics are adopted for the comprehensive analysis of the recommendation performance:

Quality. We use two common *Top-N* metrics, Recall@N (Re) and NDCG@N (Nd) to evaluate the quality (relevance) of top-scored results, N $\in \{3, 10\}$.

Diversity. Diversification is also considered in the proposed loss functions. Therefore, we evaluate the recommendation diversity by the widely used metric — category coverage (CC), which is calculated by the number of categories covered by *top-N* items divided by the total number of categories available in the dataset [Wu et al., 2019b; Puthiya Parambath et al., 2016]. A higher value of CC@N means the *Top-N* items are more diverse.

Trade-off. To evaluate the overall performance in quality and diversity, a harmonic F-score metric (F) is employed [Cheng et al., 2017].

Following previous work [Wang et al., 2019c, 2018; Liu et al., 2021b], an early stopping strategy is applied to avoid overfitting, *i.e.*, for each experiment we stop training if Nd@10 on the validation set does not increase for 10 successive epochs. To demonstrate the superior effectiveness of our loss functions that aim at sequential recommendation tasks, we deploy DSL and CDSL in three representative works (baselines). Table 4.3 reports the comparison of experimental results between the reworked models and **Caser** under different settings of target length (T). DSL and CDSL in Table 4.3 represent the reworked models that are constructed through replacing the original *binary cross-entropy* loss of Caser by DSL and CDSL, respectively. Similarly, Table 4.4 and Table 4.5 present performance comparisons based on **MARank** and **SASRec**, respectively. Under a setting T of a dataset, experiments of the baseline and

reworked models are carried out using the same training instances and divided data.

Based on these three tables, we aim to demonstrate that deploying DSL or CDSL in existing sequential recommendation models can achieve better performance than using other loss functions, thus indicating the superiority of our loss functions. Therefore, tuning model parameters is not one of our focuses. In each comparison, we directly use the default model parameters of the baseline. This means that we keep the comparison fair. As Figure 4.1 shows, the number of negative samples (Z) is an important component of the user sequence ground set, which is also necessary for the selected baselines and most other related models [He and McAuley, 2016a; Sun et al., 2019; Hu and He, 2019] to calculate the loss. For all experiments of baselines and reworked models, we set $Z = 2$ when $T = 1$ and $Z = T$ in other cases. The effects of Z on performance are also analyzed and compared based on Figure 4.2. Besides, the previous item length L equals 5 when $T = 1$, and $L = 6$ when $T > 1$ to ensure that there are enough training sequences.

4.3.3 Performance Comparison

In Table 4.2-4.4, some important results are marked for clear comparison and emphasis. The value in bold means that a baseline achieves better performance than both DSL and CDSL on an experimental dataset under the setting of T (T test items and T targets). Besides, the marker $\underset{\sim}{\ }$ of a result denotes that a baseline only achieves the higher value than DSL, and the underline means that a baseline beats CDSL under a specific setting. The superscripts *, **, and *** indicate that a reworked model (DSL or CDSL) improves by >5%, >10%, and >20%, respectively over the corresponding baseline *w.r.t.* a metric (in the same column) under a target length T. For instance, in Table 4.3 when $T = 3$ of *ML-1M* dataset, the Re@3 result of CDSL (Caser with CDSL loss) shows 0.0561, which is 10.87% higher than baseline Caser's Re@3 result (0.0506). Therefore 0.0561^{**} is presented.

We can see that there is no DSL result when $T = 1$ in Table 4.2-4.4. This is because DSL is designed for the next T targets estimation as explained in Section 4.2.3. When there is only one target in loss calculation, DSL cannot capture the targets dependency for model training, which is the core intuition of DSL. Therefore, it is not necessary to test DSL when $T = 1$. As for CDSL, even if there is only one target in the training sequence, the conditional set likelihood can be treated as the probability of selecting a DPP-distributed item set (previous items and a target) given the previous items. This means that only *sequence dependency* is considered when $T = 1$, and both *sequence dependency* and *targets dependency* are taken into account when $T > 1$. The main observations from Table 4.2-4.4 are as follows:

Table 4.3: Performance comparison based on **Caser**. Significant improvements over Caser under the same setting are designated as * >5%, ** >10%, and *** >20%. The bold values, marked ~~~~ and ___ denote the Caser achieves better performance than DSL and CDSL, than DSL and than CDSL, respectively.

Dataset		Method	Quality				Diversity		Trade-off	
			Re@3	Re@10	Nd@3	Nd@10	CC@3	CC@10	F@3	F@10
ML-1M	T=1	Caser	0.0595	0.1398	0.0430	0.0713	0.2493	0.5086	0.0850	0.1748
		DSL	-	-	-	-	-	-	-	-
		CDSL	0.0636*	0.1388	0.0460*	0.0724	0.2618*	0.5277	0.0906*	0.1760
	T=3	Caser	0.0506	0.1313	0.1034	0.1634	0.2501	0.5045	0.1177	0.2281
		DSL	0.0551*	0.1369	0.1129*	0.1756*	0.2505	0.4969	0.1258*	0.2377
		CDSL	0.0561**	0.1329	0.1151**	0.1718*	0.2581	0.5118	0.1285*	0.2348
	T=5	Caser	0.0432	0.1151	0.1467	0.2210	0.2549	**0.5371**	0.1383	0.2560
		DSL	0.0458*	0.1204	0.1513	0.2284	0.2533	0.5346	0.1419	0.2630
		CDSL	0.0459*	0.1195	0.1553*	0.2296	0.2585	0.5326	0.1448	0.2629
Anime	T=1	Caser	0.2381	0.4443	0.1829	0.2572	0.2548	0.4947	0.2305	0.4105
		DSL	-	-	-	-	-	-	-	-
		CDSL	0.2700**	0.4681*	0.2115**	0.2820*	0.2519	0.4866	0.2462*	0.4236
	T=3	Caser	0.1667	0.3332	0.3553	0.4144	0.2548	0.4858	0.2579	0.4225
		DSL	0.1814*	0.3643*	0.3588	0.4397*	0.2636	0.4971	0.2668	0.4445*
		CDSL	0.1846**	0.3841**	0.3736*	0.4589**	0.2591	0.4877	0.2687	0.4522*
	T=5	Caser	0.1456	0.2917	0.4226	0.4990	**0.2581**	0.4971	0.2705	0.4404
		DSL	0.1521	0.3170*	0.4575*	0.5292*	0.2550	0.4979	0.2777	0.4575
		CDSL	0.1537*	0.3160*	0.4601*	0.5320*	0.2551	0.4926	0.2786	0.4557
Beauty	T=1	Caser	0.0846	0.1508	0.0538	0.0806	0.0422	0.1012	0.0524	0.1080
		DSL	-	-	-	-	-	-	-	-
		CDSL	0.0944**	0.1817***	0.0716***	0.1029***	0.0441	0.1034	0.0576**	0.1198**
	T=3	Caser	0.0414	0.0975	0.0782	0.1035	**0.0463**	0.0994	0.0528	0.0999
		DSL	0.0568***	0.1388***	0.1045***	0.1420***	0.0459	0.1073	0.0585**	0.1216***
		CDSL	0.0496***	0.1289***	0.0831	0.1260***	0.0452	0.1074	0.0538	0.1166**
	T=5	Caser	0.0434	0.1219	0.1258	0.1758	0.0433	0.1032	0.0573	0.1219
		DSL	0.0482**	0.1306*	0.1320	0.1822	0.0441	0.1008	0.0592	0.1226
		CDSL	0.0554***	0.1430**	0.1468**	0.1954**	0.0440	0.1052	0.0613*	0.1297*

Table 4.4: Performance comparison based on **MARank**. Significant improvements over MARank under the same setting are designated as * >5%, ** >10%, and *** >20%. The bold values, marked ~~~ and ___ denote the MARank achieves better performance than DSL and CDSL, than DSL and than CDSL, respectively.

Dataset		Method	Quality				Diversity		Trade-off	
			Re@3	Re@10	Nd@3	Nd@10	CC@3	CC@10	F@3	F@10
ML-1M	T=1	MARank	0.0582	0.1470	0.0437	0.0746	0.2627	0.5407	0.0853	0.1839
		DSL	-	-	-	-	-	-	-	-
		CDSL	0.0679**	0.1622**	0.0500**	0.0832**	0.2711	0.5504	0.0968**	0.2007*
	T=3	MARank	0.0498	0.1222	0.1066	0.1645	0.2651	0.5451	0.1208	0.2270
		DSL	0.0524*	0.1303*	0.1105	0.1687	0.2684	0.5456	0.1250	0.2347
		CDSL	0.0528*	0.1308	0.1093	0.1701	0.2743	0.5535	0.1251	0.2366
Anime	T=1	MARank	0.1875	0.3737	0.1433	0.2090	0.2664	0.4696	0.2041	0.3596
		DSL	-	-	-	-	-	-	-	-
		CDSL	0.2211**	0.4453**	0.1683**	0.2477**	0.2617	0.4734	0.2233*	0.4001**
	T=3	MARank	0.1346	0.2616	0.2542	0.3361	**0.2521**	**0.4585**	0.2195	0.3618
		DSL	0.1876***	0.3826***	0.3644***	0.4487***	0.2476	0.4570	0.2610**	0.4353***
		CDSL	0.1738***	0.3591***	0.3361***	0.4264***	0.2458	0.4517	0.2503**	0.4202**
Beauty	T=1	MARank	0.1103	0.2310	0.0830	0.1259	0.0423	0.0967	0.0588	0.1254
		DSL	-	-	-	-	-	-	-	-
		CDSL	0.1183*	0.2214	0.0925**	0.1292	0.0415	0.0981	0.0596	0.1258
	T=3	MARank	0.0543	**0.1542**	0.0966	**0.1434**	0.0417	0.0964	0.0537	**0.1170**
		DSL	0.0606**	0.1428	0.1001	0.1413	0.0421	0.0991	0.0553	0.1168
		CDSL	0.0552	0.1348	0.0975	0.1324	0.0425	0.0970	0.0546	0.1124

Table 4.5: Performance comparison based on **SASRec**. Significant improvements over SASRec under the same setting are designated as * >5%, ** >10%, and *** >20%. The bold values, marked ~~~ and ___ denote the SASRec achieves better performance than DSL and CDSL, than DSL and than CDSL, respectively.

Dataset		Method	Quality				Diversity		Trade-off	
			Re@3	Re@10	Nd@3	Nd@10	CC@3	CC@10	F@3	F@10
ML-1M	T=1	SASRec	0.0484	0.1208	0.0351	0.0601	0.2526	0.4616	0.0717	0.1513
		DSL	-	-	-	-	-	-	-	-
		CDSL	0.0490	0.1217	0.0354	0.0604	0.2732**	0.4755	0.0731	0.1528
	T=3	SASRec	0.0330	**0.0891**	0.0596	0.1110	0.2656	0.4804	0.0789	0.1656
		DSL	0.0350*	0.0888	0.0658**	0.1131	0.2665	0.4847	0.0848*	0.1671
		CDSL	0.0331	0.0869	0.0613	0.1089	0.2707	0.4880	0.0804	0.1631
Anime	T=1	SASRec	0.2628	0.5108	0.2007	0.2888	0.2293	0.4336	0.2305	0.4160
		DSL	-	-	-	-	-	-	-	-
		CDSL	0.2777*	0.5235	0.2160*	0.3049*	0.2339	0.4459	0.2402	0.4295
	T=3	SASRec	0.1677	0.3505	0.2510	**0.3809**	0.2331	0.4462	0.2205	0.4019
		DSL	0.1749	0.3705*	0.2413	0.3642	0.2406	0.4419	0.2232	0.4012
		CDSL	0.1865**	0.3530	0.2654	0.3761	0.2384	0.4493	0.2320*	0.4025
Beauty	T=1	SASRec	0.0873	0.2079	0.0665	0.1086	0.0379	0.0782	0.0508	0.1047
		DSL	-	-	-	-	-	-	-	-
		CDSL	0.1032**	0.2286	0.0804***	0.1239**	0.0386	0.0780	0.0543*	0.1081
	T=3	SASRec	0.0645	0.1627	0.0829	0.1425	**0.0394**	0.0808	0.0513	0.1056
		DSL	0.0686*	0.1565	0.0894*	0.1433	0.0383	0.0828	0.0516	0.1066
		CDSL	0.0679*	0.1646	0.0909**	0.1528*	0.0389	0.0834	0.0522	0.1093

- Aligning these three tables, we can see that deploying DSL or CDSL on the state-of-the-art sequential recommendation models significantly improves recommendation quality (recall and NDCG) *w.r.t.* different *Top-N* on different datasets. This suggests that set likelihood-based objective functions that consider sequence dependency and targets dependency are more adaptable to sequential recommendation tasks than commonly used ranking-based loss functions.

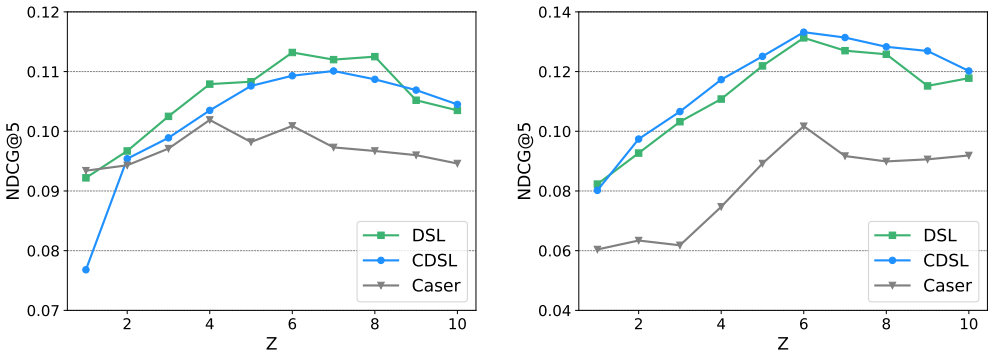
- In only one case — MARank model on Beauty dataset when $T = 3$, the value of *trade-off* metric F@10 is slightly higher than DSL and CDSL. Under two settings (ML-1M dataset when $T = 3$ and Anime dataset when $T = 3$), SASRec outperforms the reworked DSL or CDSL version *w.r.t.* trade-off metric. In addition to these three cases, DSL and CDSL achieve better trade-off performance than the original loss functions of baselines. In some cases, the reworked models outperform baselines in F-score by a large margin (over 20%), *e.g.*, reworked MARank on Anime when $T = 3$. These demonstrate that DSL and CDSL can better handle the balance between quality and diversity by DPP kernel decomposition, *i.e.*, facilitating recommending relevant and diverse results.

- When the original models achieve better diversification (CC metric) than their reworked versions, *e.g.*, experiments on Anime dataset ($T = 1$) in Table 4.3, their corresponding quality performances (Recall and NDCG at the same top N) are clearly inadequate. This observation shows that state-of-the-art baselines sacrifice recommendation relevance for diversity (*i.e.*, sub-optimally trade-off between quality and diversity).

- In those few instances that baseline obtains better recommendation quality than DSL or CDSL in three tables, they are all at top-10 (*e.g.*, Re@10 on ML-1M when $T = 1$ in Table 4.3). This indicates that in a few cases (only 3 values in Table 4.2-4.4), DSL and CDSL may focus more on providing the 'really' relevant (*e.g.*, top-3) items with a high probability of being drawn as a DPP set but ignore the relatively less relevant ones in a list.

- By comparing the performances under different settings of target length T in Table 4.3, $T = 3$ setting generally yields the most significant improvement compared to the baseline Caser. The reason might be that as the number of target items increases (*e.g.*, $T = 5$), noisy or confusing dependencies between the targets is considered, as thus affecting recommendations. In view of this phenomenon, Table 4.4 and Table 4.5 only present the results under $T = 1$ and $T = 3$ settings.

- Among three experimental datasets in Table 4.2-4.4, DSL and CDSL, in general, achieve the most significant improvement on Beauty dataset compared to the corre-



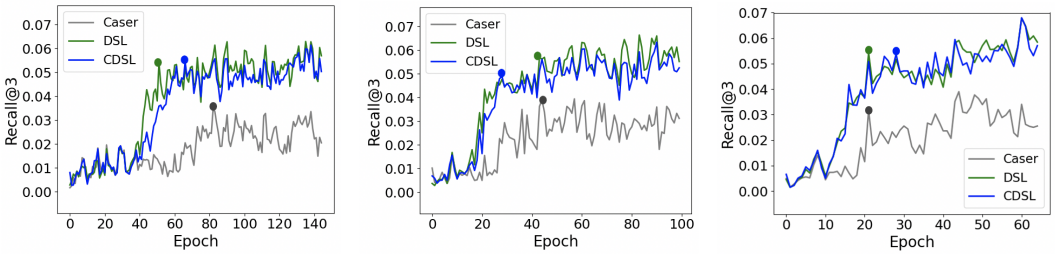
(a) ML-1M dataset (b) Beauty dataset

Figure 4.2: Affects of negative sample length Z on ML-1M and Beauty datasets.

sponding baseline. We think this is because on shopping Web platforms, users tend to search or buy products (items) that have tight correlations over a period. This verifies the importance of dependencies in the sequence or among targets to some extent, because dependencies are considered in the proposed loss functions.

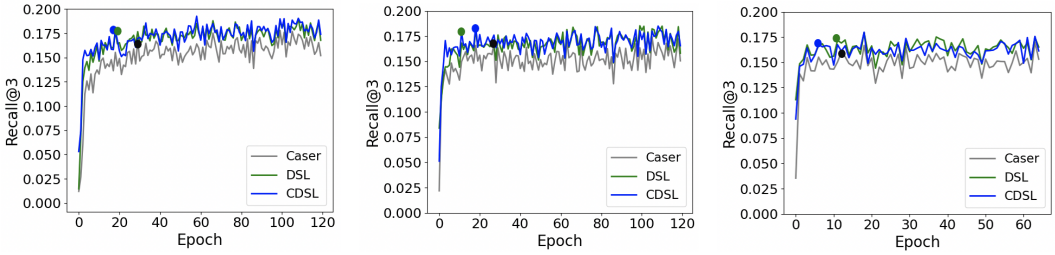
- Compared to DSL in three tables, CDSL in general achieves better overall performance (F@N) when $T > 1$. This further verifies the importance of considering sequence dependency and targets dependency in the loss function, as CDSL considers both of them. Although CDSL cannot beat DSL in most cases with respect to quality metrics, it has a distinct advantage in diversification (CC@N). This finding tells us that taking previous actions into account (*i.e.*, increasing the size of the ground set $\mathcal{Y}^{(u,t)}$) is likely to make the decomposition of user-sequence specified DPP kernel care more about diversity than quality, thus leading CDSL to achieve better diversity performance. This can also be treated as an advantage of CDSL, because it is the distinct diversity result that contributes to the better overall performance.

Among all observations, there is no comparison between the performances of any two methods under the same setting but in different tables. This is because our purpose of presenting Table 4.2-4.4 is to demonstrate the superiority of DSL and CDSL through analyzing the performance improvement brought by using DSL or CDSL compared to the corresponding baseline with its original loss function under the same setting. Therefore, there is no need to compare the results of different tables. In the design of Caser, T targets are considered in loss calculation, so we can directly feed training instances of L previous items and T targets into its *binary cross-entropy loss*. Although the loss functions in MARank and SASRec are designed only for next one target, we still can deploy DSL or CDSL in them by modifying the training sequences and expand the original losses that consider one target to comparing T targets (Equation 4.1 and



(a) lr=0.0005 (b) lr=0.001 (c) lr=0.0015

Figure 4.3: Test performance of each epoch with different learning rate on Beauty.



(a) lr=0.0005 (b) lr=0.001 (c) lr=0.0015

Figure 4.4: Test performance of each epoch with different learning rate on Anime.

Equation 4.3) when $T > 1$. As the architecture of baselines is not altered, performance comparisons are still fair. Essentially, we only compare the loss functions.

We use Figure 4.2 to investigate the effect of negative sample length Z. It shows that for all models (Caser and its reworked versions — DSL and CDSL) on ML-1M and Beauty under the setting of $T = 2$ and $L = 5$, with the increase of Z the overall trend is that the NDCG performance grows rapidly at first and then gradually drops. For all experiments reported in Table 4.2-4.4, Z is set according to the target length T like most recommendation models do, instead of searching the optimal value of Z by tuning Z. This helps to keep the comparisons fair and the training efficient. In most cases, DSL and CDSL perform better than Caser and the improvement over Caser increases as the value of Z gets larger. When Z equals 1, the performance of CDSL on ML-1M is not better than Caser. This may be because there is not enough pattern/correlation for learning a competent DPP probability measure, when conditioning on 5 previous items for drawing a DPP set with 7 items ($L + T$) that almost equals the item number of the ground set ($L + T + Z = 8$). These observations indicate that negative samples are not only important for the original loss function but also for DSL and CDSL.

Figure 4.3 shows the test performance (Recall@3) of each training epoch of three models with different learning rates (`lr`) on Beauty. Figure 4.4 presents the same type

Table 4.6: Training efficiency comparison between MARank and our approaches on Anime and Beauty.

Datasets		MARank			DSL				CDSL			
		Time	#Epochs	Total	Time	#Epochs	Total	pct.	Time	#Epochs	Total	pct.
Anime	T=1	22.03	35	771.05	-	-	-	-	48.67	17	827.39	7.30%
	T=3	25.37	38	964.06	52.72	19	1001.68	3.90%	55.15	19	1047.85	8.69%
Beauty	T=1	1.02	40	40.80	-	-	-	-	1.88	23	43.24	5.98%
	T=3	1.06	42	44.52	1.94	23	44.62	0.22%	1.98	25	49.50	11.19%

of performance comparison on Anime. The value of 0.001 is the default setting of `lr` for the Caser baseline, which is also used for DSL and CDSL versions, with the same setting ($T = 3$, $L = 5$, and $Z = 3$) for all experiments. The solid circle on a performance curve of each subfigure indicates the best performance before the early stopping. In Figure 4.3 and Figure 4.4 , we use Recall@3 metric to judge whether the early stopping strategy is triggered. The solid circle has the same color as the corresponding model's performance curve. Two main observations can be made: *i)* the proposed loss functions achieve better performance almost at any epoch in six subfigures (three `lr` settings on two datasets) compared to the baseline, indicating the superiority of our loss functions in a more detailed way; *ii)* in general, the proposed loss functions prefer a smaller learning rate compared to the original loss, and they learn faster (*i.e.*, trigger the early stopping earlier) in most cases. As the reworked models of other baselines share similar trends with Caser on different datasets, other subfigures with different settings are not presented.

Table 4.6 shows the training efficiency comparison between MARank (using pairwise loss) and reworked models with our loss functions on two datasets under different settings of target length. In the table, `Time` means average training time per epoch (in seconds), and #`Epochs` represents the number of epochs needed to converge (achieve best NDCG@10 performance before early stopping). `Total` records the total time used in a training process `Time` × #`Epochs`. Besides, `pct.` denotes the *percent change* between our reworked models and the baseline. All experiments in Table 4.6 are conducted using a single Tesla K40 GPU under the same setting. We can see that the reworked models with DSL or CDSL take more time per epoch than MARank, but converge faster (fewer epochs are needed in training process) and perform better (shown in Table 4.4). This observation is also made in Figure 4.3 and Figure 4.4 . In general, DSL and CDSL only need around 10% more time to coverage than that of MARank, but significant improvements are achieved (as shown in Table 4.2-4.4). In our experiments, when we compare the training efficiency based on Caser or SASRec (using pointwise ranking), similar observations to those of Table 4.4 are made. Therefore, comparisons based on other baselines and datasets are not presented.

From the analysis of Table 4.2-4.4 and Figure 4.2 , we can draw a clear conclusion

about how to make a choice between DSL and CDSL in certain circumstances. If a sequential recommender system emphasizes relevance and training efficiency, DSL is a better choice. On the other hand, in the case where overall performance between quality and diversity is the main concern (*e.g.*, diversity-promoting recommendation [Wu et al., 2019b; Liang and Qian, 2021]) and training efficiency is not the primary factor, choosing CDSL is more advisable.

4.4 Conclusion

In this chapter, we present two novel DPP set likelihood-based loss functions (DSL and CDSL) aiming at sequential recommendations, which enlighten a new perspective (set selection) to formulate the temporal prediction training objectives. As such, the *sequence dependency* and *targets dependency* are considered in the loss formulations. In addition, through pre-learning a diverse kernel from observed browsing history and using the quality *vs.* diversity decomposition of user-sequence specified DPP kernel, the proposed loss functions are pushed to be diversity-aware. By simply replacing the original loss of state-of-the-art sequential recommendation models with our DSL or CDSL objective functions, significant improvements can be received, which demonstrates the superiority and flexibility of the proposed loss functions. Experimental results help us analyze the different advantages of the two loss functions and provide insight into making selection between them in different circumstances.

4.4.1 Limitations

The explorations in this chapter, while insightful, present certain limitations: (i) The computation of our proposed DSL and CDSL objective functions considers only the likelihood of explicitly defined positive items, neglecting the distribution corresponding to negative items, which may constrain the model's efficacy; (ii) This chapter solely explores the application of DSL and CDSL to sequential recommendation. Given the varied definitions of quality and diversity across different tasks, the universality of DSL and CDSL requires further enhancement to facilitate swift migration across various research domains.

4.4.2 Future Work

In light of these limitations, we propose corresponding future work, (i) we plan to design a more comprehensive objective function that considers the DPP distribution of both positive and negative items, aiming to augment the model's performance; (ii) we intend to explore the performance of DSL and CDSL across different tasks, such as

clinical events prediction [Hu and He, 2019; Choi et al., 2016] and next-basket recommendation [Warlop et al., 2019], and design DPP likelihood-based objective functions with broader adaptability. Furthermore, integrating the conditional DPP generation method proposed in Chapter 3 with the DPP conditional distribution-based objective function introduced in this chapter can enhance the application of the DPP conditional distribution, thereby improving the predictive performance of the model. This integration also presents itself as a viable avenue for future work, potentially unlocking new potentials and applications in the domain of recommender systems.

Chapter 5

Learning k-Determinantal Point Processes for Diversity-Aware Ranking Optimization

In previous Chapter 3 and Chapter 4, we explore how to apply standard DPPs to building recommendation frameworks for traditional recommender systems and designing the learning objective functions for sequential recommendation, respectively. After considering items' dependencies in the conditional MAP inference (Chapter 3) and objective function (Chapter 4), improved performances are observed. This inspires us to explore the implementation of introducing dependencies (relevance level and diversity level) into personalized ranking by employing k-DPPs.

The key to personalized recommendation is to predict a personalized ranking on a catalog of items by modeling the user's preferences. There are many personalized ranking approaches for item recommendation from implicit feedback like Bayesian Personalized Ranking (BPR) and listwise ranking. Despite these methods have shown performance benefits, there are still limitations affecting recommendation performance. First, none of them directly optimize ranking of sets, causing inadequate exploitation of correlations among multiple items. Second, the diversity aspect of recommendations is insufficiently addressed compared to relevance.

In this work, we present a new optimization criterion LkP based on set probability comparison for personalized ranking that moves beyond traditional ranking-based methods. It formalizes set-level relevance and diversity ranking comparisons through a Determinantal Point Process (DPP) kernel decomposition. To confer ranking interpretability to the DPP set probabilities and prioritize the practicality of LkP, we condition the standard DPP on the cardinality k of the DPP-distributed set, known as k-DPP, a less-explored extension of DPP. The generic stochastic gradient descent based technique can be directly applied to optimizing models that employ LkP. We implement LkP in the context of both Matrix Factorization (MF) and neural networks

approaches, on three real-world datasets, obtaining improved relevance and diversity performances. LkP is broadly applicable, and when applied to seminal CF models it also yields strong performance improvements, suggesting that LkP holds significant value to the field of recommender systems.

5.1 Introduction

Recommender systems are ubiquitous services in many online platforms including social media and e-commerce, and play a crucial role in alleviating information overload by efficiently anticipating user preferences. This work focuses on item recommendation [Rendle et al., 2012; He et al., 2017b], wherein the task involves generating a personalized ranking of items based on the user's observed interactions with the system. In real-world scenarios, implicit feedback, such as clicks and views, is more readily available and collected compared to explicit feedback, e.g., ratings and reviews, making it a valuable resource for understanding user preferences.

To guide recommendation models in learning user preferences from implicit feedback, optimization objectives play a crucial role. One commonly used approach involves pointwise loss functions, *e.g.*, binary cross entropy and squared error, which measure the discrepancy between the model's predicted preference scores and the actual labels on an individual instance basis. Expanding on this, reflecting on the goal of item recommendation, which is to provide a personalized ranking list to a specific user, it naturally leads to the adoption of ranking-based optimization methods. Among these, Bayesian Personalized Ranking (BPR), listwise models, and the newer setwise models are some of the principal techniques used in this domain.

The popular pairwise ranking approach, BPR [Rendle et al., 2012], assumes that an observed item ranks higher than a randomly selected unobserved (negative) item from the perspective of a specific user. Listwise methods usually maximize the probability of permutation of the target list [Cao et al., 2007; Xia et al., 2008]. In recent years, some studies have attempted to bring the correlation within items into ranking-based optimization, leading to the topic of setwise ranking. Two eminent and state-of-the-art set-based ranking methods are SetRank [Wang et al., 2020] and Set2SetRank [Chen et al., 2021]. SetRank encourages an observed item to rank in front of multiple unobserved items in each list by making use of the concept of permutation probability. Set2SetRank models the set ranking based on: (i) item to item comparison, *i.e.*, encouraging each item of an observed set to rank higher than any items of the unobserved set, and (ii) set distance measurement, *i.e.*, reinforcing a margin of distance (summarized based on comparisons of items in sets) between observed and unobserved sets. The upper layer of Figure 5.1 is displayed to succinctly explain different types of optimization

method based on a toy example.

The approaches mentioned above indeed yield positive effects on recommendation models. However, we argue that two aspects that are important to recommender systems have not been adequately explored. First, the structure information and correlations are far from well exploited and modeled. For example, the correlations among items of observed and unobserved sets are fully ignored by pointwise and pairwise learning; listwise based models, centered solely on the order correlation in the ranking list of items, face significant challenges, especially with the binary nature of implicit feedback obstructing clear sequential relationships; the recently proposed setwise ranking optimization models (*e.g.*, SetRank and Set2SetRank) intend to construct set-level comparisons, but are still confined to a BPR optimization criterion, and the set ranking is obtained by summarizing the individual items in each set, rather than recognizing multiple items as a whole (*i.e.* set) that they are in order to more meaningfully compare them. Second, the objective of diversifying recommendations to mine additional interests for users, is not involved in existing ranking optimization approaches. Consequently, item recommendation models that employ these optimization objectives tend to favor popular and rather relevant items (often already known by users) [Wu et al., 2019b], but lose sight of raising the ranking of diversified results for broadening users' horizons.

To overcome these limitations, this work introduces a concrete set-level optimization criterion that facilitates ranking optimization through a direct comparison of different sets' probabilities of being a k-DPP. Here, k-DPP refers to a less-explored extension of the well-established concept, Determinantal Point Process (DPP). By leveraging the balance-focused quality *vs.* diversity decomposition of the DPP kernel, we are able to seamlessly integrate two types of ranking comparison (*i.e.*, *relevance comparison* and *diversity comparison*), thus offering a comprehensive measure of sets. The motivation for employing k-DPP lies in its ability to add ranking interpretability to DPP set probabilities, making them practically useful for achieving optimal recommendation sets. We name the proposed optimization criterion as LkP, an acronym underlining its aim to **l**earn the **p**robability of a specific subset under a tailored k-DPP distribution. LkP offers an effective solution that acknowledges item correlations and diversity, thereby delivering an advanced recommendation model that boosts accuracy and diversity.

We summarize the contributions of this chapter as follows:

- We propose a concrete set-level optimization criterion, named LkP, that advances ranking optimization by allowing a direct comparison of set probabilities under a tailored k-DPP distribution, moving beyond the common BPR metric. Unlike prior approaches that focus on individual items, our method provides a comprehensive comparison at the set level.

Table 5.1: The Key Symbols Used in DDPR Framework.

Notation	Interpretation
$\mathbb{U}$, $\mathbb{V}$	user set, item set
N, M	user number, item number
$\mathbf{Y}$,	observed interaction matrix
$T^{(+)}$, $T^{(-)}$	observed diverse set, unobserved item set
$\mathbf{K} \in \mathbb{R}^{M \times M}$	diverse kernel on entire ground set
Y_u^+, $Y_u^- = \mathbb{V} \backslash Y_u^+$	observed interactions of user u, non-observed feedback
$\hat{\mathbf{Y}} \in \mathbb{R}^{N \times M}$, $\hat{y}_{uv}$	predicted preference matrix, predicted relevence for u on v
$\mathbf{L} \in \mathbb{R}^{M \times M}$	DPP kernel on entire ground set
k, n	size of positive subset, negative item number
$\mathbf{L}^{u,k+n}$	user-specified DPP kernel on $k+n$ ground set

- We utilize k-DPP to provide ranking interpretability to DPP set probabilities, rendering them practically advantageous for ranking optimization. As a result, our LkP criterion recognizes item correlations and diversity, thus bolstering the accuracy and diversity of the recommendation model.
- By incorporating both the actual inclusion probability of an observed set and the exclusion probability of an unobserved set, we develop two distinct optimization approaches, namely LkP$_{PS}$ and LkP$_{NPS}$, which we introduce and compare.
- The derivation of the gradient of LkP with respect to the model parameters demonstrates the feasibility of optimizing our approaches via stochastic gradient descent-based methods, thus enabling the maximization of LkP.

5.2 Methodology

In this section, the preliminary and LkP optimization criterion based on k-DPP set probability comparison for personalized ranking are successively presented. We summarize the important notation in Table 5.1.

5.2.1 Preliminaries

Item Recommendation Model. A recommendation problem is generally defined given user set $\mathbb{U}$ and item set $\mathbb{V}$ with N and M as the size of $\mathbb{U}$ and $\mathbb{V}$, respectively. Let $\mathbf{Y} \in \{0,1\}^{N \times M}$ denotes the user-item interaction matrix. Y_u^+ indicates observed interactions of user u and $Y_u^- = \mathbb{V} \backslash Y_u^+$ is the non-observed feedback (negative). Given

the interaction matrix $\mathbf{Y}$, the goal of item recommendation is to anticipate user preferences by predicting a preference score matrix $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times M}$ with u, v-th element $\hat{y}_{uv}$, typically with ranking-based optimization approaches that are suitable for selecting items of likely interest to users.

As one of the most promising technologies to build recommender systems, collaborative filtering (CF) uses the known collaborative perception of users to make recommendations [Su and Khoshgoftaar, 2009]. There are generally four varieties of CF models: matrix factorization (MF) learns user and item embeddings based on a low-rank decomposition of the user-item interaction matrix [Mnih and Salakhutdinov, 2007; Koren et al., 2009; Koren, 2008]; neural collaborative filtering replaces the inner product (a common notion of interaction in MF) with nonlinear neural networks [He et al., 2017b]; collaborative deep learning couples deep representation learning for content information and collaborative filtering for feedback matrix [Wang et al., 2015; Chen et al., 2020]; and translation-based CF adopts latent user-item relation vectors to model interactions [Tay et al., 2018; He et al., 2017a]. Among these models, Graph Convolutional Network (GCN), a category of collaborative deep learning, has gained dominance due to their effective modeling of item-item relationships through graph neural networks. Our proposed optimization criterion, LkP, is distinct from CF models that contribute to the architectural design of recommender systems. It serves as a guiding principle for optimizing the personalized ranking of recommendation sets.

Determinantal Point Processes. We revisit the concept of DPP and present its mathematical representation using notations more closely aligned with the context of this chapter, facilitating comprehension and setting the stage for the introduction of k-DPP. Given a discrete set $\mathcal{S} = \{1, 2, \ldots, M\}$ (*e.g.*, item set $\mathbb{V}$), a DPP $\mathcal{P}$ is a probability measure on $2^{\mathcal{S}}$, the set of all subsets of $\mathcal{S}$. When $\mathcal{P}$ gives nonzero probability to the empty set, there exists a matrix $\mathbf{L} \in \mathbb{R}^{M \times M}$ such that for every subset $S \subseteq \mathcal{S}$, the probability of S is

$$P(S) = \frac{\det(\mathbf{L}_S)}{\det(\mathbf{L} + \mathbf{I})}, \tag{5.1}$$

where $\mathbf{L}$ is a real, positive semi-definite kernel matrix indexed by the elements of S, and $\mathbf{I}$ is the $M \times M$ identity matrix. In this standard DPP setup, a probability is assigned to every subset (including the empty set and the entire $\mathbb{V}$). This is not desirable in the practical problems that require explicit control over the number of items selected by the model, such as image search engines that provide a fixed-sized array of results in a page [Kulesza et al., 2012]. If the model based on a standard DPP gives some probability of suggesting no or every image, this is not ideal. Given this situation, an extension of DPP, *i.e.*, k-DPP [Kulesza and Taskar, 2011a], has been proposed, which conditions a DPP on the cardinality k of the random set.

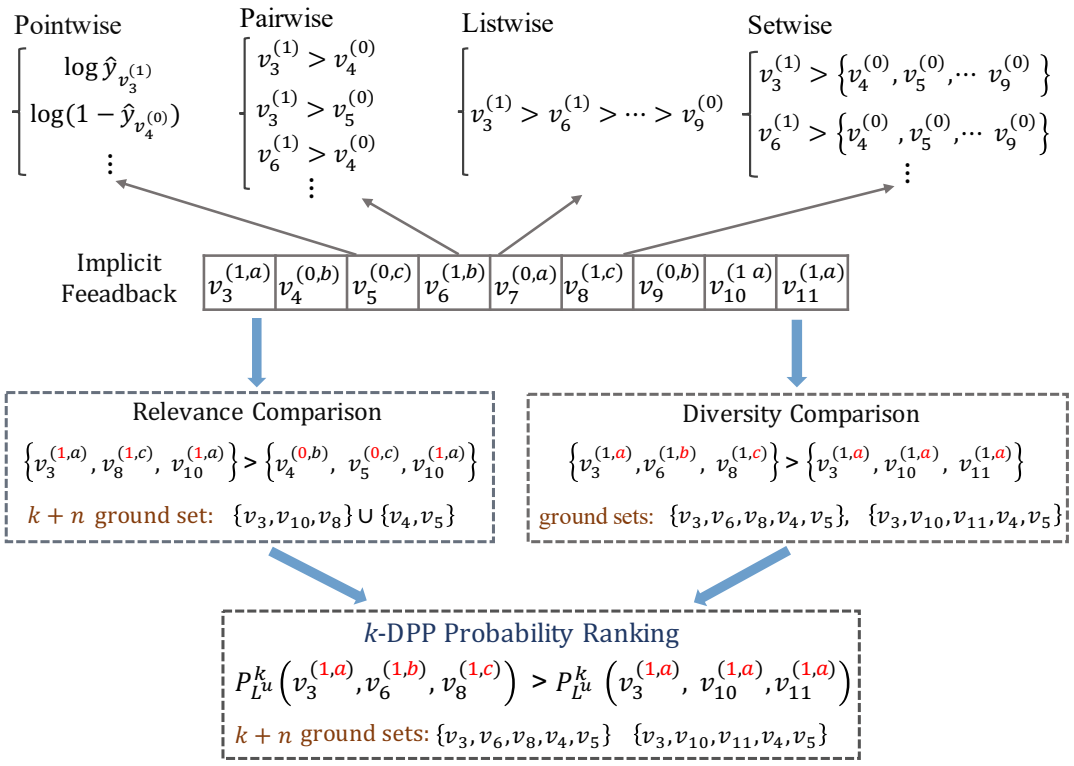


Figure 5.1: The diagrammatic sketch of LkP. In this illustration, each square in the middle represents an item (denoted as v), with the subscript numeral indicating the item's index. The superscript 1 signifies an observed implicit interaction between the user and the item, while a superscript 0 indicates an unobserved interaction. The superscripts a, b, and c denote three distinct categories of items. The top layer provides a simple comparative illustration of different optimization objectives. The layers below show our two types of ranking comparison, namely relevance and diversity comparisons, which are integrated using the k-DPP probability ranking. The focal points of comparison are indicated by red markers.

5.2.2 LkP Ranking

Our objective is to explore k-DPP in the context of optimization criterion, for providing relevant and diverse suggestions. If a purely diversity oriented kernel (whose entries are measurements of pairwise similarity) is adopted directly, the suggested results will bear no relation to user satisfaction, as only the most repulsive items (dissimilar) will be selected. Therefore, a user-specific kernel $\mathbf{L}^u$ is required, which we obtain via the so-called quality *vs.* diversity decomposition of DPP kernel,

$$\mathbf{L}^u = \mathrm{Diag}\left(\hat{\mathbf{y}}_u\right) \cdot \mathbf{K} \cdot \mathrm{Diag}\left(\hat{\mathbf{y}}_u\right). \tag{5.2}$$

Following previous studies [Warlop et al., 2019; Liu et al., 2022], the $M \times M$ matrix $\mathbf{K}$ models diversity, and is learned from historical interactions. In this work, diversity is evaluated by category coverage [Santos et al., 2010; Puthiya Parambath et al., 2016], an intuitive and popular diversity-related metric in implicit feedback recommendation. To associate $\mathbf{K}$ with the item category, we use diversified item sets (subsets that have a broad coverage) of historical interactions as ground truth sets for training [Liu et al., 2022; Wu et al., 2019b]. In this way, a subset with more categories has a higher possibility of being selected under the DPP as compared to less diverse ones. To reduce the computational complexity of calculating a $M \times M$ matrix, the diversity kernel is represented using a low-rank factorization $\mathbf{K} = \mathbf{V}^\top \mathbf{V}$, learned based on the following objective function,

$$\mathcal{J} = \sum_{(T^+, T^-) \in \mathcal{T}} \log\det\left(\mathbf{K}_{T^+}\right) - \log\det\left(\mathbf{K}_{T^-}\right). \tag{5.3}$$

Here, $T^{(+)}$ is an observed diverse set and $T^{(-)}$ represents the set that contains negative items, and $\mathcal{T}$ denotes the set of paired sets used for training. $\mathbf{K}$ is not related to users, as it is only the diversity-associated kernel.

The diagonalized $\hat{\mathbf{y}}_u$ plays a personalization role in the kernel $\mathbf{L}^u$, capturing the preference scores of user u (predicted by recommendation models) on entire items. The more we can learn from Equation 5.2 is that the quality component can be represented by different types of relevance calculation, *e.g.*, modeling user-item interaction with inner product [Rendle et al., 2012; He et al., 2020] or using nonlinear neural networks [He et al., 2017b]. This extended generality is evaluated in Section 5.3.

5.2.2.1 Relevance Comparison

To achieve a set-level relevance comparison by comparing k-DPP subsets probabilities, we propose a specialized k-DPP distribution dedicated to personalized ranking over a specific $k+n$ ground set (with k observed items in Y_u^+ and n random items in Y_u^-). It is important to stress that a k-DPP assigns probabilities only to subsets of a specific size (cardinality k). That is, given a k-DPP distribution over $k+n$ ground set from user u, we can only compare probabilities of k-sized subsets. We opt to define this ground set rather than directly using $\mathbb{V}$, primarily due to considerations of computational efficiency, as learning the k-DPP probabilities over $k+n$ items is far more efficient than attempting to do so over the entire item set. Another fundamental motivation behind this $k+n$ scheme lies in the necessity for our k-DPP distribution to exhibit interpretability in terms of ranking. By defining the $k+n$ ground set, we can confer the tailored k-DPP distribution with the *relevance ranking interpretation* of implicitly

comparing item sets' relevance rankings. This, in turn, enables the construction of a concrete set-level optimization criterion for personalized ranking.

Specifically, given a cardinality k constraint, we consider two types of subsets from the $k+n$ ground set, *i.e.*, the unique *target subset* filled with all k targets and other subsets that are comprised of $k-z$ targets and z unobserved items ($z \in \mathbb{N}^+$ and $z \leq k$, and $z = k$ denotes a subset of unobserved items). As shown in the relevance comparison part in Figure 5.1 with 3 (k) targets and 2 (n) negative samples, it becomes intuitively clear that, given equal-sized sets, we aim to achieve a ranking comparison where the k-DPP probability of the sole *target subset* $\left\{v_3^{(1,a)}, v_{10}^{(1,a)}, v_8^{(1,c)}\right\}$ is higher than the probabilities of all other k-subsets such as $\left\{v_4^{(0,b)}, v_5^{(0,c)}, v_{10}^{(1,a)}\right\}$ ($z = 2$).

We consider two additional scenarios that help to intuit why the $k+n$ ground set is needed. First, if more than k items in Y_u^+ are added into the ground set, it necessitates the comparison of a k-DPP target subset with some subsets that are formed by other combinations of k targets. For instance, in the assumed ground set, it is possible to observe a ranking comparison like $\left\{v_3^{(1,a)}, v_{10}^{(1,a)}, v_8^{(1,c)}\right\} > \left\{v_3^{(1,a)}, v_6^{(1,b)}, v_8^{(1,c)}\right\}$. Second, when applying a standard DPP over the same $k+n$ ground set for personalized ranking, it entails competition between a drawn DPP subset with k targets and all other target subsets of variable cardinality (*e.g.*, one target or two targets), resulting in ranking comparisons such as $\left\{v_3^{(1,a)}, v_{10}^{(1,a)}, v_8^{(1,c)}\right\} > \left\{v_3^{(1,a)}, v_8^{(1,c)}\right\}$. These will lead to an obscure training signal that does not properly reflect the nature of the set valued targets.

5.2.2.2 Diversity Comparison

To implement diversity comparison, we also need to consider the k-DPP scenario. This consideration allows us to rank diversity in a way that is not affected by differences in set sizes, thus ensuring a fair comparison across subsets. Our methodology brings in diversity comparison based on the diverse kernel $\mathbf{K}$. Its learning objective (Equation 5.3) facilitates attributing larger log-determinant values to sets encompassing a broader range of categories, thereby endowing the k-DPP distribution with the *diversity ranking interpretation* (detailed in the next section). As depicted in Figure 5.1, when focusing solely on the degree of diversity (categories), the corresponding model is expected to favor sets that encompass a greater number of distinct categories, that is, $\left\{v_3^{(1,a)}, v_6^{(1,b)}, v_8^{(1,c)}\right\} > \left\{v_3^{(1,a)}, v_{10}^{(1,a)}, v_{11}^{(1,a)}\right\}$. Since $\mathbf{K}$ does not bear personalization, the focus of comparison is solely on categories. We specifically evaluate the diversity of relevant items, as we aim to guide models towards diversifying the content that is of interest to users, rather than blindly pursuing diversity for its own sake.

5.2.2.3 Unified Comparison

As discussed in Section 5.2.2.1, the tailored k-DPP probabilities are formulated as

$$P^{k}_{\mathbf{L}^{(u,k+n)}}(S_u^{+k}) = \frac{\det\left(\mathbf{L}^{(u,k+n)}_{S_u^{+k}}\right)}{\sum_{|S'|=k}\det\left(\mathbf{L}^{(u,k+n)}_{S'}\right)}, \tag{5.4}$$

where S_u^{+k} represents the k-sized *target subset* drawn from the $k+n$ ground set. $\mathbf{L}^{(u,k+n)} \in \mathbb{R}^{(k+n)\times(k+n)}$ indicates the personalized kernel on $k+n$ ground set of user u, which is used to calculate the probability of S_u^{+k} being a k-DPP. It is directly calculated by incorporating predicted scores of $k+n$ items and sub-matrix of $\mathbf{K}$ indexed by $k+n$ items referring to Equation 5.2. We use S' to denote every cardinality k set from the $k+n$ ground set in the normalization constant. Hence in the tailored k-DPP setup a k-subset competes only with sets of the same cardinality, while in a standard DPP every k-set competes with all other subsets of the ground set.

For an easier understanding of relevance comparison, we can initially assume that $\mathbf{L}^{(u,k+n)}$ is solely comprised of relevance scores, *i.e.*, a diagonal matrix. By maximizing this tailored k-DPP probability, our optimization criterion LkP learns to favor the unique target subset S_u^{+k} over all other subsets of the same size. This implies that the target subset is endowed with a higher probability of being a k-DPP-distributed subset than other k-sized subsets from $k+n$ ground set. This imparts the *relevance ranking interpretation* to the k-DPP distribution from the preference perspective of user u.

As mentioned in Section 5.2.2.2, the pre-learned diverse kernel $\mathbf{K}$ (incorporated into the personalized kernel) enables the tailored k-DPP distribution to compare diversity rankings. We can use the following equation to demonstrate the role of the diverse kernel,

$$\log P\left(S_u^{+k}\right) \propto \log\det\left(\mathbf{L}^{(u,k+n)}_{S_u^{+k}}\right) = \sum_{i\in S_u^{+k}} \log\left(\hat{y}^2_{(u,i)}\right) + \log\det\left(\mathbf{K}_{S_u^{+k}}\right). \tag{5.5}$$

The log-probability of S_u^{+k} is proportional to the log-determinant that can be further decomposed into two components: the aggregated relevance scores and the determinant of the sub-matrix of $\mathbf{K}$ indexed by S_u^{+k}. This is applicable when considering two target subsets drawn from different k-DPP distributions, each over their unique $k+n$ ground sets sampled from the same user's implicit feedback. As these two target subsets consist of positive items, we can assume their relevance scores to be similar. Therefore, in the comparison of their k-DPP probabilities, the decisive factor lies in the diversity factor. Looking at Equation 5.3, it becomes evident that the learning process for the diverse kernel is geared towards maximizing the determinant value of diverse subset. Therefore,

in a comparison scenario between two target subsets drawn from respective k-DPP distributions, the one exhibiting a broader category coverage tends to possess a higher k-DPP probability. Consequently, this effect of the *diversity ranking interpretation* influences the unified ranking.

The concrete set-level optimization criterion LkP is thus built upon the tailored k-DPP distribution as defined in Equation 5.4. A target subset characterized by a broader category coverage within the k-DPP is inclined to be assigned a higher probability than other monotonous target subsets, as shown in the k-DPP probability ranking part of Figure 5.1, *i.e.*, $P^k_{\mathbf{L}^u}\left(v_3^{(1,a)}, v_6^{(1,b)}, v_8^{(1,c)}\right) > P^k_{\mathbf{L}^u}\left(v_3^{(1,a)}, v_{10}^{(1,a)}, v_{11}^{(1,a)}\right)$. This ultimately enables item recommendation models to provide users with not only diverse but also relevant results.

It is important to note that the diverse kernel $\mathbf{K}$ is pre-trained and remains fixed throughout the process of maximizing LkP. Its role lies in balancing the probabilities of the drawn target subsets, ensuring a fair and diverse representation in the recommendations. The diversity among different subsets within the same k-DPP distribution is not needed to consider. This is because, regardless of the diversity level of all other k-subsets, it will not affect the unified ranking. In the learning process of the optimization criterion, the influence of diversity remains constant. The formulation of maximizing the relevance score of the target subset continually strengthens their ranking, ultimately leading to higher ranking compared to other subsets. However, across different k-DPP distributions, different target sets are all learned to enhance their relevance scores, thereby making the difference in diversity among these subsets a key determinant of their respective rankings.

To sum up, the relevance ranking interpretation signifies that a k-DPP inherently captures the implicit ranking comparison between the target subset and other k-subsets within a k-DPP distribution. On the other hand, the diversity ranking interpretation refers to the global balancing of relevance and diversity across different k-DPP distributions (over various $k+n$ ground sets), facilitated by the pre-learned diverse kernel. The relevance and diversity comparisons are respectively demonstrated through two ranking examples in experiments.

5.2.2.4 Optimization Approaches: LkP$_{PS}$ and LkP$_{NPS}$

To implement the implicit comparison of k-DPP probabilities between the target subset and all other subsets within a specific ground set, it is crucial to accurately compute the probability of the target subset, which requires proper normalization. This normalization constant enables us to associate the target subset and other k-subsets within a k-DPP distribution, thereby establishing implicit competition relationships, which is

formulated as,

$$Z_k = \sum_{|S'|=k} \det\left(\mathbf{L}_{S'}^{(u,k+n)}\right) = e_k\left(\lambda_1, \lambda_2, \ldots, \lambda_{(k+n)}\right). \tag{5.6}$$

Here $\lambda_1, \lambda_2, \ldots, \lambda_{(k+n)}$ are the eigenvalues of $\mathbf{L}^{(u,k+n)}$ [Kulesza et al., 2012; Gelfand, 1989]. The recursive algorithm given in Algorithm 1 is used to compute the kth elementary symmetric polynomial e_k on $\lambda_1, \lambda_2 \ldots, \lambda_{(k+n)}$, which runs in $O((k+n)k)$ time. Thus we can efficiently normalize a tailored k-DPP with $n+k$ ground set in $O\left((k+n)k\right)$ time to implicitly carry out the set-level ranking comparison.

We assign a common k and n for all users. As a shorthand, we will directly use $\mathbf{L}^u$ instead of $\mathbf{L}^{(u,k+n)}$ when the meaning is clear. Now we can formulate the learning objective to derive our optimization criterion for k-DPP probability-based set-level ranking,

$$\mathcal{L} = \prod_u \prod_{S_u^{+k} \in \mathcal{S}_u^+} P_{\mathbf{L}^u}^k\left(S_u^{+k}\right) = \sum_u \sum_{S_u^{+k} \in \mathcal{S}_u^+} \log\left(P_{\mathbf{L}^u}^k\left(S_u^{+k}\right)\right). \tag{5.7}$$

$\mathcal{S}_u^+$ denotes the collection of k-sized target sets of user u adopted in the training process. In practical experiments, it is not necessary to take all k-subsets of Y_u^+ into account, given the computation burden and partially redundant calculations. Instead we guarantee that each target item of user u is selected in a k-DPP at least once. In this way we ensure that the number of set-level training instances used in our experiments is not greater than the pointwise method or BPR optimization , making for a fair comparison.

The above log-likelihood function appropriately increases the rank of target subset, but the probability of unobserved item subsets is not explicitly integrated. Previous ranking optimization approaches, *e.g.*, BPR, SetRank, and Set2SetRank, usually integrate a ranking pair by providing an observed instance (item or set) with a randomly selected lower-ranked candidate. Unlike this common approach, we take the unobserved items into account in a probabilistic manner. We first formulate the probability of selecting k unobserved items of user u as a k-DPP,

$$P_{\mathbf{L^u}}^k(S_u^{-k}) = \frac{\det\left(\mathbf{L}_{S_u^{-k}}^u\right)}{\sum_{|S'|=k} \det\left(\mathbf{L}_{S'}^u\right)}. \tag{5.8}$$

S_u^{-k} is a negative k-set selected from a specific ground set.

Our intuition in considering the probability of S_u^{-k} is to widen the gap between targets and unobserved items in a k-DPP, *i.e.*, not only increasing the ranking of S_u^{+k} but decreasing that of S_u^{-k}. To achieve this, we attempt to model an integrated probability that the event that S_u^{+k} is drawn from the specific $k-$DPP, but S_u^{-k} explicitly

is not. Basic rules of probability give

$$\begin{aligned} P\left(S_u^{+k} \text{and not } S_u^{-k}\right) &= P\left(S_u^{+k}\right) P\left(\text{not } S_u^{-k} \middle| S_u^{+k}\right) \\ &\approx P\left(S_u^{+k}\right) P\left(\text{not } S_u^{-k}\right), \end{aligned} \tag{5.9}$$

where P is actually a k-DPP on the $k+n$ ground set and the final approximation is an independence assumption. Given $P\left(\text{not } S_u^{-k}\right) = 1 - P\left(S_u^{-k}\right)$, we finally obtain the logarithmic objective function,

$$\begin{aligned} \mathcal{L} &= \prod_{(S_u^{+k}, S_u^{-k}) \in \mathcal{S}_u} \left(P_{\mathbf{L}^u}^k\left(S_u^{+k}\right)\right)\left(1 - P_{\mathbf{L}^u}^k\left(S_u^{-k}\right)\right) \\ &= \sum_{(S_u^{+k}, S_u^{-k}) \in \mathcal{S}_u} \log\left(P_{\mathbf{L}^u}^k\left(S_u^{+k}\right)\right) + \log\left(1 - P_{\mathbf{L}^u}^k\left(S_u^{-k}\right)\right), \end{aligned} \tag{5.10}$$

which can be calculated by combining with Equation 5.4 and Equation 5.8. $\mathcal{S}_u$ represents the compilation of k-sized subsets from the above described ground set.

The exclusion of S_u^{-k} implies that a k-set with unobserved items is supposed to have a lower rank compared to other subsets (with cardinality k) that contain one or more targets. To avoid extra comparisons between unobserved items, *e.g.*, two k-sized subsets with different combinations of unobserved items, we will set $n = k$ for the $k+n$ ground set when explicitly considering exclusion probability. We find that this function of considering inclusion of S_u^{+k} and exclusion of S_u^{-k} has a similar formalization to that of the popular binary cross-entropy loss. However, our optimization criterion essentially shows distinct implication behind the formulation (Equation 5.10); specifically, our proposed formulation focuses on set-level comparison.

Overall, our optimization criterion focuses on learning the probabilities of k-DPPs on specific ground sets for each user to optimize diversity-aware personalized ranking. Two types of optimization approaches are proposed, LkP$_{PS}$ and LkP$_{NPS}$, wherein only the **p**ositive sub**s**et and both exclusion of **n**egative items (unobserved elements) and inclusion of **p**ositive sub**s**et are considered, respectively.

5.2.3 LkP Optimization

We provide the gradient computation to demonstrate how to maximize our optimization criterion LkP. To learn the parameters Θ with respect to the model, we can maximize the log-likelihood, derived by substituting Equation 5.4 into equation 5.7

Algorithm 1 Computing the elementary symmetric polynomials

Input: k, eigenvalues $\lambda_1, \lambda_2, \dots \lambda_{(k+n)}$
$e_0^m \leftarrow 1 \quad \forall\, m \in \{0, 1, 2, \dots, k+n\}$
$e_l^0 \leftarrow 0 \quad \forall\, l \in \{1, 2, \dots, k\}$
for $l = 1, 2, \dots k$ **do**
  for $m = 1, 2, \dots, k+n$ **do**
    $e_l^m \leftarrow e_l^{m-1} + \lambda_m e_{l-1}^{m-1}$
  end for
end for
Output: $e_k\left(\lambda_1, \lambda_2, \dots, \lambda_{(k+n)}\right) = e_k^{(k+n)}$

and subsequently applying a logarithmic operation,

$$\begin{aligned}\mathcal{L}(\Theta) = &\sum_{S \in D_S, S^+ \in S} \log\det\left(\mathbf{L}_{S^+}(\Theta)\right) \\ &- \sum_{S \in D_S, S' \in S} \log \sum_{|S'|=k} \det\left(\mathbf{L}_{S'}(\Theta)\right).\end{aligned} \tag{5.11}$$

For brevity, D_S represents the collection of training instances from entire users, and S is one of them from a specific user, *i.e.*, a $k+n$ ground set. One thing to note here is that multiple training instances can be extracted from a user. To further simplify our notations when the meaning is clear, we directly use $\mathbf{L}$ to denote the DPP kernel specific to a user training instance, rather than using the previously introduced $\mathbf{L}^{(u,k+n)}$; $\mathbf{L}_{S^+}$ represents the sub-matrix of $\mathbf{L}$ indexed by the target set S^+ (consisting of k targets), previously represented as S_u^{+k}, and S' denotes any one of k-sized subsets of the ground set.

Gradient-based methods (*e.g.*, gradient descent and stochastic gradient descent) provide attractive approaches in learning parameters Θ of DPP kernel because of their theoretical guarantees, but the gradient of the log-likelihood $\mathcal{L}(\Theta)$ is required. In the discrete DPP setting, this gradient can be computed straightforwardly.

$$\begin{aligned}\frac{\partial \mathcal{L}(\Theta)}{\partial\, \Theta} = &\sum_{S \in D_S,\ S^+ \in S} \operatorname{tr}\left(\mathbf{L}_{S^+}(\Theta)^{-1} \frac{\partial \mathbf{L}_{S^+}(\Theta)}{\partial \Theta}\right) \\ &- \sum_{S \in D_S,\ S' \in S} \sum_{|S'|=k} w_{S'} \operatorname{tr}\left(\mathbf{L}_{S'}(\Theta)^{-1} \frac{\partial \mathbf{L}_{S'}(\Theta)}{\partial \Theta}\right),\end{aligned} \tag{5.12}$$

where $w_{S'}$ represents the importance (normalized probability) of the k-sized subset S' in the ground set.

As we employ quality *vs.* diversity decomposition to balance DPP kernel, the entry

of kernel $\mathbf{L}$ can be written as,

$$\mathbf{L}_{ij} = \exp\left(\mathbf{e}_u \mathbf{e}_i^\top\right) \mathbf{K}_{ij} \exp\left(\mathbf{e}_u \mathbf{e}_j^\top\right), \tag{5.13}$$

where $\mathbf{K}$ is the pre-learned diverse kernel. The quality value $\exp\left(\mathbf{e}_u \mathbf{e}_i^\top\right)$ corresponds the relevance of item i to user u, calculated based on the user and item embeddings, which are exactly the parameters that we need to learn.

Specifically, we can compute the gradient of DPP kernel with respect to the parameter $e_u^{(d)}$ (d is the dimension number of user u's feature vector) as

$$\begin{aligned} \frac{\partial \mathbf{L}_{ij}}{\partial e_u^{(d)}} &= \exp\left(\mathbf{e}_u \mathbf{e}_i^\top\right) \mathbf{K}_{ij} \exp\left(\mathbf{e}_u \mathbf{e}_j^\top\right) (e_i^{(d)} + e_j^{(d)}). \\ &= \mathbf{L}_{ij} \left(e_i^{(d)} + e_j^{(d)}\right). \end{aligned} \tag{5.14}$$

Denote $\mathbf{R}_{ij} = \mathbf{L}_{ij} \left(e_i^d + e_j^d\right)$, we can obtain

$$\begin{aligned} \frac{d\mathcal{L}(\Theta)}{de_u^d} = &\sum_{S \in D_S,\ S^+ \in S} \operatorname{tr}\left(\mathbf{L}_{S^+}(\Theta)^{-1} \mathbf{R}_{S^+}^{(d)}\right) \\ &- \sum_{S \in D_S,\ S' \in S} \sum_{|S'|=k} w_{S'} \operatorname{tr}\left(\mathbf{L}_{S'}(\Theta)^{-1} \mathbf{R}_{S'}^{(d)}\right), \end{aligned} \tag{5.15}$$

Similarly, we can compute the gradient of DPP kernel entry $\mathbf{L}_{ij}$ with respect to the representation of an item (*i.e.*, $\mathbf{e}_i$ for item i) as

$$\frac{\partial \mathbf{L}_{ij}}{\partial e_i^{(d)}} = = \mathbf{L}_{ij} e_u^{(d)}. \tag{5.16}$$

Let $\mathbf{G}_{ij}^{(d)} = \mathbf{L}_{ij}^{(S)} e_u^d$, then

$$\begin{aligned} \frac{d\mathcal{L}(\Theta)}{de_i^d} = &\sum_{S \in D_S,\ S^+ \in S} \operatorname{tr}\left(\mathbf{L}_{S^+}^{(S)}(\Theta)^{-1} \mathbf{G}_{S^+}^{(d)}\right) \\ &- \sum_{S \in D_S,\ S' \in S} \sum_{|S'|=k} w_{S'} \operatorname{tr}\left(\mathbf{L}_{S'}^{(S)}(\Theta)^{-1} \mathbf{G}_{S'}^{(d)}\right), \end{aligned} \tag{5.17}$$

From above formulations, it can be demonstrated that our optimization criterion based on set-level DPP probability comparison is differentiable. We therefore can apply the stochastic gradient descent based algorithms with learning rate η to optimize the model parameters based on sampled $(u, S) \in D_S$. In our experiments, we employ Adam and grid search η for our approaches and baselines. The gradient calculation process for Equation 5.10 is not presented, as Equation 5.10 and Equation 5.7 share

Table 5.2: Statistics of the datasets (Beauty, CDs, and Anime).

Dataset	#Users	#Items	#Interactions	#Categories
Beauty	52.0k	57.2k	0.4M	213
CDs	43.1k	35.5k	0.7M	340
Anime	73.5k	12.2k	1.0M	43

the same basic components and parameters.

5.3 Experiments

To comprehensively demonstrate the superiority of our approaches, three levels of evaluations and comparisons are conducted: *i*) six LkP based variants are proposed and evaluated; *ii*) two types of implementation of LkP for recommendation are achieved based on Matrix Factorization (MF) and Graph Convolutional Network (GCN), and are compared with state-of-the-art baselines; *iii*) through verifying the improvement after applying our approaches on different seminal CF methods (reworked models) over their original framework, the generality and availability can be validated.

5.3.1 Settings

5.3.1.1 Datasets.

For evaluating our proposed k-DPP based personalized ranking approaches (LkP$_{PS}$ and LkP$_{NPS}$), we select three common and real-world datasets as appropriate, and whose category numbers and matrix densities of implicit interactions differ significantly. Amazon-review contains a series of datasets [He and McAuley, 2016b], comprised of a large corpus of product ratings. We select *Beauty* and *CDs* products from the collection, which have 213 and 340 categories, respectively. *Anime* dataset consists of user ratings to anime from `myanimelist.net`. All items in Anime are grouped into 43 categories. We transform these datasets into implicit data, where each entry is marked as 1/0, depending on whether the ratings are 5. We filter out long-tailed users and items with fewer than 10 interactions for all datasets. For each user, we randomly select 20% of the rated items as ground truth for testing, and 70% and 10% ratings constitute the training and validation set, respectively. The statistics of the three datasets are summarized in Table 5.2. We can note that Beauty and CDs have more plentiful product categories and lower densities compared to Anime.

5.3.1.2 Baselines and Metrics.

We place all compared methods into three groups according to different levels of comparisons.

Six variants derived from k-DPP probability are given, *i.e.*, PR, PS, NPR, NPS, PSE, and NPSE. In these proposed methods, P means the usage of the target set inclusion probability, and N denotes the consideration of the exclusion of unobserved items. S represents a method of sampling for a training instance, *i.e*, selecting of k observed items in the order they occurred using a sliding window, along with n randomly selected unobserved items. R signifies random selection, *i.e.*, randomly selecting $k+n$ items (k targets and n un-interacted items) from user's $1/0$ feedback. Inspired by the original definition of entries of DPP diversity kernel (*i.e.*, measurements of similarity), we propose another type of formulation for diversity kernel $\mathbf{K}$. It directly regards any two items' embeddings as their feature vectors, which are used to calculate the similarity of item pairs following the calculation manner of Gaussian kernel [Affandi et al., 2014] (*i.e.*, Equation 3.8). E denotes this new formulation of diversity factor, instead of pre-learned $\mathbf{K}$. In the E setting, the optimization process incorporates the diversity factor directly. The intention is to learn the embeddings (parameters) in a direction that promotes dissimilarity. For example, NPSE represents a method that uses Equation 5.10 (exclusion of **n**egative items and inclusion of **p**ositive items) as the objective function trained using **s**equentially selected target sets, and calculates the diverse kernel based on trainable **e**mbeddings.

Four state-of-the-art and popular objective functions are included: binary cross-entropy (BCE) [He et al., 2017b], BPR [Rendle et al., 2012], SetRank [Wang et al., 2020], and S2SRank (*i.e.*, Set2SetRank) [Chen et al., 2021]. All optimization baselines and our approaches are deployed in scenarios of MF- and GCN-based collaborative filtering (CF) item recommendation models for a thorough evaluation. Our choice to implement all ranking optimization approaches using basic MF- and GCN-based CF models from a thoughtful rationale. As we mentioned in Section 5.2.1, MF is the most common CF technique and is widely adopted as a baseline in recommendation research due to its simplicity and effectiveness in capturing latent factors and item-user interactions. GCN [He et al., 2020; Wang et al., 2019c; Chen et al., 2022] has gained substantial prominence in the field of collaborative deep learning due to their ability to leverage graph structures and capture intricate item-item relationships. By selecting both MF- and GCN-based CF models for validation, we ensure a comprehensive evaluation across a spectrum of traditional and state-of-the-art approaches in item recommendation.

Two distinct seminal CF models in item recommendation field, *i.e.*, GCMC [Berg

et al., 2017] and NeuMF [He et al., 2017b], are selected to evaluate the generality and applicability of our approaches. To take different types of CF methods into account as comprehensively as possible, we select two seminal and distinct CF models as baselines in this part of evaluation. GCMC [Berg et al., 2017] is a seminal graph-based auto-encoder work for recommender systems, which inspired many graph neural networks based studies in the CF field, *e.g.*, NGCF [He et al., 2017b], lightGCN [He et al., 2020], and KGAT [Wang et al., 2019b]. It applies negative log likelihood as loss, and a probability distribution over possible rating levels by a softmax function is produced, which is distinct from commonly used relevance calculation methods (*i.e.*, inner product calculation and nonlinear neural networks introduced in Section 5.2.2). NeuMF [He et al., 2017b] is a classic and comprehensive model that combines GMF with MLP. The relevance is predicted through classifying the interaction from a user to a specific item through nonlinear neural networks. The model is optimized based on classification-aware binary cross-entropy loss.

Two types of accuracy related metrics, *i.e.*, NDCG@N (Nd) and Recall@N (Re), the popular and intuitive diversity metric — *Category Coverage* (CC) [Wu et al., 2019b; Liu et al., 2022; Puthiya Parambath et al., 2016], and a harmonic F-score (F) [Cheng et al., 2017; Liang and Qian, 2021] metric between quality (accuracy) and diversity are adopted. As we focus on implicit feedback and there is no explicit feature under consideration, ILD diversity metric [Puthiya Parambath et al., 2016; Zhang and Hurley, 2008] is not adopted in evaluation. We set $\mathrm{N} \in \{5, 20\}$ for comprehensive evaluation.

5.3.1.3 Implementations.

For the above baselines, we have carefully explored the corresponding parameters, *i.e.*, the number of dimensions, the learning rate and regularization parameters. For a fair comparison we set the trainable embedding vectors of all compared models in this work to the same size (64), and report the best results of each model by tuning the hyperparameters on a validation set. For the seminal baselines, the reworked models and original approaches both use the default parameters provided by their official source codes. A prominent variant of stochastic gradient descent method, Adam, is applied.

5.3.2 Comparisons

The overall performance comparison is reported in Table 5.3. All results are obtained by implementing GCN-based CF model, under the setting of $k = n = 5$. To maintain the generality of LkP, we employ the basic GCN framework. The message propagation part refers to NGCF [Wang et al., 2019c] that learns representations from high-order connectivities. The bold value represents the best result of a metric on a dataset. The

Table 5.3: Performance comparison between LkP and state-of-the-art objective functions, deployed on deep neural network. The bold numbers denote the best performance among all variants on a specific dataset under a certain evaluation criterion. Conversely, the worst performances are underlined. The superscript + and − symbols are used to respectively indicate the best and worst results of the baseline method.

Dataset	Method		Re@5	Re@20	Nd@5	Nd@20	CC@5	CC@20	F@5	F@20
Beauty	LkP	PR	0.0788	0.1754	0.0808	0.1209	0.0579	0.1312	0.0671	0.1391
		PS	0.0806	0.1841	0.0794	0.1213	0.0575	0.1360	0.0669	0.1439
		NPR	0.0820	0.1756	0.0801	0.1184	0.0588	0.1354	0.0682	0.1410
		NPS	**0.0868**	**0.1874**	**0.0878**	**0.1283**	0.0578	0.1334	**0.0696**	**0.1446**
		PSE	0.0726	0.1717	0.0716	0.1121	0.0589	0.1377	0.0648	0.1398
		NPSE	0.0684	0.1596	0.0695	0.1068	**0.0604**	**0.1446**	0.0644	0.1387
	Others	BPR	0.0694	0.1544^{-}	0.0704	0.1049^{-}	0.0569	0.1301	0.0627	0.1299^{-}
		BCE	0.0672^{-}	0.1700	0.0656	0.1070	0.0568	0.1336^{+}	0.0612^{-}	0.1360
		SetRank	0.0775	0.1746	0.0772	0.1166	0.0574^{+}	0.1278^{-}	0.0659^{+}	0.1361
		S2SRank	0.0783^{+}	0.1752^{+}	0.0776^{+}	0.1172^{+}	0.0566^{-}	0.1303	0.0656	0.1378^{+}
	max vs. min (%)		29.17	21.37	33.84	22.31	6.71	13.14	13.59	11.34
	max vs. max (%)		10.86	6.96	13.14	9.47	5.23	8.23	5.54	4.94
CDs	LkP	PR	0.0197	0.0520	0.0202	0.0363	**0.0678**	0.1564	0.0308	0. 0689
		PS	**0.0224**	0.0610	**0.0239**	0.0403	0.0648	0.1447	**0.0341**	0.0750
		NPR	0.0218	0.0598	0.0225	0.0380	0.0653	0.1503	0.0331	0.0738
		NPS	0.0221	**0.0627**	0.0234	**0.0410**	0.0643	0.1455	0.0336	**0.0765**
		PSE	0.0165	0.0394	0.0183	0.0299	0.0677	**0.1581**	0.0277	0.0568
		NPSE	0.0147	0.0380	0.0178	0.0283	0.0673	0.1574	0.0262	0.0548
	Others	BPR	0.0168	0.0441	0.0192	0.0291	0.0656	0.1434	0.0282	0.0577
		BCE	0.0152^{-}	0.0419^{-}	0.0180^{-}	0.0282^{-}	0.0667^{+}	0.1448^{+}	0.0266^{-}	0.0564^{-}
		SetRank	0.0208^{+}	0.0517	0.0215^{+}	0.0344	0.0605^{-}	0.1354^{-}	0.0313^{+}	0.0653
		S2SRank	0.0197	0.0530^{+}	0.0204	0.0358^{+}	0.0652	0.1429	0.0307	0.0677^{+}
	max vs. min (%)		47.37	49.64	32.78	45.39	12.07	16.77	28.32	35.47
	max vs. max (%)		7.69	18.30	11.16	14.54	1.65	9.19	8.84	12.85
Anime	LkP	PR	0.0863	0.2231	0.1421	0.1836	0.3319	0.5837	0.1699	0.3016
		PS	**0.0975**	0.2442	0.1560	0.2043	0.3359	0.5825	**0.1841**	0.3238
		NPR	0.0920	0.2340	0.1490	0.1932	0.3300	0.5657	0.1765	0.3101
		NPS	0.0974	**0.2447**	**0.1579**	**0.2059**	0.3293	0.5758	0.1840	**0.3239**
		PSE	0.0895	0.2388	0.1494	0.1966	0.3353	0.5949	0.1761	0.3188
		NPSE	0.0784	0.2089	0.1328	0.1762	**0.3381**	**0.5953**	0.1609	0.2910
	Others	BPR	0.0879	0.2184^{-}	0.1443	0.1825^{-}	0.3339	0.5890^{+}	0.1723	0.2991^{-}
		BCE	0.0863^{-}	0.2231	0.1421^{-}	0.1836	0.3320	0.5737	0.1699^{-}	0.3003
		SetRank	0.0890	0.2269	0.1467	0.1884	0.3485^{+}	0.5735^{-}	0.1761^{+}	0.3049
		S2SRank	0.0902^{+}	0.2295^{+}	0.1477^{+}	0.1892^{+}	0.3254^{-}	0.5793	0.1742	0.3076^{+}
	max vs. min (%)		12.98	12.04	11.12	12.82	3.90	3.80	8.30	8.28
	max vs. max (%)		8.09	6.62	6.91	8.83	-2.98	1.07	4.49	5.32

annotation $\underset{\sim\sim\sim}{}$ denotes the worst performance from LkP variants. The superscripts $^+$ and $^-$ are used to label the best and worst performances out of the baselines. Two types of improvements are calculated, the best performance of our variants *vs.* the best (*max*) and worst (*min*) performances from baselines. Three groups of important observations are made:
(i) *comparison between six* LkP *variants.*

- Comparing R and S (two types of training instance construction), *i.e.*, PR *vs.* PS and NPR *vs.* NPS, shows that selecting k targets in a certain sequence (S) outperforms the random selection mode (R) *w.r.t.* quality metrics (NDCG and Recall) on three datasets. This may be caused by the dependency among items in the target subset, as the items in sequence have clearer correlations (*e.g.*, similar attributes, or the same category) compared to randomly constructed training targets. These correlations naturally captured by k-DPP in S training instances contribute to collaborative filtering. With regard to diversity results, R mode achieves better performance than S, especially on Beauty and CDs datasets. This is expected, as randomly selected target subsets are endowed with more diverse correlations than is the case for S. In addition, with the same size of training instances, LkP using R needs more epochs to achieve convergence compared to S, as R makes it hard to learn the k-DPP distribution due to increased randomness. These results verify that LkP compares (relevance and diversity) rankings at set level and considers correlations within items.

- Among the two diversity factor construction methods, *i.e.*, the pre-learned kernel $\mathbf{K}$ using Equation 5.3 (default type) and the item embeddings' similarity kernel (E type), the default type approaches (*i.e.*, PR, PS, NPR, and NPS) achieve better relevance performance than E type methods (*i.e.*, PSE and NPSE). However, in general the diversity performance of these two types of kernels is exactly the opposite, especially in the cases of CC@10 and CC@20. The main reason behind these results is that the diversity is provided by the pre-learned diverse kernel in the default type, making it relatively easy to balance the quality and diversity. However, E type considers the diversity factor based on the intuition of expanding intra-list distance (*i.e.*, ILD) and tries to directly learn embeddings towards two directions (quality and diversity) by regarding embeddings as feature representations of items. The weak overall performance (F) of E type indicates that explicitly learning item embeddings with two objectives finally places the recommendation system in a dilemma, and degrades performance. The noteworthy diversity results of E type further indicate that the LkP facilitates capturing diverse correlations not only from pre-widened category coverage (default) but also from learned intra-list

distance (E). In addition, the quality results of E type are comparable to those of the baselines (even superior to the baselines on Beauty and CDs) demonstrating the superiority of our approaches. The comparison between R and S above have shown that S mode is more suitable for LkP, which is the reason why only the combination of S and E are listed in Table 5.3.

- Two main LkP based approaches are compared, *i.e*, considering only the probability of selecting a target subset (P) and considering both the inclusion of targets and exclusion of unobserved items (NP), which are based on Equation 5.7 and Equation 5.10, respectively. We only compare PS with NPS here, as other variants, *e.g.*, R and E, are not effective enough. In general, NPS performs better than PS in both quality and diversity metrics. This suggests that explicitly taking the exclusion probability of unobserved items into account enriches the relevance ranking of LkP, and allows the model to capture more relationships between relevance and diversity, thereby preventing the recommendation model from suggesting monotonous results. In some cases, *e.g.*, Re@5 and Nd@5 on CDs and Re@5 on Anime, PS beats NPS. This may occur because only focusing on selecting preferred items as a k-DPP highlights the relevance of potential positive items in the top rankings without the distraction of unobserved items. The notable performance of PS and NPS shows the prospect of LkP.

- We can find that NPSE is generally weaker than PSE in relevance metrics, which is contrary to the previous comparison between P and NP. This may arise from the usage of the E type diversity factor, as adding the probabilistic learning of unobserved items in the objective function (Equation 5.10) makes the embedding learning process that balances quality and diversity less congruent, compared to only drawing targets (Equation 5.7).

(*ii*) *comparison between our approaches and baselines.*

- A significant improvement over pointwise loss (BCE) and pairwise ranking (BPR) achieved by our main LkP approaches (PS and NPS) is apparent in quality metrics, exceeding 20% on CDs and Beauty and around 10% on Anime. This improvement trend also exists between LkP and other set-level ranking methods (SetRank and S2SRank), *i.e.*, around 10% on CDs and Beauty and more than 5% on Anime. This indicates that LkP approaches have clear advantages in dealing with more sparse interaction matrices (CDs and Beauty) over baselines, as Anime has the highest density matrix (as calculated from Table 5.2). The superiority of our approaches is verified with a significant boost even in comparison with the recently proposed set ranking optimization methods.

Table 5.4: Performance comparison between LkP and state-of-the-art ranking models, deployed on matrix factorization.

Dataset	Method	Re@5	Re@20	Nd@5	Nd@20	CC@5	CC@20	F@5	F@20
Beauty	LkP$_{PS}$-MF	0.0785	0.1787	0.0765	0.1176	**0.0569**	**0.1346**	**0.0656**	0.1409
	LkP$_{NPS}$-MF	**0.0817**	**0.1849**	**0.0768**	**0.1190**	0.0567	0.1338	0.0661	**0.1422**
	BPR-MF	0.0690^{-}	0.1584^{-}	0.0701^{-}	0.1012^{-}	0.0574^{+}	0.1304^{+}	0.0629^{-}	0.1301^{-}
	SetRank-MF	0.0761^{+}	0.1689	0.0753^{+}	0.1106	0.0567	0.1292	0.0648^{+}	0.1343^{+}
	S2SRank-MF	0.0757	0.1715^{+}	0.0750	0.1115^{+}	0.0559^{-}	0.1254^{-}	0.0642	0.1330
	max vs. min	18.41	16.73	9.56	17.59	1.79	7.26	5.11	9.68
	max vs. max	7.36	7.81	1.99	6.73	-0.87	3.14	1.96	6.28
CDs	LkP$_{PS}$-MF	**0.0201**	**0.0565**	0.0197	0.0340	**0.0662**	**0.1587**	**0.0306**	**0.0704**
	LkP$_{NPS}$-MF	0.0192	0.0560	**0.0200**	**0.0351**	0.0660	0.1516	0.0302	0.0701
	BPR-MF	0.0153^{-}	0.0326^{-}	0.0173^{-}	0.0253^{-}	0.0661^{+}	0.1494	0.0262^{-}	0.0533^{-}
	SetRank-MF	0.0176	0.0506^{+}	0.0185	0.0305	0.0650^{-}	0.1452^{-}	0.0283	0.0634^{+}
	S2SRank-MF	0.0183^{+}	0.0475	0.0189^{+}	0.0312^{+}	0.0657	0.1533^{+}	0.0290^{+}	0.0626
	max vs. min	31.37	42.68	15.61	38.74	1.85	9.30	17.02	32.07
	max vs. max	9.84	11.66	5.82	1.25	0.15	3.52	5.55	10.08
Anime	LkP$_{PS}$-MF	**0.0887**	0.2427	**0.1450**	0.1942	0.3012	0.5504	**0.1684**	0.3128
	LkP$_{NPS}$-MF	0.0875	**0.2438**	0.1425	**0.1950**	**0.3020**	**0.5553**	0.1666	**0.3145**
	BPR-MF	0.0791^{-}	0.2157^{-}	0.1306^{-}	0.1720^{-}	0.3143^{+}	0.5453^{-}	0.1554^{-}	0.2860^{-}
	SetRank-MF	0.0833^{+}	0.2245	0.1358^{+}	0.1848	0.3101	0.5501^{+}	0.1619^{+}	0.2983
	S2SRank-MF	0.0827	0.2268^{+}	0.1349	0.1850^{+}	0.3073^{-}	0.5461	0.1608	0.2990^{+}
	max vs. min	12.14	13.03	11.03	13.37	0.57	1.83	8.33	9.97
	max vs. max	6.48	7.50	6.77	5.41	-1.88	0.95	4.00	5.18

- Regarding diversity evaluation, PS and NPS can provide more diversified suggestions for users than baselines in most cases. This means that the proposed approaches based on k-DPP really consider diversity in the k-DPP learning process.

- In the comparison of trade-off metric, our approaches outperform baselines on three datasets at different Top-N metrics (5, 20). That is, in line with their inherent modeling of both quality and diversity, k-DPP based optimization approaches are better at balancing relevance and diversity, compared to common quality *vs.* diversity trade-off methods that sacrifice quality for diversity, *e.g.*, SetRank on Beauty and BCE on CDs.

(*iii*) *comparison between baselines.*

- In most cases, BPR is more powerful than BCE *w.r.t.* Top-5 related quality metrics. This suggests that the pairwise ranking BPR enables GCN based CF models to assign user-interested items with high rankings.

- The *min* results mainly occur in BPR and BCE, indicating the importance of considering the correlations among items.

To further demonstrate the effectiveness of LkP approaches, we conduct more experiments on the basic MF implementation (*i.e.*, directly optimizing the optimization criterion). According to the comparison in Table 5.3, only the two main and most noteworthy variants (PS and NP) of LkP are further compared, denoted by LkP$_{PS}$ and LkP$_{NPS}$. Following previous CF studies [Wang et al., 2020; Chen et al., 2021; Wan et al., 2022], only the ranking optimization approaches (*i.e.*, BPR, SetRank, S2SRank) are compared in the implementation of MF. The corresponding comparison is presented in Table 5.4. All experiments are conducted directly following basic MF implementation. The bold values and superscripts denote the same meanings as those in Table 5.3.

The main findings are: (i) NPS type of LkP is generally more powerful than PS type *w.r.t.* quality evaluation metrics, which reinforces the meaning of taking the items that are of less interest to users in ranking comparisons. A trend similar to the comparison of Table 5.3 can be found here. That is, LkP$_{PS}$ beats LkP$_{NPS}$ when evaluating the relevance of the higher ranking items (*e.g.*, Top-5); (ii) In some cases, BPR achieves the best diversity performance but the worst relevance results among the compared methods, *e.g.*, CC@5 on Beauty and Anime. This might be caused by the application of basic MF, as simple MF endows embeddings learned from BPR (without considering correlations) with more randomness compared to other set-level ranking optimization approaches, and therefore diversity is promoted but relevance is sacrificed; (iii) Compared to SetRank and S2SRank that consider the correlations of item sets, around 5% improvement is achieved on overall performance (F). This again verifies the effectiveness of our approaches even deploying them on basic and simple MF. Aligning with analysis from GCN and MF implementations, additional and extensive evidence can be provided to support the superiority of our approaches.

To evaluate the adaptability and generality of k-DPP based approaches for the item recommendation area, we deploy two main types of LkP (PS and NPS) on seminal CF-based item recommendation models (*i.e.*, above mentioned GCMC and NeuMF) by replacing their original objective functions. If the reworked models that apply LkP obtain better performance than the original frameworks, we can validate two advantages of this work: (i) LkP can be adaptively applied to existing CF models as an objective function, *i.e.*, can be generalized to distinct recommendation models with different relevance measurements; (ii) the boosted performance is another evidence to show the superiority of LkP compared to previously used objective functions.

The corresponding comparison is shown in Table 5.5, where the methods with subscripts $_{PS}$ and $_{NPS}$ denote the reworked models by LkP$_{PS}$ and LkP$_{NPS}$, respectively. The bold value is the best among three methods (a baseline and two reworked models on a dataset), since we only consider the improvement achieved by reworked models

Table 5.5: Performance comparison between strong baselines and their k-DPP reworked counterparts.

Dataset	Method	Re@5	Re@20	Nd@5	Nd@20	CC@5	CC@20	F@5	F@20
Beauty	GCMC	0.0767	0.1609	0.0790	0.1215	0.0590	0.1320	0.0671	0.1364
	GCMC$_{PS}$	**0.0794**	0.1655	0.0813	0.1307	0.0582	0.1356	0.0675	0.1416
	GCMC$_{NPS}$	0.0792	**0.1706**	**0.0816**	**0.1334**	**0.0592**	**0.1371**	**0.0685**	**0.1442**
	Improv (%)	3.52	6.03	3.29	9.79	1.02	3.86	5.26	5.66
	NeuMF	0.0781	0.1689	0.0804	0.1163	0.0557	0.1273	0.0654	0.1345
	NeuMF$_{PS}$	0.0790	0.1724	**0.0823**	0.1185	**0.0562**	0.1285	**0.0662**	0.1365
	NeuMF$_{NPS}$	**0.0793**	**0.1750**	0.0811	**0.1190**	0.0560	**0.1286**	0.0660	**0.1372**
	Improv (%)	1.54	3.61	2.36	2.32	0.90	1.02	1.25	1.98
CDs	GCMC	0.0166	0.0460	0.0253	0.0434	0.0535	0.1254	0.0301	0.0659
	GCMC$_{PS}$	0.0204	0.0523	0.0278	0.0462	0.0548	0.1311	0.0335	0.0716
	GCMC$_{NPS}$	**0.0216**	**0.0529**	**0.0281**	**0.0465**	**0.0553**	**0.1315**	**0.0343**	**0.0721**
	Improv (%)	30.12	15.00	11.07	7.14	3.36	4.86	13.89	9.45
	NeuMF	0.0227	0.0521	0.0249	0.0407	0.0601	0.1302	0.0341	0.0684
	NeuMF$_{PS}$	**0.0254**	0.0516	0.0263	0.0419	0.0598	0.1313	0.0361	0.0690
	NeuMF$_{NPS}$	0.0252	**0.0527**	**0.0267**	**0.0420**	**0.0604**	**0.1320**	**0.0363**	**0.0697**
	Improv (%)	11.89	1.15	7.23	3.19	0.50	1.38	6.47	1.87
Anime	GCMC	0.0883	0.2199	0.1521	0.2150	0.3508	0.5749	0.1790	0.3155
	GCMC$_{PS}$	0.0910	0.2315	**0.1579**	0.2216	0.3496	0.5760	**0.1836**	0.3252
	GCMC$_{NPS}$	**0.0914**	**0.2326**	0.1567	**0.2232**	**0.3513**	**0.5765**	0.1834	**0.3267**
	Improv (%)	3.51	5.78	3.81	3.81	0.14	0.28	2.52	3.52
	NeuMF	0.0710	0.1985	0.1142	0.1576	0.3829	0.6099	0.1491	0.2756
	NeuMF$_{PS}$	0.0750	0.2027	0.1190	0.1611	0.3836	0.6106	0.1548	0.2803
	NeuMF$_{NPS}$	**0.0757**	**0.2042**	**0.1198**	**0.1621**	**0.3838**	**0.6107**	**0.1559**	**0.2818**
	Improv (%)	6.62	2.87	4.90	2.86	0.39	0.13	4.51	2.23

over the original baseline, the comparison between different baselines is not needed. *Improv*(%) shows how much LkP (the better approach out of PS and NPS) improves over the original baseline. We can find that (*i*) In most cases, obvious improvements (particularly on CDs dataset) are obtained by the reworked models compared to original baselines on both quality and diversity evaluation metrics; (*ii*) In the comparison between LkP$_{PS}$ and LkP$_{NPS}$, NPS type provides better suggestions (more relevant and diversified) for users. This finding is consistent with the results of Table 5.3 and Table 5.4; (*iii*) Overall, reworked models on GCMC achieve more significant improvements than that of NeuMF based models. The possible explanation might be that the original NeuMF has achieved outstanding performance compared to other baselines (listed in Table 5.3 and Table 5.4), which means that there is not much room left for improvement. On the other hand, this also indicates that LkP can boost the performance of existing models that are already well established, as the k-DPP based approaches regard the items in a subset as a whole, whose ranking is comprehensively compared and in which the correlations among items are well captured.

Table 5.3-Table 5.5 have verified the superiority of LkP from multiple aspects. We now further discuss the attributes of our approaches. As mentioned in Section 5.2.2, the

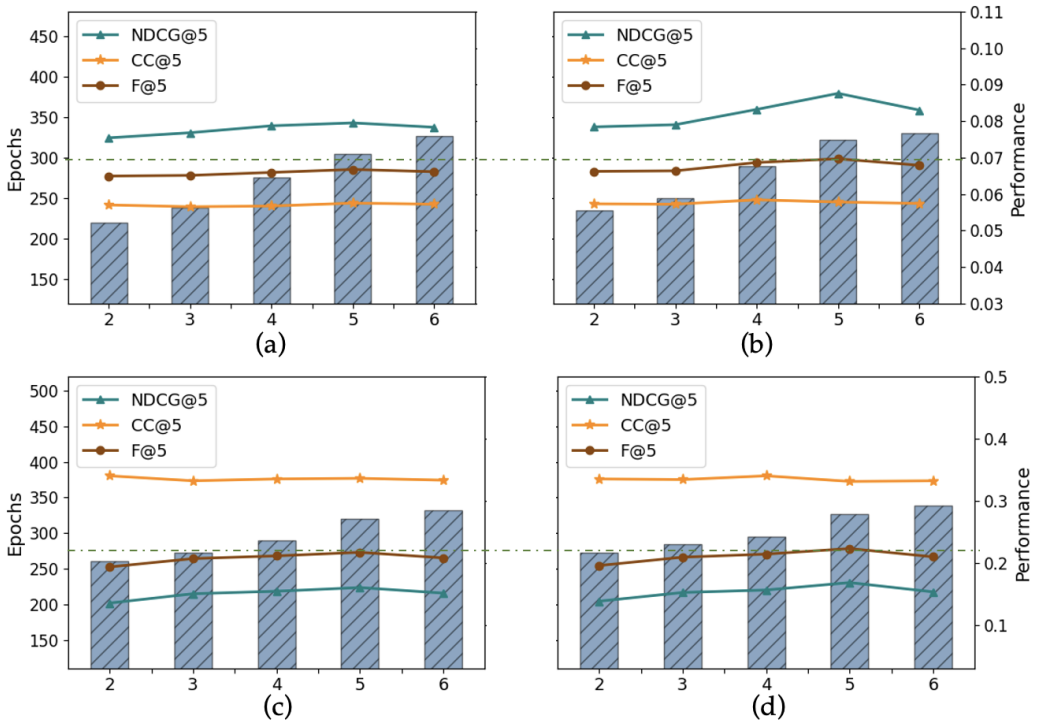
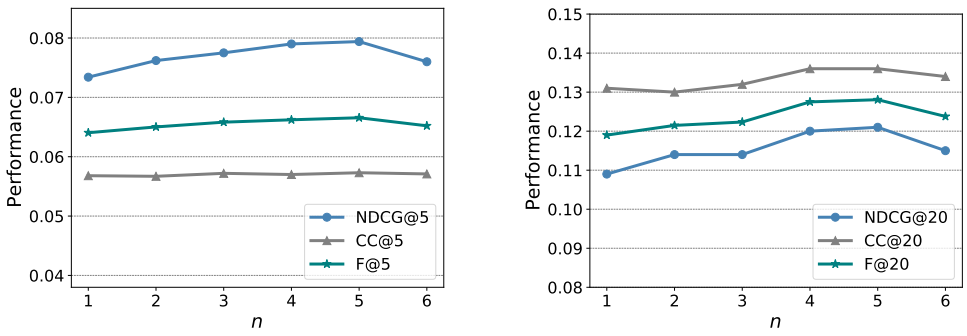


Figure 5.2: Performance of LkP$_{PS}$ and LkP$_{NPS}$ at different k.

tailored ground set setting plays a vital role in maintaining the *ranking interpretation*. All experiments in Table 5.3-Table 5.5 are conducted under the same setting of $k = 5$ and $n = 5$. We therefore perform additional experiments by fixing all experimental parameters except k and n.

We use Figure 5.2 to illustrate the effect of k, *i.e.*, number of observed items in the specific ground set, for CF recommendations. Figure (a) and (b) present the trends of LkP$_{PS}$ and LkP$_{NPS}$ on Beauty dataset. n is set to equal k in the corresponding experiments. LkP$_{PS}$ and LkP$_{NPS}$ results on Anime are shown in Figure (c) and (d). Besides the performance of three evaluation metrics, the epochs that are needed by LkP$_{PS}$ or LkP$_{NPS}$ to achieve the best performance under each setting are also presented. The dotted line on sub-figures is used to indicate the best trade-off performance for comparison. As LkP regards multiple items as a whole and tries to capture correlations among them, we therefore only perform experiments on $k > 1$. The slight decrease of CC results when $k > 4$ might be caused by the correlations among items, as k-DPP enjoys more correlations for recommendation relevance but lowers the weight of diversity. The quality performance (NDCG@5) clearly increases at first when $k < 5$, but a downward trend happens when $k > 5$. This suggests that if excessive correlations among items are considered, the final performance will be negatively affected. As for the epochs, an obvious trend, *i.e.*, an increase with k, can be found, which is straight-forward to



(a) Top-5 performance (b) Top-20 performance

Figure 5.3: Performance of LkP$_{PS}$ at different n.

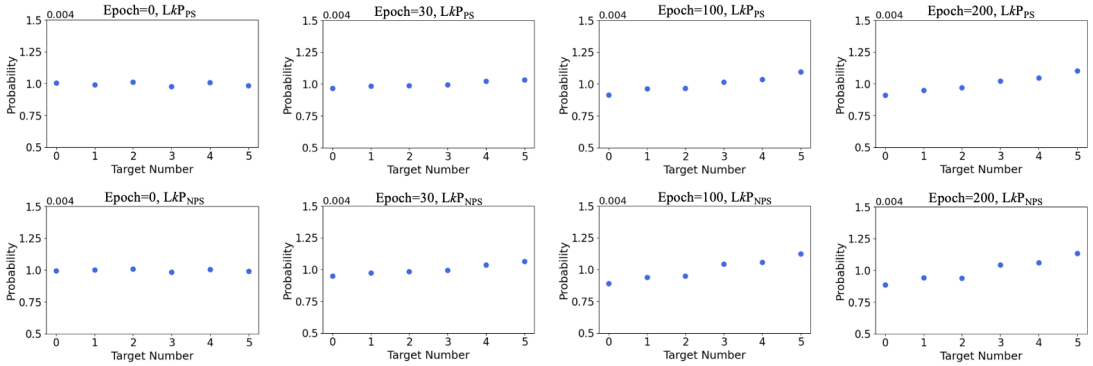


Figure 5.4: Examples of subset probability ranking of LkP$_{PS}$ and LkP$_{NPS}$ at different epochs on Anime.

explain as k-DPP needing more time to learn the more complex distribution.

Figure 5.3 shows LkP$_{PS}$ performance (Top-5 and Top-20) on Beauty with different numbers of unobserved items (*i.e.*, n). We find that all three metrics at different rankings show a similar trend. That is, they smoothly increase at first and then decrease. These results mean that the moderate size of n provides a sufficient set-level ranking comparison for learning k-DPP probabilities. However, when redundant comparisons are provided (*e.g.*, $n > 5$), it becomes difficult for k-DPP to capture effective correlations. As mentioned in Section 5.2.2, we set $k = n$ and the number of targets in the ground set equals to k for LkP$_{NPS}$ and LkP$_{PS}$ to keep the ranking interpretation. External experiments in the cases of $n > k$ on NPS and the number of targets (6 and 7) $> k$ (5) on PS have been performed. However, the worse performance compared to the presented cases suggests the importance of maintaining the ranking interpretation. Experiments in Figure 5.2 and Figure 5.3 are conducted on GCN implementation. Similar conclusions are obtained from experiments deployed on MF.

To diagrammatically present the ranking interpretation, we use real examples to

show the probability ranking of k-DPP subsets. We randomly select 100 training instances ($k+n$ ground set where $k=n=5$) in the learning process, and divide all k-sized subsets (252) of a ground set into six groups according to the target number in a subset, *e.g.*, the group of k-subsets with 1 target comprised of 25 subsets (the subsets filled by 1 observed item and 4 unobserved ones), and target number = 5 represents the target subset S_u^{+k} drawn from a k-DPP formulated in Equation 5.4. The k targets and n random items of training instances contain the same number of categories.

Figure 5.4 presents probability ranking trends of LkP$_{PS}$ and LkP$_{NPS}$ employing GCN at four training epochs (0, 30, 100, 200) on Beauty. As we attempt to illustrate the ranking interpretation of k-DPP probability learning, probabilities of 252 k-sized subsets from each ground set (*i.e.*, training instance with 5 targets and 5 unobserved items) are calculated. Before the k-DPP learning process starts, randomly distributed subsets (cardinality k) are assumed to have the equivalent probability of being selected as a k-DPP, *i.e.*, 0.004 (1/252) for each subset. Each dot in a plot denotes the averaged probabilities of subsets that contain the same number of observed items from randomly selected training instances.

We can see that subsets with differing numbers of targets indeed have close normalized k-DPP probabilities (*i.e.*, near 0.004) when epoch = 0. With the learning process proceeding, the discrepancy between k-DPP probabilities of subsets with different target numbers obviously grows at first, *i.e.*, subsets with more targets are endowed with higher probabilities, and the trend tends to be stable when epochs reach 200. These trends illustrate that *relevance ranking interpretation* behind LkP are validated by the k-DPP probability examples, as the target subset ranks higher (with greater k-DPP probability) than other subsets, and the subset with unobserved items ranks lower than the subsets that contain one or more targets. In addition, considering the exclusion of negative subsets in k-DPP (*i.e.*, LkP$_{NPS}$) facilitates widening the ranking gap between target set and unobserved item set compared to LkP$_{NS}$, further enhancing performance improvements (as shown in Table 5.3-Table 5.5). The similar trends can be found on other datasets and MF implementation.

We also conduct a comparison between the averaged k-DPP probabilities (k=n=5) of two types of target subsets on Anime using LkP$_{NS}$, *i.e.*, monotonous target subsets with limited categories ($<$ 4 categories) and diversified target subsets with broader categories ($>$ 5 categories). The observed average probabilities of 0.0041 *vs.* 0.0040, 0.0045 *vs.* 0.0042, and 0.0046 *vs.* 0.0043 for diversified and monotonous target subsets at 0, 100, and 200 epochs respectively, confirm the significant role of *diversity ranking interpretation* across $(k+n)$ k-DPP distributions in our optimization approaches. Because the diverse kernel $\mathbf{K}$ is pre-learned, on average, diverse target sets from half of randomly selected training instances (k-DPP distributions) have a relatively higher

probability than other monotonous target sets from the k-DPP distributions of the remaining half, even before the LkP optimization begins.

These results again indicate the importance of k-DPP normalization, as only in this case the ranking interpretation is provided. We also perform experiments in the case of removing normalization of k-DPP, and then two serious problems occur: (i) the deep learning frameworks (both Pytorch and TensorFlow) cannot handle the complex gradient calculation of the DPP, as the non-normalized determinant values are unstable; (ii) even if we use some techniques, *e.g.*, normalizing diverse kernel or adding activation function to control the values of relevance scores, to avoid the gradient calculation error, the final recommendation results still cannot compare with LkP approaches, as the important ranking interpretation for personalized ranking is ignored. As we mentioned in Section 5.2.2, using standard DPP for item recommendation (a ranking optimization problem) makes the ranking interpretation unreasonable. According to our experiments under the same setting as Table 5.3-Table 5.5, applying standard DPP for ranking probability formulation achieves an unacceptable performance that is much weaker than BPR. This is the reason why the results based on standard DPP are not presented in experiments.

5.4 Conclusion

In this work, we propose a new optimization criterion LkP based on k-DPP set probability comparison for personalized ranking focusing on implicit feedback, enlightening a new perspective (set-level probability maximization) in the formulation of ranking optimization. Based on this, two ranking optimization approaches LkP$_{PS}$ and LkP$_{NPS}$ are designed that take into account different ranking considerations. As such, multiple observed items and unobserved items are directly regarded as subsets with fixed cardinality k, and are compared through the ranking interpretation based on the tailored k-DPP distribution, notably with probabilistic normalization that respects the cardinality constraint. Three levels of evaluation and comparison in the experiments comprehensively demonstrate the superiority of our approaches, and the ranking interpretation is validated through real-world examples. Notably, plugging our optimization criterion into strong baselines provides a ready source of improved performance. Experimental results help us to analyze the different advantages of two LkP approaches and provide insight into making selection between them in different circumstances. LkP$_{PS}$ and LkP$_{NPS}$ also have potential in ranking related tasks, such as Web search and spam detection [Chirita et al., 2005; Saini et al., 2019; Liu, 2020], which will be explored in the future.

5.4.1 Limitations

In this section, considering the characteristics of our proposed set-level ranking optimization, two limitations are acknowledged: (i) The LkP$_{NPS}$ optimization method we proposed employs an approximate calculation to determine the explicitly defined probabilities of exclusion of the negative set and inclusion of the positive set. This approximation may not fully leverage the efficacy of our proposed set-level ranking optimization; (ii) This chapter only explores the application of our proposed ranking optimization method to recommendation tasks, without delving into its application on other ranking tasks.

5.4.2 Future Work

Consequently, we have two-fold future work planned: firstly, to design a method based on DPP subset marginal probability that precisely defines the joint probability of the positive item set and negative item set, thereby enhancing the effect of ranking optimization; and secondly, to explore the application of LkP across various ranking tasks, thereby further enhancing the universality and practicality of the LkP learning model.

Chapter 6

Structured Determinantal Point Processes for Temporal Sets Prediction

Previous chapters focus on diversifying individual recommendations by means of standard DPPs (Chapter 3 and Chapter 4) and k-DPPs (Chapter 5). The dependency correlation within items captured by DPP distributed subsets has been validated in the experiments of different scenarios. We now explore the diversification of an extended personalized prediction task namely *temporal sets prediction* (TSP), which is a challenging task with complicated correlations.

Temporal sets, such as purchasing products successively, ordering takeouts in different meals, selecting courses in each semester, etc., are prevalent in real-world scenarios. Predicting the subsequent set according to previous set sequence holds great practical significance in providing bundle lists for sale, anticipating user's favor, arranging the courses, etc. Given this, *temporal sets prediction* (TSP) has been a trending research topic. Even though some studies have been carried out on this topic, there are certain drawbacks: (*i*) none of them directly models the temporal sets themselves; (*ii*) they make no effort in the direction of designing dedicated temporal sets level objective function; (*iii*) complex relationships within temporal sets sequence are not well defined and captured.

In this chapter, we provide a unified *Structured Temporal Sets Prediction* framework (STSP), characterized by three major innovations to overcome existing limitations. First, a temporal set in a sequence is directly modeled as a structure of Structured Determinantal Point Process (SDPP), in which the probabilistic DPP distribution is modeled over collections of structures (temporal sets) instead of over individual elements (items/products). Second, we propose a specialized temporal sets level objective function with the intention of enhancing the probability of selecting the desired subset of structures as a SDPP. Third, we explore and exploit three types of underlying

relationships, *i.e.*, *preference correlation*, *co-occurrence pattern*, and *temporal sets dependency*, with the first two being integrated into the SDPP structure and the last one being captured by the new proposed objective function. Unlike previous methods that model individual components (items) within the temporal set, STSP views temporal sets from a broader perspective via directly representing the set as a structure element, enabling us to explicitly formulate and capture the underlying relationships. Extensive experiments on four real-world datasets show that our approach consistently outperforms previous methods, and the importance of the three formulated relationships is demonstrated.

6.1 Introduction

Temporal sets represent a sequence of sets, in which each timestamped set contains an arbitrary number of elements (products or items) [Sun et al., 2020c]. In practice, plentiful scenarios can be formalized as temporal sets, such as products of a customer purchased (transactions) at a series of visits of a retail store, dishes of takeouts that a user ordered online in different meals, prescribed drugs for a patient at each time point. Temporal sets prediction (TSP) intends to predict the subsequent set based on previous sets, which could help service providers to understand users for rendering quality service to users [Morsy and Karypis, 2017; Yu et al., 2016], *e.g.*, providing daily song recommendations or suggesting popular bundles. In view of the arbitrary structure of sets and complex relationships within set sequence, more specialized design and broader consideration are needed to achieve this prediction mechanism compared to other sequential predictive problems, *e.g.*, time series [Zhou et al., 2021; Zhang, 2003] and temporal events [Zhao et al., 2020; Feng et al., 2015].

Several efforts have focused on formulating and addressing the task of TSP in recent years. However, we argue that these approaches do not hold the keys to solving the TSP problem. First, they fail to achieve direct modeling of temporal sets, as they focus on using *latent vectors* of the temporal set to implicitly capture relationships within the set, rather than explicitly modeling the set as a whole. The latent vectors are usually obtained by simply combining representations of individual elements in the set, *e.g.*, pooling operations [Hu and He, 2019; Yu et al., 2016] and attention methods [Sun et al., 2020c; Yu et al., 2020a], indicating that their emphasis remains on modeling the elements within the set. Second, there is no contribution made in designing new loss functions specifically adapted for sets prediction. Previous studies are still constrained in exploiting existing loss functions that merely compare individual items in sets, *e.g.*, the usage of weighted mean square loss [Hu and He, 2019], the common binary cross-entropy loss [Sun et al., 2020c; Yu et al., 2022], and the multi-

label soft margin loss [Jung et al., 2021; Yu et al., 2020a]. Individual item-wise objective functions are inadequate to learn the overall properties and structure of sets [Qi et al., 2017; Wang et al., 2017a]. Third, the underlying relationships within temporal sets are not well explored. Considering the complex relationships involved, *e.g.*, the correlations among arbitrary number of items in a set and dependencies across sets sequence, TSP has been a challenging task [Hu and He, 2019]. However, existing methods mainly focus on employing widely-used sequence processing models (*e.g.*, Recurrent Neural Networks (RNNs) [Hu and He, 2019; Choi et al., 2016] and Transformer [Sun et al., 2020c; Yu et al., 2022]) to identify general interrelated relationships, instead of uncovering the more specific underlying relationships in TSP.

We suggest that directly modeling temporal sets themselves is an intuitive and natural approach for temporal sets prediction. It implies explicitly manipulating a temporal set comprised of an arbitrary number of items as an integrated element, rather than transforming the set into a hidden vector by combining the latent representations of the items within it. To achieve this, an extension of Determinantal Point Processes (DPPs) — Structured DPPs (SDPPs) [Kulesza and Taskar, 2010] are employed, which offer natural probabilistic models over collections of structures (such as temporal sets in this work) instead of over subsets of individual items. Based on this, a temporal set in a sequence is directly modeled as a structure (integrated element) of SDPP, which earns three advantages. (*i*) Structure-based probabilistic distribution provides holistic understanding of temporal sets and their properties, and sets-level data can be better described because it captures overall statistical characteristics [Gillenwater et al., 2012a]. (*ii*) By virtue of the direct focus of structures, the relationships related to temporal sets (*preference correlation* and *co-occurrence pattern*) can be explicitly formulated in the construction of structure. The co-occurrence pattern relationship among the temporal set is first explored and defined, which contributes to understanding the organization of sets; (*iii*) From a broader perspective, we propose a new objective function tailored for temporal sets based optimization, allowing for taking the third relationship (*temporal sets dependency* across sets sequence) into account. In this sense, we can directly compare sets (or structures) rather than individual items by measuring the probabilities of corresponding temporal sets sequences being the SDPP-distributed subsets.

On balance, we build a Structured Temporal Sets Prediction (STSP) framework, in which the temporal set with arbitrary number of elements is modeled as a SDPP structure. This explicit model of temporal sets and global probabilistic distribution over structures enable us to formulate three relationships within temporal sets, which can be intuitively analyzed and evaluated in experiments. Considering the repulsive characteristic of DPP-based probabilistic models, STSP is capable of producing diverse

set-level predictions.

We summarize the contributions of this chapter as follows:

- We develop a new perspective on temporal sets prediction by explicitly modeling the temporal set as an integrated element, breaking away from traditional manner of combining latent representations in existing studies.
- We propose a new concept of representation — co-occurrence for representing temporal sets, based on which the structure of each set is first explored.
- We discuss and formulate three types of underlying relationships within temporal sets, and reveal their effects by specialized experimental settings. This advances our understanding of the complexities of temporal sets.
- A tailored sets-level objective function is introduced for TSP tasks. This new objective function elevates the STSP framework, making it more intuitively aligned with the goals of temporal sets prediction, and contributes to making our methodology more effective and easily comprehensible.

6.2 Preliminary

This section provides some preliminaries, including the formalization of TSP and structured determinantal point process (SDPP). Considering the complexity of the task of temporal sets prediction, and given that the probabilistic SDPP model has not been extensively explored or discussed, we understand that it could involve numerous mathematical symbols that may be difficult to comprehend. Therefore, in order to facilitate better understanding and cross-reference, we have provided a comprehensive list of key mathematical symbols used in this work in Table 6.1.

6.2.1 Problem Formalization

Temporal sets can be defined as a sequence of sets, where each set consists of an arbitrary number of elements and a timestamp. In real-word scenarios, the phenomena of temporal sets are pervasive, *e.g.*, multiple transactions of shopping lists at different visits of supermarket, several collections of items added into an online shopping cart at a series of browses of E-commerce platform, etc. As a new category of temporal data, temporal sets are more complicated than time series and temporal events [Yu et al., 2020a]. The main reason for the extensive attention received by the temporal sets prediction problem owes to the significant practical implications. For example, forecasting users' subsequent transactions or online shopping carts helps users to make decisions on choosing items and enables service providers to properly organize supplies. Unlike

Table 6.1: The Key Symbols Used in STSP Framework.

Notation	Interpretation
$\mathbb{U}, \mathbb{V}$	entire user, entire item
$S_t, \boldsymbol{S}$	a temporal set (structure), a temporal sequence consisting of sets
$v_i, S_a^{(r)}$	an item in $\mathbb{V}$ indexed by i, the rth item in S_a
$S_a^{(\alpha)}, S_b^{(\beta)}$	the αth item in set S_a and βth item in set S_b
$\mathbf{E}^P, \mathbf{E}^C \in \mathbb{R}^{d \times \lvert V \rvert}$	preference embeddings and co-occurrence embeddings for $\mathbb{V}$
$\mathbf{e}_i^P, \mathbf{e}_i^C$	preference embedding and co-occurrence embedding for item v_i
$\mathbf{h}_{v_i}^P, \mathbf{h}_{S_t}^P$	hidden representations of item v_i and set S_t
$\mathbf{h}_v^P, \mathbf{h}_{\boldsymbol{S}}^P$	preference representation of a sequence on item-level and set-level
$\mathbf{p}^{(\boldsymbol{S})} \in \mathbb{R}^d$	final preference representation for set sequence $\boldsymbol{S}$
$\mathbf{H}^C = \left[\mathbf{h}_1^C, \cdots, \mathbf{h}_R^C\right] \in \mathbb{R}^{d \times R}$	hidden co-occurrence representations of R items in a set
$\mathbf{C}^{(\boldsymbol{S})} \in \mathbb{R}^{d \times \lvert V \rvert}$	final co-occurrence representations of entire items *w.r.t.* a sequence $\boldsymbol{S}$
$\mathbf{c}_r^{(\boldsymbol{S})}, \mathbf{c}_{r'}^{(\boldsymbol{S})}$	final co-occurrence representations of rth and r'th items in a set *w.r.t.* $\boldsymbol{S}$
$\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d \times d}$	trainable weight matrices for gated fusion of $\mathbf{h}_v^P, \mathbf{h}_{\boldsymbol{S}}^P$
$\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d}$	trainable weight matrices of self-attention for learning co-occurrence
$q\left(S_a\right), \phi(S_a)$	quality of structure S_a, diversity measure of structure S_a
$w(S_a)$	weight of structure S_a, $q\left(S_a\right) = \exp\left(w\left(S_a\right)\right)$
$w\left(S_a^{(r)}\right),\ \hat{\boldsymbol{y}}_i^P$	node weight of rth item in S_a, represented by preference score $\hat{\boldsymbol{y}}_i^P$
$w\left(S_a^{(r)},\ S_a^{(r')}\right),\ \hat{\boldsymbol{y}}_{rr'}^C$	edge weight of rth to r'th in S_a, represented by co-occurrence score $\hat{\boldsymbol{y}}_{rr'}^C$
$Sim(S_a, S_b)$	similarity measure between two sets S_a and S_b
$\phi_\alpha\left(S_a^{(\alpha)}\right), \phi_\alpha\left(S_a^{(\beta)}\right)$	diversity feature of $S_a^{(\alpha)}$ and $S_a^{(\beta)}$ in sets S_a and S_b
$\mathbf{K}$	diverse kernel, a $\lvert\mathbb{V}\rvert \times \lvert\mathbb{V}\rvert$ matrix
$T^{(+)}, T^{(-)}$	diversified observed set and randomly formed set for learning $\mathbf{K}$
ℓ	objective function for diversity kernel learning
$\mathcal{P}^{(\boldsymbol{S})}$	a SDPP over a sequence specified ground set
$\mathbf{L}^{(\boldsymbol{S})}$	A SDPP kernel on a sequence specified ground set (structures)
$\mathbf{L}_{A \cup B}^{(\boldsymbol{S})}$	A sub-matrix of $\mathbf{L}^{(\boldsymbol{S})}$ indexed by previous sets and target sets
$\hat{\boldsymbol{y}}_i$	final prediction scores of item v_i for evaluation

previous work that focuses mainly on the accuracy of predicted temporal sets, we take another commonly used metric, *i.e.*, diversity, in recommendation [Zhang and Hurley, 2008; Liu et al., 2022] and natural language processing [Henderson and Brill, 2000; Gimpel et al., 2013] fields into account based on determinantal point process (DPP). Forecasting diversified and accurate sets is endowed with more practical significance, as diversification facilitates raising potential purchasing/browsing desire of users and avoiding monotonous sets.

Temporal sets prediction can be formalized as follows: Let $\mathbb{U} = \left\{u_1, \cdots, u_{|U|}\right\}$ and $\mathbb{V} = \left\{v_1, \cdots, v_{|V|}\right\}$ be the entirety of users and items (goods or products), respectively. S_t with an arbitrary number of interacted elements is adopted to denote a user interaction set, $S_t \subset \mathbb{V}$. Given a user's historical user-set interactions represented as a temporal sequence of sets $\boldsymbol{S} = \{S_1, S_2, \cdots, S_t\}$, the goal of temporal sets prediction is to predict the subsequent set according to the historical records, that is, $\hat{S}_{t+1} = f(\boldsymbol{S})$. Compared to other temporal data prediction methods [Zhou et al., 2021; Hidasi et al., 2015; Liu et al., 2014a], relationships within temporal sets that need to be considered to solve the above problem are more complicated, *e.g.*, relations (co-occurrence and preference) among elements and temporal dependencies across the set sequence. These relationships are formulated in the following section. Incorporating the diversity factor into the TSP task further elevates the challenge of this work.

6.2.2 Structured Determinantal Point Processes

To smoothly introduce the concept of Structured Determinantal Point Processes (SDPP) and to clarify the differences between standard DPP and SDPP, as well as the connection between SDPP and Temporal Sets Prediction (TSP) tasks, we initially present a concise overview of DPP, using notations that align with the tasks in this chapter. A DPP $\mathcal{P}$ over a ground set $\mathcal{Y} = \{1, 2, \ldots, M\}$, namely the item or product catalog, is a probability measure on $2^{\mathcal{Y}}$ (the set of all subsets of $\mathcal{Y}$). If $Y \subseteq \mathcal{Y}$ is a random subset drawn according to $\mathcal{P}$, the probability is $\mathcal{P}(Y) \propto \det(\mathbf{L}_Y)$, where $\mathbf{L} \in \mathbb{R}^{M \times M}$ is a real, positive semi-definite kernel matrix indexed by the elements of Y. The marginal probability of including one element Y_i is $\mathcal{P}(Y_i) = \mathbf{L}_{ii}$. That is, the diagonal of $\mathbf{L}$ gives the marginal probabilities of inclusion for individual elements of $\mathcal{Y}$. The probability of selecting two items Y_i and Y_j is $\mathrm{L}_{ii}\mathrm{L}_{jj} - \mathrm{L}_{ij}^2 = \mathcal{P}(Y_i)\mathcal{P}(Y_j) - \mathrm{L}_{ij}^2$. The entries of the kernel $\mathbf{L}$ usually measure similarity between pairs of elements in $\mathcal{Y}$. Thus, highly similar elements are unlikely to appear together. In this way, the repulsive characteristic (*i.e.*, diversity) with respect to a similarity measure is captured.

Reviewing the formalization of temporal sets prediction, *i.e.*, forecasting the subsequent sets given previous set sequence, the basic component of this prediction task

is the set instead of individual items or products, which is also the point that makes this problem apart from previous temporal prediction tasks. This means that directly applying standard DPPs on subsets of items is not sufficient to handle the complicated relations inherent in temporal sets. To address this, we employ structured determinantal point process (SDPP) [Kulesza and Taskar, 2010], which offers a natural probabilistic model over sets of structures (*i.e.*, temporal sets). In SDPP case, elements of a ground set $\mathbb{S}$ are structures, which means that we will no longer think of $\mathbb{S} = \{1, 2, \ldots, M\}$; instead, each element $S_t \in \mathbb{S}$ is a structure given by a collection of R parts $\left(S^{(1)}, S^{(2)}, \ldots, S^{(R)}\right)$, each of which takes a value from a finite set of M^R possibilities. In the context of TSP, each structure element is a user interaction set S_t at time t containing R items/products in $\mathbb{V}$. Thus, a temporal set can be explicitly represented as an integrated structure. A random subset drawn from a SDPP $\mathcal{P}$ can be a temporal sequence of sets (structures), *i.e.*, $\boldsymbol{S}$. In this setting, we can regard TSP as a problem of drawing salient (accurate and diversified) sets of temporal structures according to the SDPP distribution. SDPP directly treats the temporal set as an element (structure) and possesses powerful ability of capturing dependency across structures incorporating notions of quality and diversity by means of global probabilistic measure, and thus naturally fits the TSP problem. As a result, introducing SDPP into temporal sets prediction will push the corresponding research into a new frontier.

However, we need to note that when the number of elements in $\mathbb{S}$ is exponential, structured DPPs actually model a distribution over the doubly exponential number of subsets of an exponential $\mathbb{S}$ [Gillenwater et al., 2012a]. An immediate challenge is that it is difficult to explicitly write down the SDPP kernel $\mathbf{L}$. In this work, we therefore conduct SDPP distribution calculation based on the temporal set sequence specified SDPP kernel $\mathbf{L}^{(\boldsymbol{S})}$, *i.e.*, a kernel over *sequence ground set* that only involves structures (sets) of a temporal set sequence $\boldsymbol{S}$ instead of over all structures. To further efficiently compute $\mathbf{L}^{(\boldsymbol{S})}$, we use the quality/diversity decomposition to define its entries presented in the next section.

6.3 Methodology

In this section, we introduce our structured temporal sets prediction framework namely STSP in detail. To make the proposed method comprehensible, we first present how to learn preference representation and co-occurrence representation, and then describe how to incorporate these two types of representations for formulating SDPP structures. Finally, the training mechanism of STSP based on SDPP distributions is displayed.

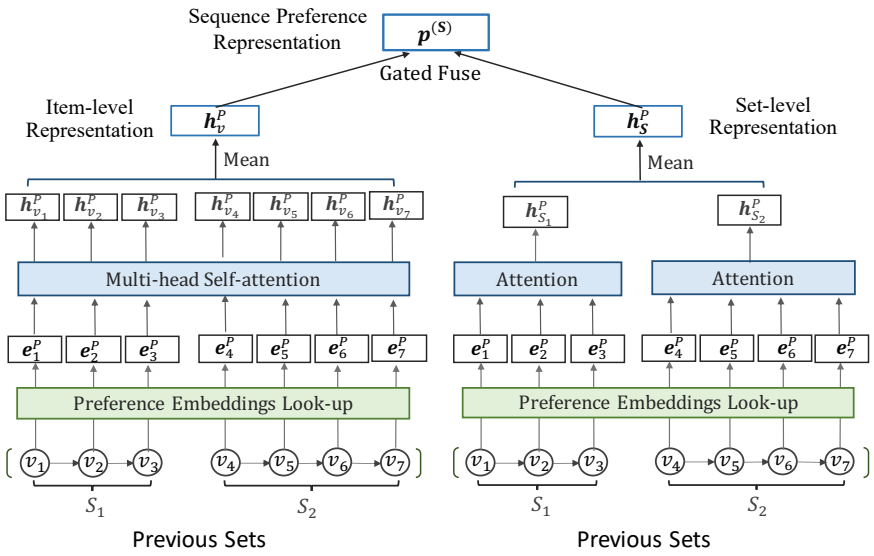


Figure 6.1: Preference Representation Learning.

6.3.1 Preference Representation

Learning **p**ref**f**erence **r**epresentation (PFR) based on previous set sequence is a critical step to capture correlations between candidate items (that likely appear in subsequent sets) and users' dynamic tastes or interests. The preference representation in this work is defined to model temporal sets' sequence-level personalization, namely *preference correlation*. As mentioned above, relationships within temporal sets are complicated. To learn expressive PFR, we define trainable embeddings for preference learning on multiple levels, *i.e.*, item-level and set-level representations. The final PFR incorporates these two levels of representations through a gated neural module, which helps to capture users' dynamic preference correlations among individual items and temporal sets. We use the learned PFR to formulate SDPP structures for building STSP framework as presented in Section 6.3.3.1.

We use $\mathbf{E}^P \in \mathbb{R}^{d \times |V|}$ to denote the preference embedding matrix for items, where d is the embedding dimension. Then $\mathbf{e}_i^P$ denotes the embedding vector of item v_i. Embeddings of items in a sequence of previous sets will be fed into multi-head self-attention which has been widely used to process the sequence data [Sun et al., 2020c; Voita et al., 2019]. We set H heads for multi-head self-attention layers. A softmax function is adopted to normalize attention coefficients. Based on the normalized attention coefficient matrix of each head, an item representation can jointly capture the focus from different parts of the previous sets sequence. The hidden representations of items from multiple heads are concatenated and linearly transformed as the output (*i.e.*, $\mathbf{h}_{v_i}^P \in \mathbb{R}^d$) of the self-attention layer. To obtain the item-level PFR of a set sequence

namely $\mathbf{h}_v^P$, we directly apply a popular manner called average pooling operation to aggregating item hidden representations in sequence. The architecture of item-level PFR is shown in the left part of Figure 6.1. As we aim to design a general temporal sets prediction framework based on SDPP, the standard multi-head self-attention and popular aggregation operation are directly employed. As the multi-head self-attention is well known, we omit the detailed formulations.

Learning set-level representation means computing a vector representation for each set in the temporal set sequence based on the preference embeddings of items in those sets. In this work, the set-level representation is computed referring to the set embedding method [Sun et al., 2020c] that uses multi-head attention rather than simply adopting pooling methods, which is shown in the right part of Figure 6.1. To achieve this, K trainable queries, each with d/K dimensions, are defined to extract multi-faceted features from a set. The attention weight matrix of items in a set for K queries is represented by calculating the similarity between each query and each item embedding vector. Then, a d/K-dimensional set representation from a query's perspective is calculated by accumulating weighted (attention weight of items for the query) embedding vectors of items that belongs to the set. We concatenate K set representations from K queries to get the preference representation of a set S_t, *i.e.*, $\mathbf{h}_{S_t}^P$. Similarly, the set-level sequence preference representation $\mathbf{h}_{\boldsymbol{S}}^P$ is acquired using the average pooling method.

To obtain the final sequence preference representation $\mathbf{p}^{(\boldsymbol{S})} \in \mathbb{R}^d$ *w.r.t.* the temporal sets sequence $\boldsymbol{S}$, which integrates preference correlations from two levels (item-level and set-level), we apply a gated fusion module [Pan et al., 2020; Lv et al., 2019; Sun et al., 2020c] as follows:

$$\begin{aligned} \mathbf{g} &= \sigma\left(\mathbf{W}_1\mathbf{h}_v^P + \mathbf{W}_2\mathbf{h}_{\boldsymbol{S}}^P + b_f\right), \\ \mathbf{p}^{(\boldsymbol{S})} &= \mathbf{g} \odot \mathbf{h}_v^P + (1-\mathbf{g}) \odot \mathbf{h}_{\boldsymbol{S}}^P \end{aligned} \tag{6.1}$$

where σ is the sigmoid activation function and $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d\times d}$ are trainable parameters. Finally, the sequence preference representation from gated fusion can be used to compute the sequence-related potential preferences of a user for candidate items.

6.3.2 Co-occurrence Representation

Temporal sets sequence stands apart from other sequence data, as the relations among items/products in the set-level element complicate the task of predicting next set. Previous studies on TSP only focus on learning preference representation of a temporal sequence [Hu and He, 2019; Yu et al., 2020a], similar to $\mathbf{p}^{(\boldsymbol{S})}$ in this work, which is also the common technique of sequential recommendation [Liu et al., 2022] or basket com-

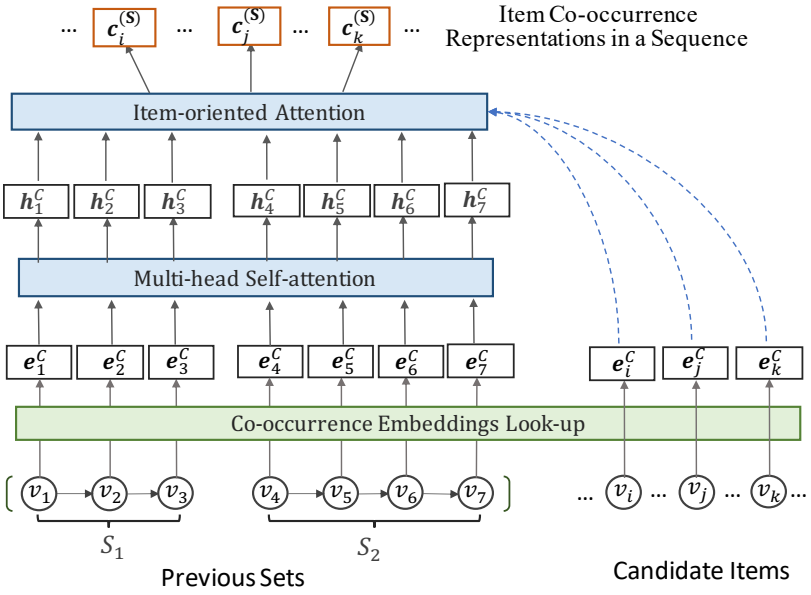


Figure 6.2: Co-occurrence Representation Learning.

.

pletion [Warlop et al., 2019]. A consequent limitation of this manner is that cohesion relations among items in a set are ignored. In this work, the cohesion of items represents the regular pattern of which item pairs tend to co-occur in a set. For example, it would be found that a smart phone and screen protector usually appear in an online shopping cart, and steak and salad often jointly show in a transaction. These are so-called *co-occurrence pattern*, *i.e.*, one of the three defined relationships within temporal sets. The preference correlation focuses on users' dynamic preference, *i.e.*, relevance between sequence representation and candidate items. In contrast, co-occurrence patterns aim to identify *potential items*, which may not be immediately evident from user preference estimation but have a strong bond with user-preferred items (as predicted by preference representations). These *potential items* might spark users' interest due to their strong connection with already preferred items. In essence, the goal of mining co-occurrence patterns is to learn the cohesion strength between candidates, which is important to complete or complement users' subsequent sets.

To capture these valuable co-occurrence patterns behind set-level behaviors, we propose a new type of representation, *i.e.*, **c**o-**o**ccurrence **r**epresentation (COR), which is learned based on the co-occurrence embedding matrix $\mathbf{E}^C \in \mathbb{R}^{d \times |V|}$ for entire items. Unlike preference representation, COR is not directly associated with personalization, but provides extra evidence in the stage of computing relevance score for a candidate item via estimating the probability of the candidate co-occurring with other ones in

the same set. In this work, we propose to learn COR using the item-oriented attention network [Zhou et al., 2018; Sun et al., 2020c], which adaptively learns each candidate's representation with respect to sets in the sequence. The architecture of learning COR is presented in Figure 6.2. Similar to item-level representation learning, co-occurrence embeddings of items in a set sequence are fed into a multi-head self-attention layer to obtain the hidden item representations that consider sequence dependency, *i.e.*, $\mathbf{H}^C = \left[\mathbf{h}_1^C, \cdots, \mathbf{h}_R^C\right] \in \mathbb{R}^{d \times R}$ with R items in a temporal sequence. The co-occurrence representations for all items derived based on the sequence $\boldsymbol{S}$, from the item-oriented attention layer, can be formulated as follows:

$$\mathbf{C}^{(\boldsymbol{S})} = \mathbf{W}_I^V \mathbf{H}^C \operatorname{Softmax}\left(\left(\mathbf{W}_I^K \mathbf{H}^C\right)^\top \mathbf{W}_I^Q \mathbf{E}^C\right), \tag{6.2}$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d}$ are learnable matrices; $\mathbf{C}^{(\boldsymbol{S})} \in \mathbb{R}^{d \times |V|}$ contains sequence $\boldsymbol{S}$ specified item-level co-occurrence representations (final co-occurrence representations) for each candidate in $\mathbb{V}$; and $\mathbf{E}^C$ represents candidates' co-occurrence embeddings.

To introduce COR learning into the training procedure, we calculate cohesion scores (strength measure of two items co-occurring in the same set) using items' CORs based on the structure of sets in training sequences (previous sets and target sets), which will be involved in the training objective. In this setting, the learning process of COR will be guided to consider the cohesion between candidates in training sets, and thus the co-occurrence patterns among items in sets of the entire training sequences are modeled after optimization. In the evaluation procedure, co-occurrence representations of entire items with respect to a specific set sequence can be obtained from the item-oriented attention network, which will engage in ranking all candidate items for suggesting next sets. The learned CORs are supposed to offer high co-occurrence scores for item pairs that actually co-occur in the subsequent set, and thus provide extra evidence for ranking. By implanting cohesion among items into SDPP structure, a specialized set-level objective function is constructed, which is described in the next section.

6.3.3 STSP FRAMEWORK

The key of explicitly modeling temporal sets, *i.e.*, treating items of a set as one unit, is to construct SDPP structures by means of transforming multifarious information (*e.g.*, relevance and cohesion of items in a set) into a basic element. We now start to describe the formulation of SDPP structures for TSP.

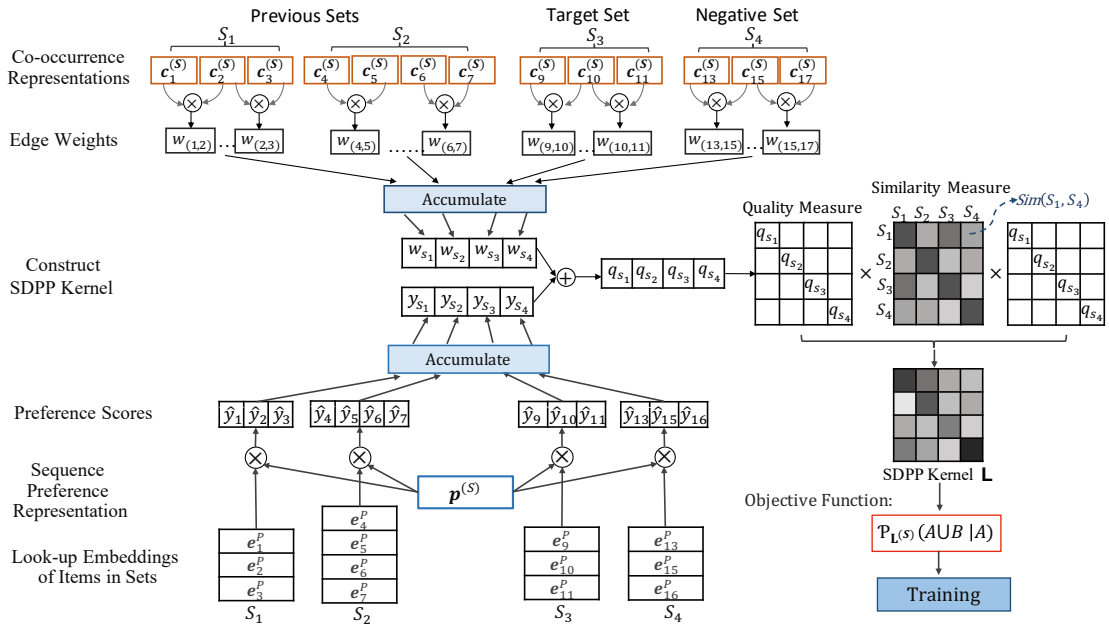


Figure 6.3: STSP Training Architecture.

6.3.3.1 Structure Formulation

As mentioned above, entries of SDPP kernel $\mathbf{L}$ is defined using quality/diversity decomposition that is formulated by

$$L_{ab} = q(S_a)\phi(S_a)^\top \phi(S_b) q(S_b), \tag{6.3}$$

where $q(S_a)$ is a nonnegative measure of the quality of a structure (set) S_a, and $\phi(S_a)$ represents a vector of diversity features so that $\phi(S_a)^\top \phi(S_b)$ is a measure of the similarity between structures S_a and $\boldsymbol{S}_b$. SDPPs assume a factorization of the quality score $q(S_a)$ and similarity score $\phi(S_a)^\top \phi(S_b)$ into parts for allowing efficient normalization and sampling. Formally, SDPP decomposes quality multiplicatively and similarity additively:

$$q(S_a) = \prod_{r=1}^{R} q\left(S_a^{(r)}\right) \quad \phi(S_a) = \sum_{r=1}^{R} \phi\left(S_a^{(r)}\right). \tag{6.4}$$

For temporal sets, the part $S_a^{(r)}$ denotes the rth item in set S_a that contains R items.

To measure the importance or salience of items and the relative strength of relationships between them, we calculate the weight of a temporal set for constructing

SDPP structure defined as

$$w(S_a) = \sum_{r=1}^{R} w\left(S_a^{(r)}\right) + \sum_{S_a^{(r)} \in S_a, S_a^{(r')} \in S_a} w\left(S_a^{(r)}, S_a^{(r')}\right), \tag{6.5}$$

where $r \neq r'$. The first term denotes the accumulation of items' importance. For TSP tasks, the importance of part $S_a^{(r)}$ (*i.e.*, an item) can be represented by its predicted relevance score *w.r.t.* the sequence $\boldsymbol{S}$, which is calculated by

$$\hat{\boldsymbol{y}}_i^P = \mathbf{p}^{(\boldsymbol{S})^\top} \mathbf{e}_i^P, \tag{6.6}$$

where $\mathbf{p}^{(\boldsymbol{S})}$ is the final sequence preference representation of a sequence $\boldsymbol{S}$, and $\mathbf{e}_i^P$ denotes the preference embedding of item v_i in $\mathbb{V}$ (*i.e.*, the rth part $S_a^{(r)}$ in S_a). The second term in Equation 6.5 is used to represent the strength of edges in S_a. SDPP is originally designed to build structures of threads or paths [Gillenwater et al., 2012a], which means that there is an ordering in the structure. Therefore, only the strength of edges of adjacent elements in the path (structure) is considered. As for TSP, items in the same set usually do not have an inherent ordering. We therefore propose to capture co-occurrence patterns of items on set level. Specifically, this work considers the relations (cohesion) between any two items of a set. In this sense, the edge weight of any two items ($S_a^{(r)}$, $S_a^{(r')}$ in S_a), *i.e.* $w\left(S_a^{(r)}, S_a^{(r')}\right)$, used to represent the strength of cohesion between them is calculated by

$$\hat{\boldsymbol{y}}_{rr'}^C = \mathbf{c}_r^{(\boldsymbol{S})^\top} \mathbf{c}_{r'}^{(\boldsymbol{S})}, \tag{6.7}$$

where $\mathbf{c}_r^{(\boldsymbol{S})}$ and $\mathbf{c}_{r'}^{(\boldsymbol{S})}$ are the final co-occurrence representations of items with respect to temporal sets sequence $\boldsymbol{S}$, learned based on co-occurrence embeddings. This equation indicates that we use the quantified cohesion score to measure the possibility of two items co-occurring in the same set. By means of incorporating it into the training objective function, item pairs that tend to co-occur in the subsequent set will receive relatively high cohesion scores. From Equation 6.5, we can learn that PFR learning and COR learning are integrated into the formalization of set weight. In the scenario of our temporal sets prediction task, we employ concepts of relevance and co-occurrence to represent the importance (weight) of a structure (temporal set) to the prediction model, which are distinct from the notion of weight in original SDPP. Consequently, we have chosen to use different symbols to represent the node weight and edge weight within the SDPP framework. In order to convert the set weight function into the appropriate multiplicative form of the quality measure from Equation 6.4, a log-linear model, setting $q(S_a) = \exp(w(S_a))$, is adopted.

We have detailed how to calculate the quality of a SDPP structure above. Here we present the calculation of set similarity measurement, which contributes to the diversity component of SDPP. Aligning with Equation 6.3 and Equation 6.4, the set similarity function can be factorized into the item similarity function as follows

$$\begin{aligned} Sim(S_a, S_b) &= \phi(S_a)^\top \phi(S_b) \\ &= \left(\sum_{S_a^{(\alpha)} \in S_a} \phi_\alpha \left(S_a^{(\alpha)} \right)^\top \right) \left(\sum_{S_b^{(\beta)} \in S_b} \phi_\beta \left(S_b^{(\beta)} \right) \right) \\ &= \sum_{\alpha,\beta} \phi_\alpha \left(S_a^{(\alpha)} \right)^\top \phi_\beta \left(S_b^{(\beta)} \right). \end{aligned} \tag{6.8}$$

Here, $\phi_\alpha \left(S_a^{(\alpha)} \right)$ and $\phi_\beta \left(S_b^{(\beta)} \right)$ represent k-dimensional diversity features of two items belong to S_a and S_b, respectively. This factorization indicates that the diversity measurement of two structures for each SDPP kernel entry is built by accumulating the similarity of any two items from different structures.

To connect the common diversity concept (*i.e.*, scope of categories) in related fields (*e.g.*, recommendation [Wu et al., 2019b; Liang and Qian, 2021] and basket completion [Warlop et al., 2019; Gartrell et al., 2016], etc.) and diversity measurement in SDPP, we propose a category-aware diverse kernel $\mathbf{K}$. The number of items $|\mathbb{V}|$ in TSP dataset is usually large, which means that learning a nonparametric full-rank diverse kernel $\mathbf{K}$ is computationally expensive. Referring to [Gartrell et al., 2017], we use the low-rank factorization of the $|\mathbb{V}| \times |\mathbb{V}|$ $\mathbf{K}$ matrix as: $\mathbf{K} = \mathbf{A}\mathbf{A}^\top$. In this setting, a learning objective function is built by maximizing the log-likelihood of sampling observed diverse subsets $T^{(+)}$ and minimizing the log-likelihood of sampling negative subsets $T^{(-)}$,

$$\ell = \sum_{\left(T^{(+)}, T^{(-)}\right) \in \mathcal{T}} \log\det\left(\mathbf{K}_{T^{(+)}}\right) - \log\det\left(\mathbf{K}_{T^{(-)}}\right), \tag{6.9}$$

where $\mathcal{T}$ denotes the collection of paired sets used for training. To obtain the observed diverse subsets, we select as many items belonging to different categories as possible from an observed temporal set. For example, there are four items from three categories (c_1, c_2, c_3) in a set $\{v_5^{c_1}, v_6^{c_2}, v_7^{c_3}, v_8^{c_3}\}$, and two diverse subsets $\{v_5^{c_1}, v_6^{c_2}, v_7^{c_3}\}$ and $\{v_5^{c_1}, v_6^{c_2}, v_8^{c_3}\}$ can be selected. For each selected diverse subset, a same-sized negative subset with randomly selected items will be provided. Considering the dependency among items and computational efficiency, observed temporal sets with item size in the range of 5 to 20 are used for selecting diversified subsets, and a subset should contain at least 3 items. Based on this, a category-aware diverse kernel $\mathbf{K}$ can be learned, which tends to make the corresponding DPP-distributed samples containing diverse

categories. Namely, the items drawn from pre-learned $\mathbf{K}$ are diversified. Reviewing the description of DPPs in Section 6.2.2, the diversity characteristic is contributed by the entries of DPP kernel that measure the similarity between pairs of elements. Since $\mathbf{K}$ facilitates sampling diversified subsets, we can directly take the corresponding entries of $\mathbf{K}$ into Equation 6.8 as the similarity measurement of two items, which means that the corresponding vectors of items in the matrix $\mathbf{A}$ (low-rank factorization of $\mathbf{K}$) represent their diversity features.

6.3.3.2 Sets-level Objective Function

Up to this point, we have formulated the individual components, *i.e.*, quality weight and diversity measurement, for constructing the SDPP structure, in which two types of relationships within temporal sets, namely *sequence preference correlation* and *co-occurrence pattern* are respectively embedded. Actually, the previous descriptions in this section have all been laying the groundwork for the specialized objective function of STSP framework, which is designed with the motivation of considering the third type of relationships, *i.e.*, *temporal sets dependency*. To capture the dependency across temporal sets, various sequence models are often used, *e.g.*, RNN [Hu and He, 2019], multi-head self-attention [Sun et al., 2020c; Yu et al., 2020a], etc. However, these approaches only consider the dependency in representation learning part, such as the description of Section 6.3.1 and Section 6.3.2. In this work, we attempt to capture the dependency in both representation learning (Figure 6.1 and Figure 6.6) and objective function. Inspired by the conditional DPP distribution [Liu et al., 2022; Kulesza et al., 2012], we define a dedicated sets-level conditional likelihood function based on conditional SDPP probability,

$$\mathcal{P}^{(\boldsymbol{S})}\left(\boldsymbol{Y}=A\cup B \mid \boldsymbol{Y}=A\right)=\frac{\det\left(\mathbf{L}^{(\boldsymbol{S})}_{A\cup B}\right)}{\det\left(\mathbf{L}^{(\boldsymbol{S})}+\mathbf{I}_{\bar{A}}\right)}. \tag{6.10}$$

In this equation, $\boldsymbol{Y}$ is distributed as a SDPP with temporal sets sequence $\boldsymbol{S}$ specified kernel $\mathbf{L}^{(\boldsymbol{S})}$; A denotes a subset containing previous sets, and $A\cup B$ represents a subset comprised of sets of the complete training sequence (previous sets and target sets). $\mathcal{P}^{(\boldsymbol{S})}\left(\boldsymbol{Y}=A\cup B \mid \boldsymbol{Y}=A\right)$ means the probability of if the previous sets is drawn as a SDPP subset of structures, how likely it is that sets of the complete training sequence is SDPP-distributed. This means that the specialized probabilistic function is naturally appropriate to the concept of TSP, *i.e.*, predicting the next sets conditioned on previous ones. As mentioned in Section 6.2.2, we cannot afford to construct kernel $\mathbf{L}$ over all possible structures. Fortunately, the training instance in the period of TSP learning is a definite set sequence. We therefore can form the sequence-specified ground set

$\mathbb{S}^{(\boldsymbol{S})}$ that only involves the sequence related structures, *i.e.*, previous sets (A), target sets (B), and randomly selected negative sets (Z) that correspond to the target ones. Thus $\mathbf{L}^{(\boldsymbol{S})}$ represents the sequence-specified kernel on the ground set $\mathbb{S}^{(\boldsymbol{S})}$. In the normalization constant, $\mathbf{I}_{\bar{A}}$ is the matrix with ones in the diagonal entries indexed by elements of $\mathbb{S}^{(\boldsymbol{S})} - \bar{A}$ and zeros everywhere else.

6.3.3.3 Optimization

We then reach the objective function for TSP training by taking the logarithm of the likelihood function formulated by

$$\mathcal{L}(\Theta) = \prod_{u \in \mathbb{U}} \prod_{\boldsymbol{S} \in \mathcal{D}_u} \mathcal{P}^{(\boldsymbol{S})} = \sum_{u \in \mathbb{U}} \sum_{\boldsymbol{S} \in \mathcal{D}_u} \log \left(\mathcal{P}^{(\boldsymbol{S})} \right), \tag{6.11}$$

where $\boldsymbol{S}$ denotes an observed temporal sets sequence of user u, and $\mathcal{D}_u$ is the compilation of training sequences from u; $\mathcal{P}^{(\boldsymbol{S})}$ indicates the conditional likelihood (Equation 6.10) specified by a training sequence of user u. Θ represents model parameters including attention weights in two types of representation learning, parameters of gated fusion module, preference embeddings, and co-occurrence embeddings. It can be noted that the dependency across temporal set sequence is considered in this dedicated objective function, as the corresponding calculation is not based on independent elements or sets, but directly treats the sets of temporal sequence as a SDPP subset. The objective function enables us to transform the goal of temporal sets prediction into maximizing the probability of drawing the complete sets sequence as a SDPP, given the condition where previous sets are distributed as a SDPP, making the STSP framework rather *intuitive*. Looking at the architecture of STSP as depicted in Figure 6.3, it is evident that all preparatory tasks, from learning representations to establishing diversity, are inherently aligned with this intuitive objective. This suggests that STSP is a model that is purposefully designed with its goal at the core. Figure 6.3 also demonstrates that STSP is a general framework, as it can be built upon other types of representation forms for constructing SDPP structure, thus reinforcing its adaptability.

To perform optimization for learning parameters Θ of STSP, we can maximize the log-likelihood

$$\mathcal{L}(\Theta) = \sum_{u \in \mathbb{U}} \sum_{\boldsymbol{S} \in \mathcal{D}_u} \left[\log \left(\det \left(\mathbf{L}^{(\boldsymbol{S})}_{A \cup B}(\Theta) \right) \right) - \log \left(\det \left(\mathbf{L}^{(\boldsymbol{S})}(\Theta) + \mathbf{I}_{\bar{A}} \right) \right) \right] \tag{6.12}$$

Gradient-based learning methods, such as gradient descent and stochastic gradient descent, provide rigorous foundation for optimization because of their theoretical guarantees, but require knowledge of the gradient of $\mathcal{L}(\Theta)$. In the discrete DPP setting, this gradient can be computed straightforwardly [Affandi et al., 2014], and we provide

the computation process as follows.

$$\begin{aligned}\frac{\partial \mathcal{L}(\Theta)}{\partial \Theta} = &\sum_{u \in \mathbb{U}} \sum_{S \in \mathcal{D}_u} \operatorname{tr}\left(\mathbf{L}_{A \cup B}(\Theta)\right)^{-1} \frac{\partial \mathbf{L}_{A \cup B}(\Theta)}{\partial \Theta} \\ &- \sum_{u \in \mathbb{U}} \sum_{S \in \mathcal{D}_u} \operatorname{tr}\left(\mathbf{L}(\Theta) + \mathbf{I}_{\bar{A}}\right)^{-1} \frac{\partial \mathbf{L}(\Theta)}{\partial \Theta}\end{aligned} \tag{6.13}$$

For brevity, we drop the superscript ($\boldsymbol{S}$) of the sequence specified SDPP kernel.

According to the quality-diversity decomposition of structure formulation (described in Section 6.3.3.1), the entry of SDPP kernel is

$$L_{ab} = \exp\left(w(S_a)\right) Sim(S_a, S_b) \exp\left(w(S_b)\right). \tag{6.14}$$

As a structure is regarded as an element in SDPP, we can directly denote the quality of S_a as q_a and compute the gradient with respect to the quality,

$$\frac{\partial L_{ab}}{\partial q_a} = Sim(S_a, S_b) q_b. \tag{6.15}$$

Here, q_b represents the quality value of the temporal set S_b. In Equation 6.14, we derive the similarity measure $Sim(S_a, S_b)$ using the diverse kernel $\mathbf{K}$, as outlined in Equation 6.8. This kernel has been pre-learned and does not require further learning during the STSP optimization process. Furthermore, the qualities of the structures (S_a and S_b) are computed by accumulating the weights of their elements as shown in Equation 6.5, a simple process that ensures all components are differentiable and thus allows for the computation of gradients at each location [Affandi et al., 2014]. We can simplify the computation of the gradient of q_a with respect to its parameters within the SDPP kernel by applying the chain rule to decompose the quality weight for calculating gradients of each component [Chao et al., 2015]. These parameters, which include parameterized preference and co-occurrence representations in Equation 6.6 and Equation 6.7, respectively, form the SDPP objective function, denoted as θ_a, and $\frac{\partial q_a}{\partial \theta_a} = \exp\left(w\left(S_a\right)\right) \cdot \frac{\partial w(S_a)}{\partial \theta_a}$. Substitute decomposed gradients back into the previous gradient formula Equation 6.13, we can receive the final results. That is, SDPP objective function are differentiable with respect to its parameters. To obtain the gradients of entire model parameters, the attractive backpropagation algorithm [Rosenblatt, 1961] can be applied. Finally, the gradient-based algorithms are allowed to be applied for optimization. In experiments, Adam (a variant of stochastic gradient descent) [Kingma and Ba, 2014] is used.

Table 6.2: Statistics of the datasets (TaoBao, JD-buy, JD-add, and TaFeng).

Dataset	#Users	#Items	#Sets	#Categories
TaoBao	12.0K	14.2K	85.9K	68
JD-buy	13.5K	19.6K	76.8K	77
JD-add	3.4K	12.1K	16.1K	75
TaFeng	10.3K	8.7K	73.3K	650

6.3.3.4 Predictions

Now we can introduce how to synthetically apply learned co-occurrence representation $\mathbf{C}^{(\boldsymbol{S})}$ of entire items and sequence preference representation $\mathbf{p}^{(\boldsymbol{S})}$ for predicting the subsequent set of temporal sets sequence $\boldsymbol{S}$. For an item v_i, the final predicted score for evaluation is

$$\hat{\boldsymbol{y}}_i = (1-\lambda)\mathbf{p}^{(\boldsymbol{S})^\top}\mathbf{e}_i^P + \lambda\frac{1}{|V|}\sum_{n=1}^{|V|}\mathbf{c}_i^{(\boldsymbol{S})^\top}\mathbf{c}_n^{(\boldsymbol{S})}. \tag{6.16}$$

the co-occurrence score of v_i in the subsequent set given previous sequence is measured by averaging the co-occurrence strength between v_i and any one of other items in $\mathbb{V}$. In this setting, we expect the cohesion score for a target item v_t in the subsequent set to be relatively high. This is because the co-occurrence representations (CORs) are learned to reflect co-occurrence patterns, thereby enhancing the cohesion strength between v_t and other target items of the next set. These cohesion scores are aggregated as shown in Equation 6.16. Furthermore, we use the parameter λ to balance the contribution of co-occurrence and preference scores when evaluating predictions.

6.4 Experiments

This section comprehensively evaluates and compares STSP. To verify the importance of three proposed relationships, specific experiments are designed.

6.4.1 Experimental Settings

6.4.1.1 Datasets

Experiments are conducted on four real-world datasets from three application scenarios: *i*) **TaoBao**[1] denotes a public online e-commerce dataset provided by Ant Financial services. It contains the interactions about which items are purchased or clicked by each user with the timestamp. The items purchased in the same day are treated as a set; *ii*) **JingDong**[2] is comprised of user action records, including browsing, purchasing, following, commenting and adding to shopping carts. Two datasets are extracted from

[1] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
[2] https://jdata.jd.com/html/detail.html?id=

JingDong, that is, **JD-buy** and **JD-add**, which contain purchasing and adding to shopping carts records respectively. We treat all the items purchased/added in the same day as a set; *iii*) **TaFeng**[3] dataset contains the transactions about which items are bought by each customer in each basket with the time stamp. We treat products purchased in the same day by the same customer as a set. The statistics of all the datasets are shown in Table 6.2.

For each user in the dataset, we can get a series of training instances. As indicated in Figure 6.3, a training set instance (*i.e.*, the sequence specified ground set with previous sets, target sets, and randomly constructed negative sets) for the SDPP objective function is comprised of previous sets, target sets, and negative sets, whose sizes (number of sets) are denoted using A, B, and Z, respectively, in subsequent discussions. If a temporal set contains only one item, the edge weight will be omitted. Following previous TSP work [Yu et al., 2020a; Sun et al., 2020c], each dataset is divided into three parts. The last set, the second last set, and the remaining sets of each user are used for testing, validation, and training, respectively.

6.4.1.2 Baselines.

To validate the superiority of our framework, multiple state-of-the-art models built based on different techniques are selected as baselines, which are related to this work and are competitive. **DREAM** [Yu et al., 2016] is a RNN-based method that uses a pooling layer to represent a basket for next basket recommendation. **CDSL** [Liu et al., 2022] uses the standard DPP likelihood as the loss function of sequential recommendation. To implement CDSL for TSP, we treat the preference representation and kernel **K** of STSP as the quality and diversity components of DPP in CDSL. **FPMC** [Rendle et al., 2010] designs a hybrid model to take temporal dependencies between items and user's general interests into account by combining markov chain and matrix factorization for next basket recommendation. **DIN** [Zhou et al., 2018] proposes a DNN-based method used for click-though rate prediction, which uses attention mechanism to learn the representations of candidate items *w.r.t.* historical behaviors. **Sets2Sets** [Hu and He, 2019] is the seminal work for temporal sets prediction. The average pooling operation is used to obtain the representations for sets. **DSNTSP** [Sun et al., 2020c] designs a dual attention-based network to learn item-level and set-level representations of user sequences for TSP. **ETGNN** [Yu et al., 2022] is a competitive TSP work. It uses graph neural network to propagate set-level information for obtaining collaborative signals.

[3]https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset

6.4.1.3 Configuration.

We adopt PyTorch to implement our framework. Adam is adopted as the optimizer in our experiments. The number of heads (H) in our representation learning module is set to 4, and 4 trainable queries (K) are used. We tune the hyper parameters in all the compared methods with grid search to choose the best performance for comparison. Some important parameters, such as weight λ in Equation 6.16, targets number B, and previous sets length A, are further analyzed in the following content.

6.4.1.4 Evaluation Metrics.

To comprehensively evaluate the quality and diversity of ITSP, three groups of evaluation metrics are adopted: (*i*) **Accuracy**. We evaluate the accuracy of set prediction using Recall@N and NDCG@N. To check the relevance of the predicted set in different ranking positions of the ranking list, we evaluate $N = 20$ and $N = 80$; (*ii*) **Diversity**. Category coverage (CC) [Wu et al., 2019b; Puthiya Parambath et al., 2016] is a widely used diversity evaluation metric, which is appropriate to the diversity concept in recommendation diversification area. A higher value of category coverage means that predictions are more diverse; (*iii*) **Trade-off**. To evaluate the balance performance between accuracy and diversity, the harmonic metric F-score (F1) is also used [Cheng et al., 2017; Liu et al., 2020]. F1@$N = 2 \times$ Q@$N \times$ CC@N / (Q@N + CC@N), where Q@N is the averaged value of the two accuracy metrics at N.

6.4.2 Experimental Results

The overall comparison is reported in Table 6.3. The best performance *w.r.t.* an evaluation metric among all methods on a dataset is in bold. Meanwhile, the highest result from baselines of a metric is marked with a superscript *. The relative improvement (*Improv*) of STSP over the highest performance out of baselines is also listed. The main observations obtained from the overall comparison are:

- In most cases, the improvements of quality performance (Recall and NDCG) over the best baselines are greater than 10% (some cases even achieve $> 20\%$, *e.g.*, Re@80 on TaoBao and NDCG@50 on JD-add). The convincing outcomes of our experiments affirm the value of our primary innovations, including the direct modelling of temporal sets, the formulation of a specialized objective function, and the exploration of complex interrelationships. These results provide compelling evidence that our proposed STSP contributes significantly to improved predictive performances.

- Aligning with the results of four datasets, we can find that the improvements on TaoBao and JingDong (especially JD-add) datasets achieved by STSP are more sig-

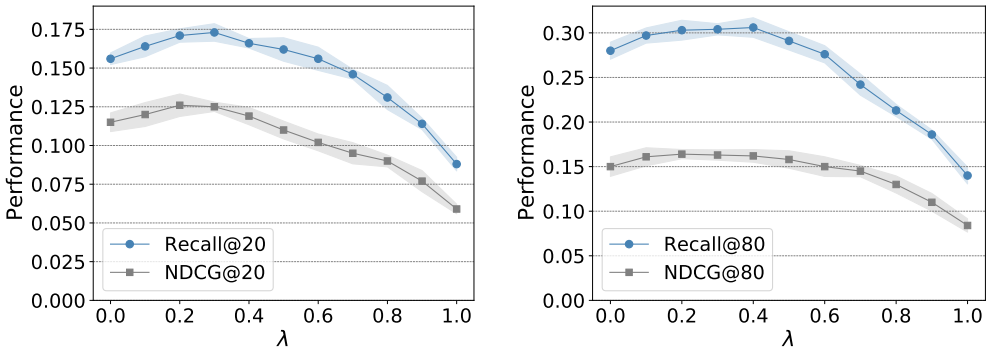
Table 6.3: Overall performance comparisons. The bold values and superscripts denote the best performances among all approaches and all baselines, respectively

Dataset	Method	Quality				Diversity		Trade-off	
		Recall@20	Recall@80	NDCG@20	NDCG@80	CC@20	CC@80	F1@20	F1@80
TaoBao	DREAM	0.1053	0.2109	0.0885	0.1140	0.0905	0.2572	0.0936	0.1991
	CDSL	0.1306	0.2206	0.1092	0.1291	0.1253*	0.2999*	0.1225	0.2181
	FPMC	0.1032	0.2087	0.0850	0.1098	0.0912	0.2692	0.0926	0.2001
	DIN	0.1160	0.2169	0.0915	0.1218	0.1001	0.2528	0.1019	0.2028
	Sets2Sets	0.1286	0.2231	0.1001	0.1369	0.0986	0.2607	0.1059	0.2129
	ETGNN	0.1503*	0.2453*	0.1170*	0.1426	0.1104	0.2544	0.1209	0.2201
	DSNTSP	0.1457	0.2296	0.1095	0.1495*	0.1187	0.2997	0.1230*	0.2322*
	ITSP	**0.1716**	**0.3039**	**0.1247**	**0.1622**	**0.1360**	**0.3049**	**0.1418**	**0.2642**
	Improv (%)	14.17	23.89	6.58	8.49	8.54	1.74	15.31	13.76
JD-add	DREAM	0.1047	0.1692	0.0810	0.1014	0.1105	0.2436	0.1009	0.1740
	CDSL	0.1107	0.1834	0.0957	0.1159	0.1118*	0.2625*	0.1073	0.1906
	FPMC	0.0973	0.1606	0.0737	0.0934	0.0937	0.2399	0.0894	0.1661
	DIN	0.1027	0.1651	0.0794	0.0985	0.1067	0.2541	0.0983	0.1736
	Sets2Sets	0.1085	0.1731	0.0843	0.1036	0.0918	0.2458	0.0940	0.1770
	ETGNN	0.1306*	0.1959	0.1027	0.1243	0.1085	0.2476	0.1124	0.1945
	DSNTSP	0.1291	0.2004*	0.1206*	0.1358*	0.1112	0.2621	0.1154*	0.2048*
	ITSP	**0.1640**	**0.2358**	**0.1430**	**0.1592**	**0.1125**	**0.2650**	**0.1298**	**0.2263**
	Improv (%)	25.57	17.66	18.57	17.23	0.72	0.95	12.55	10.49
JD-buy	DREAM	0.3122	0.3810	0.2157	0.2601	0.0942	**0.2563***	0.1388	0.2848
	CDSL	0.3106	0.4110	0.2510	0.2731	0.0954*	0.2455	0.1424	0.2858
	FPMC	0.3015	0.3902	0.2172	0.2519	0.0929	0.2276	0.1368	0.2664
	DIN	0.3298	0.4062	0.2453	0.2700	0.0913	0.2428	0.1386	0.2826
	Sets2Sets	0.3158	0.4013	0.2312	0.2587	0.0950	0.2365	0.1410	0.2755
	ETGNN	0.3325	0.4276*	0.2637*	0.2810	0.0914	0.2303	0.1399	0.2791
	DSNTSP	0.3406*	0.4254	0.2576	0.2874*	0.0947	0.2461	0.1439*	0.2912*
	ITSP	**0.3812**	**0.4760**	**0.2804**	**0.3114**	**0.0974**	0.2544	**0.1505**	**0.3091**
	Improv (%)	11.92	11.32	6.33	8.35	2.10	-0.74	4.61	6.16
TaFeng	DREAM	0.0593	0.1308	0.0512	0.0702	0.0210	0.0785*	0.0304	0.0881
	CDSL	0.0682	0.1411	0.0548	0.0741	0.0220*	0.0764	0.0324*	0.0894
	FPMC	0.0617	0.1215	0.0516	0.0696	0.0206	0.0790	0.0302	0.0865
	DIN	0.0652	0.1348	0.0497	0.0715	0.0221	0.0782	0.0319	0.0890
	Sets2Sets	0.0648	0.1445	0.0509	0.0754	0.0217	0.0711	0.0316	0.0864
	ETGNN	0.0718*	0.1490	0.0562*	0.0804*	0.0197	0.0692	0.0301	0.0863
	DSNTSP	0.0676	0.1533*	0.0531	0.0787	0.0213	0.0770	0.0315	0.0926*
	ITSP	**0.0814**	**0.1675**	**0.0630**	**0.0883**	**0.0223**	**0.0793**	**0.0341**	**0.0979**
	Improv (%)	13.37	9.26	12.10	9.83	0.91	0.38	5.15	5.80

nificant compared against those on TaFeng. This disparity could be attributed to varying degrees of cohesion within sets and dependencies across them. It is understandable that the purpose of browsing/purchasing in online shopping platforms is more specific than visiting a store, as a customer's shopping list for a retail store visit is typically plentiful, which means that the stronger cohesion among a set (transaction) exists on TaoBao and JingDong datasets than TaFeng. In addition, the dependency between different visits to a store is relatively loose compared to opening an app for shopping. For example, it is likely that a user who purchased an iPhone will order a screen protector and silicone case in the following day. These specific relationships (co-occurrence and dependency) modeled by STSP enhance the performance improvements on TaoBao and JingDong datasets, further verifying the contribution of defining and capturing different types of relationships.

- When comparing the improvements in Recall and NDCG across four datasets, it is apparent that the performance of STSP in terms of recall is notably superior to its NDCG performance. This can be attributed to the incorporation of co-occurrence representations in our model. Co-occurrence patterns, by evaluating cohesion among candidate items, rather than relevance between user sequence preference and candidates, are effective at supplementing *potential items* for final predictions. Here, *potential items* refer to relevant items that may be inadvertently missed by preference prediction yet are indeed positive items. These elements, due to their strong cohesive relationships with relevant items that are already identified by preference prediction, can be integrated into the same cohesive set, and thus facilitates recalling (complementing) relevant items. This performance improvement highlights the significance of the co-occurrence patterns we introduced in STSP, and provides clear evidence of their effectiveness in enhancing recall performance.

- STSP exhibits superior diversity performance (CC) compared to the baselines (including CDSL that considers the diversity of individual items) in the majority of cases (except CC@80 on JD-buy). This highlights the effectiveness of utilizing set-level diversity measurements for TSP tasks. Notably, promoting the diversity of the subsequent set holds significant practical value. For instance, in a real-world application scenario such as a retail store, diversifying predictions enables the store to offer bundles with a variety of products, thereby increasing consumers' potential purchasing desire. Furthermore, our method exhibits advantages in both quality and diversity when comparing the overall metric F1. This demonstrates the comprehensive effectiveness of our approach in balancing the relevance and diversity aspects of predictions. This finding further solidifies the potential and value of our method in enhancing TSP performance by offering both relevance and diverse predictions



(a) Top-20 on TaoBao

(b) Top-80 on TaoBao

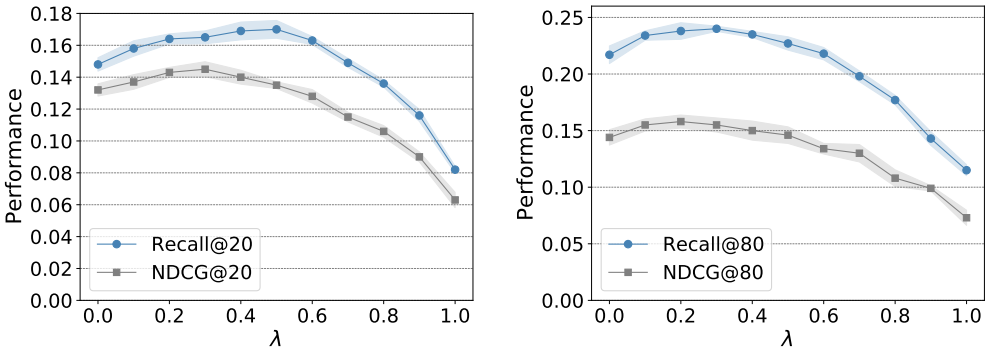
Figure 6.4: Performance trends with parameter λ on JD-add. Error bars denote 95% confidence intervals

to users.

- Among the seven baselines, the models especially recent proposed ETGNN and DSNTSP dedicated to temporal sets prediction outperform other approaches. This observation emphasizes the distinct and intricate nature of TSP as a unique task. It highlights the significance of considering special relationships within temporal sets for accurate prediction. Our model delves into exploring and leveraging these relationships, enabling us to capture the temporal dependencies and patterns inherent in the sets.

Table 6.3 has shown the superiority of our proposed framework from multiple aspects. We now investigate the contributions of three types of proposed relationships (*i.e.*, sequence preference correlation, co-occurrence pattern, and temporal sets dependency) by regulating specific hyper-parameters. We first conducting experiments with different values of λ to analyze the importance and effectiveness of preference correlation and co-occurrence pattern. Figure 6.4 and Figure 6.5 shows the quality performance (recall and NDCG) trends with parameter λ introduced in Equation 6.16 on TaoBao and JD-add. We need to describe two extreme cases at first, *i.e.*, $\lambda = 0$ and $\lambda = 1$. In the case of $\lambda = 1$, only co-occurrence score is used for calculating prediction score, and thus the learning of preference representation is deprecated from STSP. Namely, the preference correlation is not captured. When $\lambda = 0$, the co-occurrence score is not activated, and the learning of co-occurrence representations is not involved in the training process, *i.e.*, the weight of a set (structure) is represented only using relevance.

We aim to use Figure 6.4 and Figure 6.5 to analyze the following two aspects:



(a) Top-20 on JD-add

(b) Top-80 on JD-add

Figure 6.5: Performance trends with parameter λ on JD-add. Error bars denote 95% confidence intervals

Importance of Co-occurrence. When λ equals 0 (without using co-occurrence), STSP does not reach its full capacity. With the joining of co-occurrence ($\lambda > 0$), impressive improvements are achieved, and STSP gradually reaches the peak (when λ is around 0.4) of quality performance. These improvements are obvious evidence of co-occurrence pattern's importance for TSP. As the involvement of co-occurrence becomes deeper (λ>0.4), the prediction performance declines. This is mainly because that an item's co-occurrence score is determined by calculating its co-occurrence patterns with all other items in $\mathbb{V}$. As the parameter value increases, the co-occurrence score becomes more susceptible to the influence of unrelated items, which in turn, adversely affects the final prediction scores. This adverse effect can potentially overshadow the relevance aspect of the predictions, resulting in a decrease in overall effectiveness. However, it is important to note that we cannot dismiss the role of co-occurrence due to the subsequent decline in performance, as the significant improvement in the earlier stage has already demonstrated the effectiveness and importance of co-occurrence pattern.

Importance of Preference. We can treat the performances in the cases of $\lambda = 0$ and $\lambda = 1$ as the capabilities of preference and co-occurrence respectively, as only one of them is involved in STSP in the corresponding case. We can see that the preference representations dominate the quality performance of STSP, which is a quite common phenomenon in sequence prediction tasks [Sun et al., 2020c]. This is also the reason why we mention that the co-occurrence pattern is used to complement items for prediction instead of playing a leading role. While co-occurrence information provides valuable insights, it is not as influential as preference representations in terms of improving the quality of predictions. This can further explain the reason of subsequent decline, when the λ value becomes excessively large, the co-occurrence component gradually outweighs the preference component, leading to a decline in prediction performance.

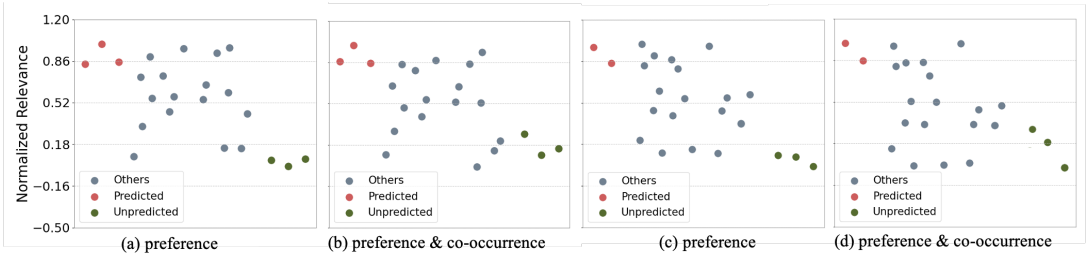
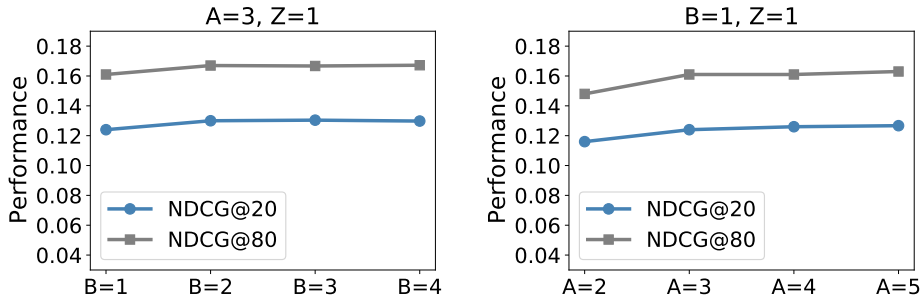


Figure 6.6: Examples to demonstrate the effectiveness of co-occurrence representations.



(a) targets number B (b) previous sets length A

Figure 6.7: Performance trends *w.r.t.* sequence dependency.

As the trends on CC with λ is not quite obvious and similar trends can be found on other situations (Top-50 quality performance, and TaFeng and JD-buy), we omit corresponding presentations.

To further demonstrate the role of co-occurrence in predicting and completing the subsequent set, we employ two examples presented in Figure 6.6 to provide a clear explanation. We randomly choose one suitable user from each of two datasets (TaoBao and JD-add), ensuring that they have at least 10 sets, with the final test set containing a minimum of 5 items. The four subplots presented are organized to clearly illustrate the relevance of items in our model. Each point within the subplots represents an item, with the vertical axis denoting the normalized prediction score, which ranges from 0 to 1 for better visualization of trends. There are three distinct groups of points: leftmost points represent positive items in the test subsequent set ranking in the top 20 positions (*i.e.*, *predicted* targets by our model in Top-20 rankings), middle points correspond to negative items (*others*) appearing among the top 20, and rightmost points indicate positive items that are not included in Top-20 results (*i.e.*, *unpredicted* items).

Figure 6.6(a) and Figure 6.6(c) each depicts the ranking examples for the two selected users from the TaoBao and JD-add datasets, respectively, when only preference representations are utilized. Figure 6.6(b) and Figure 6.6(d) correspond to the ranking results for the two users (from TaoBao and JD-add) after integrating both preference

and co-occurrence representations. Since the composition of the target set is known in these two examples, we only need to evaluate the co-occurrence patterns of desired structure (cohesion among *unpredicted* and *predicted* items) and undesired structure (cohesion among *others* and *predicted* items). Therefore, in Figure 6.6(b) and Figure 6.6(d), the final scores consist of each item's preference score combined with its averaged co-occurrence scores in relation to the *predicted* items. To show the overall change, the final score of a *predicted* item is also integrated with co-occurrence factors *w.r.t.* other *predicted* ones. Significant improvements in the rankings of true items (*predicted* and *unpredicted*) are observed after including co-occurrence scores, compared to the rankings of other items in both sampled sets. These intuitive results indicate that our model is capable of assigning the relatively higher co-occurrence scores to items within the same set via effectively capturing the co-occurrence patterns of the true sets. The role of co-occurrence is indeed important in our framework. In essence, the co-occurrence patterns effectively bridge the gap between individual items by quantifying the underlying relationships among items in the same set more accurately.

Figure 6.7 is used to analyze the **importance of temporal sets dependency**, in which NDCG performance trends of Top-20 and Top-80 on TaoBao with varying B and A introduced in Equation 6.10 are presented. A consistent trend can be observed in both sub-figures of Figure 6.7, *i.e.*, NDCG performances (NDCG@20 and NDCG@80) improve at first and then tend towards stability with the increase of B and A when fixing the other two factors (A and Z in Figure 6.7(a), and B and Z in the other subfigure). This trend suggests that by considering more target sets or previous sets in our specialized sets-level objective function, we capture more dependencies across the sets sequence, leading to substantial performance improvements. This effectively validates the importance of sequence dependencies for TSP and demonstrates that our proposed objective function is efficacy at capturing these dependencies. We only display the performance trends *w.r.t.* NDCG on TaoBao dataset, as the corresponding recall performance trends on other datasets also arrive at the similar conclusion.

To demonstrate the generality and applicability of STSP framework, results on different datasets displayed in Table 6.3 are all obtained with $\lambda = 0.2, B = 1, A = 3,$ and $Z = 1$. We also compare the training efficiency between STSP and two representative baselines (CDSL and DSNTSP). We perform experiments on TaoBao using a single Tesla K40 GPU under the same setting (*e.g.*, batch size, sequence length, optimizer, etc.), the average training time (in seconds) per epoch of CDSL, DSNTSP, and STSP is 102s, 98s, and 107s, respectively. Compared to significant performance improvements ($> 10\%$ in most cases), the gap of training time between STSP and DSNTSP is relatively modest ($< 10\%$). This conclusion with respect to training efficiency is also obtained when performing experiments on other datasets.

6.5 Conclusion

In this chapter, we focus on addressing the problem of temporal sets prediction via untangling naturally complicated relationships by means of SDPP. To achieve this, three types of relationships, *i.e.*, preference correlation, co-occurrence pattern, and temporal sets dependency, are proposed and formulated. By viewing temporal sets as integrated elements and structuring them using SDPP models, we designed a tailored sets-level objective function based on the conditional SDPP distribution. This innovative method of treating temporal sets as unified entities, combined with our tailored objective function, led to the development of the STSP framework, the effectiveness of which is highlighted through robust experimental outcomes.

Our experiments also provided valuable insights into the significance of preference correlations, co-occurrence patterns, and temporal sets dependencies, by adjusting the parameter λ and varying the length of the training set sequences. Experimental results and analysis underscore the impact of these relationships in enhancing the performance of temporal sets prediction tasks. Looking ahead, we believe that our framework could potentially be extended to other research areas that are hindered by complex structural constraints. For example, video summarization [Gong et al., 2014] and bundle list recommendation [Bai et al., 2019], where the ability to capture and understand intricate relationships within structured data is crucial, could greatly benefit from the approach outlined in this study. Finally, we recognize the potential of the co-occurrence representations that show considerable promise in our TSP experiments. This prompts a call for further exploration and refinement of the formulation and use of co-occurrence representations, which could potentially unlock even greater performance improvements in temporal sets prediction tasks.

6.5.1 Limitations

While the framework proposed in this chapter is both effective and novel, it is not without limitations, including: (i) our method for learning preference representation is primarily based on the self-attention framework, without a deep integration with the SDPP structure; (ii) the exploration and utilization of co-occurrence representation remain limited, not fully leveraging the potential impact of the co-occurrence concept on temporal sets prediction tasks.

6.5.2 Future Work

Accordingly, we propose two specific avenues for future work. First, we design a new targeted sequence representation framework that can integrate more seamlessly with

the SDPP structure, enhancing predictive performance. On the other hand, further exploration into the impact and limitations of co-occurrence on temporal sets prediction tasks will be undertaken, thereby advancing the learning and effectiveness of co-occurrence representation.

Chapter 7

Conclusions and Future Work

In this thesis, we first summarize comparisons among three fields of producing personalized and diversified predictions, *i.e.*, traditional recommendation, sequential recommendation, and temporal sets prediction, and then explore new perspectives on these three fields on basis of specific DPP models in accordance with the inherent nature of different fields. The superiority of these perspectives are validated on real-world datasets. What is more, this thesis pave the way for future research directions with regard to employing DPP for diversification. This chapter is outlined according to the title of this chapter, *i.e.*, detailed conclusions and future work directions.

7.1 Conclusions

We first present the introduction and related work in Chapter 1 and Chapter 2 respectively. Then four original projects constitute the remaining body of this thesis, whose conclusions can be summarized as follows:

- Chapter 3 presents a **diversity-aware GAN framework**, in which diversity is considered in both training and serving procedures. In the training procedure, diversity is reflected from two aspects: (*i*) the diversity of a generated item set from generator is measured and maximized via the formulation of DPP likelihood; (*ii*) the common manner of providing items for discriminator according to relevance rankings of generator model is replaced by the DPP MAP inference that balances quality and diversity, which facilitates discriminator in learning to distinguish diversified suggestions and then enhances generator's ability to generate accurate and diverse items. To prevent overemphasis on maximizing diversity, we design an accuracy-promoting objective (*i.e.*, minimizing recommendation errors) to complement the diversity maximization objective function. Thus, the two objectives of diversity-promoting recommendation are directly formulated. To guarantee providing users with diversified recommendations, a MAP inference on conditional DPP distribution is proposed in serving/evaluating procedure to employ

the learned diversity-aware embeddings and capture dependency between historical interactions and predictions. By performing ablation experiments, we find that the combination of these designs is superior to other baselines and variants on both diversity and accuracy.

- Capturing the dependency across the sequence plays a key role in modeling users' dynamic preferences for sequential recommendation. We first **introduce dependency into objective functions** in Chapter 4. Two types of dependencies, *targets dependency* and *sequence dependency*, are defined, which capture dependency within targets (next items used in loss calculation) and across sequence (dependency between previous items and targets in a training sequence), respectively. We employ the likelihoods of standard DPP and conditional DPP distributions to formulate these two types of dependencies. Based on which, the goal of sequential recommendation is transformed into maximizing the probability of selecting the targets or the training sequence as a DPP-distributed subset. We design specific experimental settings to validate the importance of dependencies in objective functions. The proposed objective functions can be easily applied in existing sequential models, indicating their important potential applicability.

- We pave the way for **exploiting k-DPP to achieve ranking optimization** in Chapter 5. We adapt the k-DPP distribution to endow subsets of cardinality k with a ranking interpretation through their probabilities. This enables us to enhance the ranking of target subsets and lower negative subset's ranking, ultimately guiding item recommendation models to learn personalized latent factors. We propose two types of ranking objective, one only enhancing target subset's ranking and the other combining the promotion (target subset) and demotion (negative subset). We empirically analyze and experimentally validate why the specialized ground set ($k + n$ ground set) and normalization are necessary. The ranking interpretation is also demonstrated by toy examples.

- We **bring SDPP to the field of temporal sets prediction** as described in Chapter 6. Based on SDPP, the temporal set can be explicitly viewed as a structure of SDPP. Three types of relations, *preference correlation*, *co-occurrence pattern*, and *temporal sets dependency*, are then defined, The first two relations, captured by the common preference representations and the new proposed co-occurrence representations, respectively, are incorporated into the SDPP structure. As the SDPP structure simplifies the calculation and representation of the temporal set, we therefore can directly treat a temporal set as a comparable element in loss calculation and then define a SDPP-level objective function that

captures the third type of relations. The notable gains in experimental results show the prospects of our SDPP-based prediction framework. The importance of different relations are individually demonstrated by specific settings.

7.2 Future Work

This thesis provides new insights into the task of producing personalized and diversified predictions, and the corresponding potential future research directions can be summarized as follows:

- In the recommendation scenario, diversity and relevance are two types of destinations, *i.e.*, to expand the scope of recommendations' topics and to recommend accurate results. Existing work (not only related to recommendation task [Wu et al., 2019b; Chen and Ahmed, 2020] but also to text summarization [Gong et al., 2014] or Web search [Kulesza et al., 2012]) has suggested that diversification is usually improved by sacrificing accuracy, as they focus on different objectives. This means that we can think that there is a competition relationship between these two objectives. In Chapter 3, we directly formulate two cooperative objectives to represent two destinations. However, the competition relationship is not well captured. **Building a minimax game between quality and diversity** referring to the adversarial mechanism of GAN can be an interesting research problem, which unifies two schools of models (relevance recommendation and diversity recommendation) into an overall objective. Towards this research direction, relevance and diversity may potentially be further boosted through adversarial competition.

- The use of contrastive learning for recommendation has been a trending technique to improve recommendation performance [Shuai et al., 2022; Xie et al., 2022; Wei et al., 2021], as training instances are augmented by the data augmentation strategy and the objective of contrastive learning can provide new insights into the embedding representations. However, current data augmentation strategies often generate augmentations that closely resemble the actual training instances (*e.g.*, observed historical interaction sequences). In this setting, it is likely that recommendation models become more accuracy-guided (*i.e.*, prefer accuracy and similar suggestions) than traditional methods. **Designing a contrastive learning approach that balances similarity and diversity of augmentations** still remains an open research direction, which can be implemented by a specialized DPP distribution that is capable of sampling diverse and relevant training instances for contrastive learning.

- k-DPP and SDPP are employed individually in Chapter 5 and Chapter 6. The connections between these two extended models of DPP are not explored. There will probably be wider potential if we **combine k-DPP and SDPP**. For example, a set-level ranking optimization algorithm can be developed to retrieve structured information, such as answer selection [Lai et al., 2018] and image retrieval [Datta et al., 2008].

- In Chapter 6, a structured framework is built by exploiting SDPP for temporal sets prediction, in which formulating a new relationship, namely the co-occurrence pattern, is a key contribution. However, there remain open research questions regarding the definition of edge weight, which would facilitate **mining new representation forms of structures**. In addition, SDPP can also be applied to other tasks with complex structures, and then makes contributions on modeling structures. For example, the focus of edge weight for sign language translation [Camgoz et al., 2018] and generative language model [Radford et al., 2018] could be context and semantic. By formulating the edge weight of SDPP structures according to specific tasks, a chance of boosting performance can be offered.

Bibliography

Affandi, R. H.; Fox, E.; Adams, R.; and Taskar, B., 2014. Learning the parameters of determinantal point process kernels. In *International Conference on Machine Learning*, 1224–1232. PMLR. (cited on pages 17, 75, 103, and 104)

Agrawal, R.; Gollapudi, S.; Halverson, A.; and Ieong, S., 2009. Diversifying search results. In *Proceedings of the second ACM international conference on web search and data mining*, 5–14. (cited on page 20)

Arjovsky, M.; Chintala, S.; and Bottou, L., 2017. Wasserstein GAN. (cited on page 21)

Ashkan, A.; Kveton, B.; Berkovsky, S.; and Wen, Z., 2015. Optimal greedy diversity for recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*. (cited on pages 1, 2, 15, 20, and 33)

Bai, J.; Zhou, C.; Song, J.; Qu, X.; An, W.; Li, Z.; and Gao, J., 2019. Personalized bundle list recommendation. In *The World Wide Web Conference*, 60–71. (cited on pages 1, 18, and 114)

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N., 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28 (2015). (cited on page 16)

Berg, R. v. d.; Kipf, T. N.; and Welling, M., 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, (2017). (cited on pages 31, 32, 75, and 76)

Borodin, A. and Rains, E. M., 2005. Eynard–Mehta theorem, Schur process, and their Pfaffian analogs. *Journal of statistical physics*, 121, 3 (2005), 291–317. (cited on page 47)

Calandriello, D.; Derezinski, M.; and Valko, M., 2020. Sampling from a k-DPP without looking at all items. *Advances in Neural Information Processing Systems*, 33 (2020), 6889–6899. (cited on pages 17 and 18)

Camgoz, N. C.; Hadfield, S.; Koller, O.; Ney, H.; and Bowden, R., 2018. Neural sign language translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7784–7793. (cited on page 119)

Cao, Z.; Qin, T.; Liu, T.-Y.; Tsai, M.-F.; and Li, H., 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, 129–136. (cited on pages 13, 44, and 61)

Carbonell, J. and Goldstein, J., 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 335–336. (cited on pages 15 and 20)

Chao, W.-L.; Gong, B.; Grauman, K.; and Sha, F., 2015. Large-Margin Determinantal Point Processes. In *UAI*, 191–200. (cited on page 104)

Chen, L.; Wu, L.; Hong, R.; Zhang, K.; and Wang, M., 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 27–34. (cited on pages 13 and 64)

Chen, L.; Wu, L.; Zhang, K.; Hong, R.; and Wang, M., 2021. Set2setrank: collaborative set to set ranking for implicit feedback based recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 585–594. (cited on pages 13, 15, 61, 75, and 81)

Chen, L.; Zhang, G.; and Zhou, H., 2017. Improving the diversity of top-N recommendation via determinantal point process. In *Large Scale Recommendation Systems Workshop*. (cited on page 3)

Chen, L.; Zhang, G.; and Zhou, H., 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 5627–5638. (cited on pages 2, 3, 5, 15, 16, 20, 21, 23, 26, 28, 29, 31, 33, 34, and 42)

Chen, W. and Ahmed, F., 2020. PaDGAN: A Generative Adversarial Network for Performance Augmented Diverse Designs. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 84003, V11AT11A010. American Society of Mechanical Engineers. (cited on pages 23, 25, 28, 33, and 118)

Chen, Z.; Xiao, T.; and Kuang, K., 2022. Ba-gnn: On learning bias-aware graph neural network. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 3012–3024. IEEE. (cited on page 75)

Cheng, P.; Wang, S.; Ma, J.; Sun, J.; and Xiong, H., 2017. Learning to recommend accurate and diverse items. In *Proceedings of the 26th international conference on World Wide Web*, 183–192. (cited on pages 2, 5, 15, 20, 31, 50, 76, and 107)

Chirita, P.-A.; Diederich, J.; and Nejdl, W., 2005. MailRank: using ranking for spam detection. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, 373–380. (cited on page 86)

Chiu, C.-C.; Sainath, T. N.; Wu, Y.; Prabhavalkar, R.; Nguyen, P.; Chen, Z.; Kannan, A.; Weiss, R. J.; Rao, K.; Gonina, E.; et al., 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4774–4778. IEEE. (cited on page 16)

Choi, E.; Bahadori, M. T.; Schuetz, A.; Stewart, W. F.; and Sun, J., 2016. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine learning for healthcare conference*, 301–318. PMLR. (cited on pages 9, 59, and 90)

Covington, P.; Adams, J.; and Sargin, E., 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, 191–198. (cited on pages 4 and 14)

Dai, B.; Shen, X.; Wang, J.; and Qu, A., 2021. Scalable Collaborative Ranking for Personalized Prediction. *Journal of the American Statistical Association*, (2021), 1215–1223. (cited on page 1)

Datta, R.; Joshi, D.; Li, J.; and Wang, J. Z., 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40, 2 (2008), 1–60. (cited on page 119)

Davidson, J.; Liebald, B.; Liu, J.; Nandy, P.; Van Vleet, T.; Gargi, U.; Gupta, S.; He, Y.; Lambert, M.; Livingston, B.; et al., 2010. The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, 293–296. (cited on page 9)

Deshpande, M. and Karypis, G., 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22, 1 (2004), 143–177. (cited on pages 9 and 39)

Dhoolia, P.; Kumar, V.; Contractor, D.; and Joshi, S., 2021. Bootstrapping Dialog Models from Human to Human Conversation Logs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 16024–16025. (cited on page 16)

Ebesu, T.; Shen, B.; and Fang, Y., 2018. Collaborative memory network for recommendation systems. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, 515–524. (cited on page 31)

Fang, H.; Zhang, D.; Shu, Y.; and Guo, G., 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)*, 39, 1 (2020), 1–42. (cited on pages 1, 5, 12, and 39)

Feng, S.; Li, X.; Zeng, Y.; Cong, G.; Chee, Y. M.; and Yuan, Q., 2015. Personalized ranking metric embedding for next new poi recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*. (cited on page 89)

Gan, L.; Nurbakova, D.; Laporte, L.; and Calabretto, S., 2020. Enhancing recommendation diversity using determinantal point processes on knowledge graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001–2004. (cited on pages 2, 3, 5, 15, and 16)

Gao, R.; Xia, H.; Li, J.; Liu, D.; Chen, S.; and Chun, G., 2019. DRCGR: Deep reinforcement learning framework incorporating CNN and GAN-based for interactive recommendation. In *2019 IEEE International Conference on Data Mining (ICDM)*, 1048–1053. IEEE. (cited on page 13)

Gartrell, M.; Brunel, V.-E.; Dohmatob, E.; and Krichene, S., 2019. Learning nonsymmetric determinantal point processes. *arXiv preprint arXiv:1905.12962*, (2019). (cited on page 42)

Gartrell, M.; Paquet, U.; and Koenigstein, N., 2016. Bayesian low-rank determinantal point processes. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 349–356. (cited on page 101)

Gartrell, M.; Paquet, U.; and Koenigstein, N., 2017. Low-rank factorization of determinantal point processes. In *Thirty-First AAAI Conference on Artificial Intelligence*. (cited on pages 26, 45, 46, and 101)

Gelfand, I. M., 1989. *Lectures on linear algebra*. Courier Corporation. (cited on page 70)

Gillenwater, J.; Kulesza, A.; and Taskar, B., 2012a. Discovering diverse and salient threads in document collections. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 710–720. (cited on pages 5, 18, 90, 94, and 100)

Gillenwater, J.; Kulesza, A.; and Taskar, B., 2012b. Near-optimal map inference for determinantal point processes. *Advances in Neural Information Processing Systems*, 25 (2012). (cited on pages 23 and 29)

Gillenwater, J. A.; Kulesza, A.; Fox, E.; and Taskar, B., 2014. Expectation-maximization for learning determinantal point processes. *Advances in Neural Information Processing Systems*, 27 (2014). (cited on page 17)

Gimpel, K.; Batra, D.; Dyer, C.; and Shakhnarovich, G., 2013. A systematic exploration of diversity in machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1100–1111. (cited on page 93)

Gong, B.; Chao, W.-L.; Grauman, K.; and Sha, F., 2014. Diverse sequential subset selection for supervised video summarization. *Advances in neural information processing systems*, 27 (2014). (cited on pages 23, 25, 29, 114, and 118)

Gong, W.; Zhang, X.; Chen, Y.; He, Q.; Beheshti, A.; Xu, X.; Yan, C.; and Qi, L., 2022. DAWAR: Diversity-aware Web APIs Recommendation for Mashup Creation based on Correlation Graph. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 395–404. (cited on page 2)

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y., 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27 (2014). (cited on page 5)

Hadsell, R.; Chopra, S.; and LeCun, Y., 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 1735–1742. IEEE. (cited on page 15)

He, R.; Kang, W.-C.; and McAuley, J., 2017a. Translation-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*, 161–169. (cited on pages 48 and 64)

He, R. and McAuley, J., 2016a. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 191–200. IEEE. (cited on pages 14, 41, and 51)

He, R. and McAuley, J., 2016b. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, 507–517. (cited on page 74)

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M., 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 639–648. (cited on pages 13, 66, 75, and 76)

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S., 2017b. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 173–182. (cited on pages 1, 2, 4, 12, 13, 31, 34, 43, 61, 64, 66, 75, and 76)

Hemmi, S.; Oki, T.; Sakaue, S.; Fujii, K.; and Iwata, S., 2022. Lazy and Fast Greedy MAP Inference for Determinantal Point Process. *arXiv preprint arXiv:2206.05947*, (2022). (cited on page 29)

Henderson, J. C. and Brill, E., 2000. Exploiting diversity in natural language processing: Combining parsers. *arXiv preprint cs/0006003*, (2000). (cited on page 93)

Hidasi, B. and Karatzoglou, A., 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, 843–852. (cited on pages 14 and 15)

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D., 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, (2015). (cited on pages 14, 43, 44, and 93)

Hsieh, C.-K.; Yang, L.; Cui, Y.; Lin, T.-Y.; Belongie, S.; and Estrin, D., 2017. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web*, 193–201. (cited on page 14)

Hu, H. and He, X., 2019. Sets2sets: Learning from sequential sets with neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1491–1499. (cited on pages 4, 9, 11, 16, 51, 59, 89, 90, 96, 102, and 106)

Hu, L.; Sun, A.; and Liu, Y., 2014. Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 345–354. (cited on page 1)

Hu, Y.; Koren, Y.; and Volinsky, C., 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, 263–272. Ieee. (cited on pages 9 and 39)

Huang, W.; Da Xu, R. Y.; and Oppermann, I., 2019. Efficient diversified mini-batch selection using variable high-layer features. In *Asian Conference on Machine Learning*, 300–315. PMLR. (cited on page 18)

Jung, S.; Park, Y.-J.; Jeong, J.; Kim, K.-M.; Kim, H.; Kim, M.; and Kwak, H., 2021. Global-local item embedding for temporal set prediction. In *Fifteenth ACM Conference on Recommender Systems*, 674–679. (cited on page 90)

Kang, W.-C. and McAuley, J., 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, 197–206. IEEE. (cited on pages 1, 3, 14, 41, 48, and 49)

Kim, Y.; Kim, K.; Park, C.; and Yu, H., 2019. Sequential and Diverse Recommendation with Long Tail. In *IJCAI*, vol. 19, 2740–2746. (cited on pages 2 and 3)

Kingma, D. P. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, (2014). (cited on page 104)

Koren, Y., 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 426–434. (cited on pages 13 and 64)

Koren, Y.; Bell, R.; and Volinsky, C., 2009. Matrix factorization techniques for recommender systems. *Computer*, 42, 8 (2009), 30–37. (cited on pages 13 and 64)

Krishnan, A.; Cheruvu, H.; Tao, C.; and Sundaram, H., 2019. A modular adversarial approach to social recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1753–1762. (cited on pages 13 and 21)

Kulesza, A. and Taskar, B., 2010. Structured determinantal point processes. *Advances in neural information processing systems*, 23 (2010). (cited on pages 18, 90, and 94)

Kulesza, A. and Taskar, B., 2011a. k-DPPs: Fixed-size determinantal point processes. In *ICML*. (cited on pages 5, 18, 22, 23, 41, and 64)

Kulesza, A. and Taskar, B., 2011b. Learning determinantal point processes. (2011). (cited on page 23)

Kulesza, A.; Taskar, B.; et al., 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5, 2–3 (2012), 123–286. (cited on pages 17, 18, 23, 25, 28, 41, 45, 47, 64, 70, 102, and 118)

Kunaver, M. and Požrl, T., 2017. Diversity in recommender systems–A survey. *Knowledge-based systems*, 123 (2017), 154–162. (cited on page 20)

Lai, T.; Bui, T.; and Li, S., 2018. A review on deep learning techniques applied to answer selection. In *Proceedings of the 27th international conference on computational linguistics*, 2132–2144. (cited on page 119)

Li, C.; Jegelka, S.; and Sra, S., 2016. Fast dpp sampling for nystrom with application to kernel methods. In *International Conference on Machine Learning*, 2061–2070. PMLR. (cited on page 17)

Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J., 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1419–1428. (cited on page 14)

Li, J.; Wang, Y.; and McAuley, J., 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*, 322–330. (cited on page 49)

Li, Y.; Chen, T.; Zhang, P.-F.; and Yin, H., 2021. Lightweight Self-Attentive Sequential Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 967–977. (cited on page 49)

Liang, D.; Krishnan, R. G.; Hoffman, M. D.; and Jebara, T., 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, 689–698. (cited on page 13)

Liang, Y. and Qian, T., 2021. Recommending accurate and diverse items using bilateral branch network. *arXiv preprint arXiv:2101.00781*, (2021). (cited on pages 1, 15, 20, 39, 58, 76, and 101)

Liu, C.; Zhang, K.; Xiong, H.; Jiang, G.; and Yang, Q., 2014a. Temporal skeletonization on sequential data: patterns, categorization, and visualization. In

Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 1336–1345. (cited on page 93)

Liu, N. N.; Xiang, E. W.; Zhao, M.; and Yang, Q., 2010. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, 1445–1448. (cited on page 12)

Liu, Y., 2020. Recommending Inferior Results: A General and Feature-Free Model for Spam Detection. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 955–974. (cited on page 86)

Liu, Y.; Shen, Z.; Zhang, Y.; and Cui, L., 2021a. Diversity-promoting deep reinforcement learning for interactive recommendation. In *5th International Conference on Crowd Science and Engineering*, 132–139. (cited on page 2)

Liu, Y.; Walder, C.; and Xie, L., 2022. Determinantal Point Process Likelihoods for Sequential Recommendation. *arXiv preprint arXiv:2204.11562*, (2022). (cited on pages 2, 66, 76, 93, 96, 102, and 106)

Liu, Y.; Wei, W.; Sun, A.; and Miao, C., 2014b. Exploiting geographical neighborhood characteristics for location recommendation. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, 739–748. (cited on page 1)

Liu, Y.; Xiao, Y.; Wu, Q.; Miao, C.; Zhang, J.; Zhao, B.; and Tang, H., 2020. Diversified interactive recommendation with implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 4932–4939. (cited on pages 2, 31, 34, and 107)

Liu, Y.; Zhang, Y.; Wu, Q.; Miao, C.; Cui, L.; Zhao, B.; Zhao, Y.; and Guan, L., 2019. Diversity-promoting deep reinforcement learning for interactive recommendation. *arXiv preprint arXiv:1903.07826*, (2019). (cited on pages 20 and 27)

Liu, Z.; Chen, Y.; Li, J.; Yu, P. S.; McAuley, J.; and Xiong, C., 2021b. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*, (2021). (cited on page 50)

Luc, P.; Couprie, C.; Chintala, S.; and Verbeek, J., 2016. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, (2016). (cited on page 26)

Lv, F.; Jin, T.; Yu, C.; Sun, F.; Lin, Q.; Yang, K.; and Ng, W., 2019. SDM: Sequential deep matching model for online large-scale recommender system. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2635–2643. (cited on page 96)

McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A., 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 43–52. (cited on page 48)

Mnih, A. and Salakhutdinov, R. R., 2007. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20 (2007). (cited on page 64)

Morsy, S. and Karypis, G., 2017. Cumulative knowledge-based regression models for next-term grade prediction. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, 552–560. SIAM. (cited on page 89)

Pan, Z.; Cai, F.; Ling, Y.; and de Rijke, M., 2020. An intent-guided collaborative machine for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 1833–1836. (cited on page 96)

Pande, V.; Mukherjee, T.; and Varma, V., 2013. Summarizing answers for community question answer services. In *Language Processing and Knowledge in the Web: 25th International Conference, GSCL 2013, Darmstadt, Germany, September 25-27, 2013. Proceedings*, 151–161. Springer. (cited on page 18)

Puthiya Parambath, S. A.; Usunier, N.; and Grandvalet, Y., 2016. A coverage-based approach to recommendation diversity on similarity graph. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 15–22. (cited on pages 30, 50, 66, 76, and 107)

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 90)

Qin, L. and Zhu, X., 2013. Promoting diversity in recommendation by entropy regularizer. In *Twenty-Third International Joint Conference on Artificial Intelligence*. (cited on pages 1, 2, 11, and 20)

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al., 2018. Improving language understanding by generative pre-training. (2018). (cited on page 119)

Rafiei, D.; Bharat, K.; and Shukla, A., 2010. Diversifying web search results. In *Proceedings of the 19th international conference on World wide web*, 781–790. (cited on page 37)

Rahimi, R.; Montazeralghaem, A.; and Allan, J., 2019. Listwise neural ranking models. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, 101–104. (cited on page 4)

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L., 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, (2012). (cited on pages 2, 4, 31, 43, 61, 66, and 75)

Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L., 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, 811–820. (cited on pages 14, 41, 48, and 106)

Roberts, N.; Liang, D.; Neubig, G.; and Lipton, Z. C., 2020. Decoding and diversity in machine translation. *arXiv preprint arXiv:2011.13477*, (2020). (cited on page 37)

Rosenblatt, F., 1961. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY. (cited on page 104)

Ruby, U. and Yendapalli, V., 2020. Binary cross entropy with deep learning technique for image classification. *International Journal of Advanced Trends in Computer Science and Engineering*, 9, 10 (2020). (cited on page 42)

Saini, M.; Verma, S.; and Sharan, A., 2019. Multi-view ensemble learning using rough set based feature ranking for opinion spam detection. In *Advances in computer communication and computational sciences*, 3–12. Springer. (cited on page 86)

Santos, R. L.; Macdonald, C.; and Ounis, I., 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*, 881–890. (cited on page 66)

Shi, Y.; Larson, M.; and Hanjalic, A., 2010. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, 269–272. (cited on page 13)

Shuai, J.; Zhang, K.; Wu, L.; Sun, P.; Hong, R.; Wang, M.; and Li, Y., 2022. A review-aware graph contrastive learning framework for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1283–1293. (cited on page 118)

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A., 2013. Reasoning with neural tensor networks for knowledge base completion. *Advances in neural information processing systems*, 26 (2013). (cited on page 14)

Song, Y.; Yan, R.; Feng, Y.; Zhang, Y.; Zhao, D.; and Zhang, M., 2018. Towards a neural conversation model with diversity net using determinantal point processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32. (cited on pages 16 and 44)

Steck, H., 2015. Gaussian ranking by matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, 115–122. (cited on page 43)

Su, X. and Khoshgoftaar, T. M., 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009 (2009). (cited on page 64)

Sun, C.; Liu, H.; Liu, M.; Ren, Z.; Gan, T.; and Nie, L., 2020a. LARA: Attribute-to-feature adversarial learning for new-item recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 582–590. (cited on page 13)

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P., 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1441–1450. (cited on pages 14, 49, and 51)

Sun, J.; Guo, W.; Zhang, D.; Zhang, Y.; Regol, F.; Hu, Y.; Guo, H.; Tang, R.; Yuan, H.; He, X.; et al., 2020b. A framework for recommending accurate and diverse items using bayesian graph convolutional neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2030–2039. (cited on page 2)

Sun, L.; Bai, Y.; Du, B.; Liu, C.; Xiong, H.; and Lv, W., 2020c. Dual sequential network for temporal sets prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1439–1448. (cited on pages 9, 16, 89, 90, 95, 96, 98, 102, 106, and 111)

Tang, J. and Wang, K., 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 565–573. (cited on pages 1, 3, 14, 41, 48, and 49)

Tay, Y.; Anh Tuan, L.; and Hui, S. C., 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 world wide web conference*, 729–739. (cited on page 64)

Tong, Y.; Luo, Y.; Zhang, Z.; Sadiq, S.; and Cui, P., 2019. Collaborative generative adversarial network for recommendation systems. In *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, 161–168. IEEE. (cited on page 13)

Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; and Titov, I., 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, (2019). (cited on page 95)

Wan, Q.; He, X.; Wang, X.; Wu, J.; Guo, W.; and Tang, R., 2022. Cross Pairwise Ranking for Unbiased Item Recommendation. In *Proceedings of the ACM Web Conference 2022*, 2370–2378. (cited on pages 13 and 81)

Wang, B.; Yang, Y.; Xu, X.; Hanjalic, A.; and Shen, H. T., 2017a. Adversarial cross-modal retrieval. In *Proceedings of the 25th ACM international conference on Multimedia*, 154–162. (cited on page 90)

Wang, C.; Zhu, H.; Zhu, C.; Qin, C.; and Xiong, H., 2020. Setrank: A setwise bayesian approach for collaborative ranking from implicit feedback. In *Proceedings of the aaai conference on artificial intelligence*, vol. 34, 6127–6136. (cited on pages 2, 13, 14, 61, 75, and 81)

Wang, H.; Wang, N.; and Yeung, D.-Y., 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1235–1244. (cited on page 64)

Wang, J.; Yu, L.; Zhang, W.; Gong, Y.; Xu, Y.; Wang, B.; Zhang, P.; and Zhang, D., 2017b. 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*, 515–524. (cited on pages 13, 21, 24, 25, 27, 30, and 31)

Wang, P.; Chen, H.; Zhu, Y.; Shen, H.; and Zhang, Y., 2019a. Unified collaborative filtering over graph embeddings. In *Proceedings of the 42nd International ACM*

SIGIR Conference on Research and Development in Information Retrieval, 155–164. (cited on page 12)

Wang, W.; Huang, M.; Xu, X.-S.; Shen, F.; and Nie, L., 2018. Chat more: Deepening and widening the chatting topic via a deep model. In *The 41st international acm sigir conference on research & development in information retrieval*, 255–264. (cited on page 50)

Wang, X.; He, X.; Cao, Y.; Liu, M.; and Chua, T.-S., 2019b. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 950–958. (cited on page 76)

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S., 2019c. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 165–174. (cited on pages 13, 31, 50, 75, and 76)

Warlop, R.; Mary, J.; and Gartrell, M., 2018. Multi-Task Determinantal Point Processes for Recommendation. *arXiv preprint arXiv:1805.09916*, (2018). (cited on pages 20, 21, and 28)

Warlop, R.; Mary, J.; and Gartrell, M., 2019. Tensorized determinantal point processes for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1605–1615. (cited on pages 17, 42, 46, 59, 66, 97, and 101)

Wei, Y.; Wang, X.; Li, Q.; Nie, L.; Li, Y.; Li, X.; and Chua, T.-S., 2021. Contrastive learning for cold-start recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*, 5382–5390. (cited on page 118)

Weng, X.; Yuan, Y.; and Kitani, K., 2021. PTP: Parallelized tracking and prediction with graph neural networks and diversity sampling. *IEEE Robotics and Automation Letters*, 6, 3 (2021), 4640–4647. (cited on page 2)

Wilhelm, M.; Ramanathan, A.; Bonomo, A.; Jain, S.; Chi, E. H.; and Gillenwater, J., 2018. Practical diversified recommendations on youtube with determinantal point processes. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2165–2173. (cited on page 42)

Wu, H.; Xu, J.; Wang, J.; and Long, M., 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34 (2021), 22419–22430. (cited on page 16)

Wu, Q.; Gao, Y.; Gao, X.; Weng, P.; and Chen, G., 2019a. Dual sequential prediction models linking sequential recommendation and information dissemination. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 447–457. (cited on pages 4 and 15)

Wu, Q.; Liu, Y.; Miao, C.; Zhao, B.; Zhao, Y.; and Guan, L., 2019b. PD-GAN: Adversarial Learning for Personalized Diversity-Promoting Recommendation. In *IJCAI*, vol. 19, 3870–3876. (cited on pages 1, 2, 3, 15, 20, 21, 23, 25, 26, 27, 30, 31, 34, 41, 42, 44, 50, 58, 62, 66, 76, 101, 107, and 118)

Xia, F.; Liu, T.-Y.; Wang, J.; Zhang, W.; and Li, H., 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, 1192–1199. (cited on pages 4, 13, and 61)

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Ding, B.; and Cui, B., 2020. Contrastive Learning for Sequential Recommendation. *arXiv preprint arXiv:2010.14395*, (2020). (cited on page 15)

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; and Cui, B., 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, 1259–1273. IEEE. (cited on page 118)

Xie, Z.; Liu, C.; Zhang, Y.; Lu, H.; Wang, D.; and Ding, Y., 2021. Adversarial and Contrastive Variational Autoencoder for Sequential Recommendation. In *Proceedings of the Web Conference 2021*, 449–459. (cited on page 15)

Xu, C.; Zhao, P.; Liu, Y.; Xu, J.; S. Sheng, V. S. S.; Cui, Z.; Zhou, X.; and Xiong, H., 2019. Recurrent convolutional neural network for sequential recommendation. In *The world wide web conference*, 3398–3404. (cited on page 49)

Yang, S.; Yu, W.; Zheng, Y.; Yao, H.; and Mei, T., 2019. Adaptive semantic-visual tree for hierarchical embeddings. In *Proceedings of the 27th ACM International Conference on Multimedia*, 2097–2105. (cited on page 15)

Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J., 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 974–983. (cited on page 31)

Yu, F.; Liu, Q.; Wu, S.; Wang, L.; and Tan, T., 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM*

SIGIR conference on Research and Development in Information Retrieval, 729–732. (cited on pages 89 and 106)

Yu, L.; Sun, L.; Du, B.; Liu, C.; Xiong, H.; and Lv, W., 2020a. Predicting temporal sets with deep neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1083–1091. (cited on pages 5, 16, 89, 90, 91, 96, 102, and 106)

Yu, L.; Wu, G.; Sun, L.; Du, B.; and Lv, W., 2022. Element-guided Temporal Graph Representation Learning for Temporal Sets Prediction. In *Proceedings of the ACM Web Conference 2022*, 1902–1913. (cited on pages 16, 89, 90, and 106)

Yu, L.; Zhang, C.; Liang, S.; and Zhang, X., 2019. Multi-order attentive ranking model for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 5709–5716. (cited on pages 14, 49, and 50)

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y., 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, vol. 31. (cited on page 25)

Yu, R.; Liu, Q.; Ye, Y.; Cheng, M.; Chen, E.; and Ma, J., 2020b. Collaborative list-and-pairwise filtering from implicit feedback. *IEEE Transactions on Knowledge and Data Engineering*, (2020). (cited on page 13)

Yuan, F.; Karatzoglou, A.; Arapakis, I.; Jose, J. M.; and He, X., 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 582–590. (cited on page 14)

Zhang, C.; Wang, Y.; Zhu, L.; Song, J.; and Yin, H., 2021. Multi-graph heterogeneous interaction fusion for social recommendation. *ACM Transactions on Information Systems (TOIS)*, 40, 2 (2021), 1–26. (cited on page 21)

Zhang, G. P., 2003. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50 (2003), 159–175. (cited on page 89)

Zhang, M. and Hurley, N., 2008. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, 123–130. (cited on pages 15, 20, 30, 76, and 93)

Zhang, W.; Chen, T.; Wang, J.; and Yu, Y., 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of the 36th international*

ACM SIGIR conference on Research and development in information retrieval, 785–788. (cited on pages 30 and 32)

Zhao, P.; Luo, A.; Liu, Y.; Zhuang, F.; Xu, J.; Li, Z.; Sheng, V. S.; and Zhou, X., 2020. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*, (2020). (cited on page 89)

Zhao, X.; Xia, L.; Zou, L.; Yin, D.; and Tang, J., 2019. Toward simulating environments in reinforcement learning based recommendations. *arXiv preprint arXiv:1906.11462*, (2019). (cited on page 26)

Zheng, J. and Lu, G., 2021. k-SDPP: fixed-size video summarization via sequential determinantal point processes. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 774–781. (cited on page 18)

Zheng, Y.; Gao, C.; Chen, L.; Jin, D.; and Li, Y., 2021. DGCN: Diversified Recommendation with Graph Convolutional Networks. In *Proceedings of the Web Conference 2021*, 401–412. (cited on page 31)

Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; and Gai, K., 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1059–1068. (cited on pages 98 and 106)

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W., 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 11106–11115. (cited on pages 16, 89, and 93)